REPORT ON NETWORK

Created by Casey Donaldson

26/10/2023

Contents

Introduction

ACME Inc has recently allocated a new network engineer in response to the departure of the previous engineer. The hard over of the network was unhelpful and proven to show a lot of inconsistencies. Including a lack of documentation regarding the network. The board of ACME Inc's are understandably concerned at the current state of the network. Due to the rising concerns its ideal to proceed with a security test.

This document will serve as a thorough analysis of the security test. Presenting the procedures and final results of the security test. This includes a detailed subnet table, providing information of IP addresses in use and a detailed network diagram. This diagram illustrates the connection between devices, including interfaces and the corresponding IP address. The diagram will also reveal the layout and location of the subnets. The report will further explain how the network engineer came to this conclusion following a procedure with screen shot evidence.

Furthermore, this paper will discuss the security concerns of each device and if applicable, how to secure the device. Additionally, the paper will cover a critical network evaluation which will reveal the overall vulnerabilities at the network level including the structure of the network and viable solutions to help enhance the security.

Network diagram

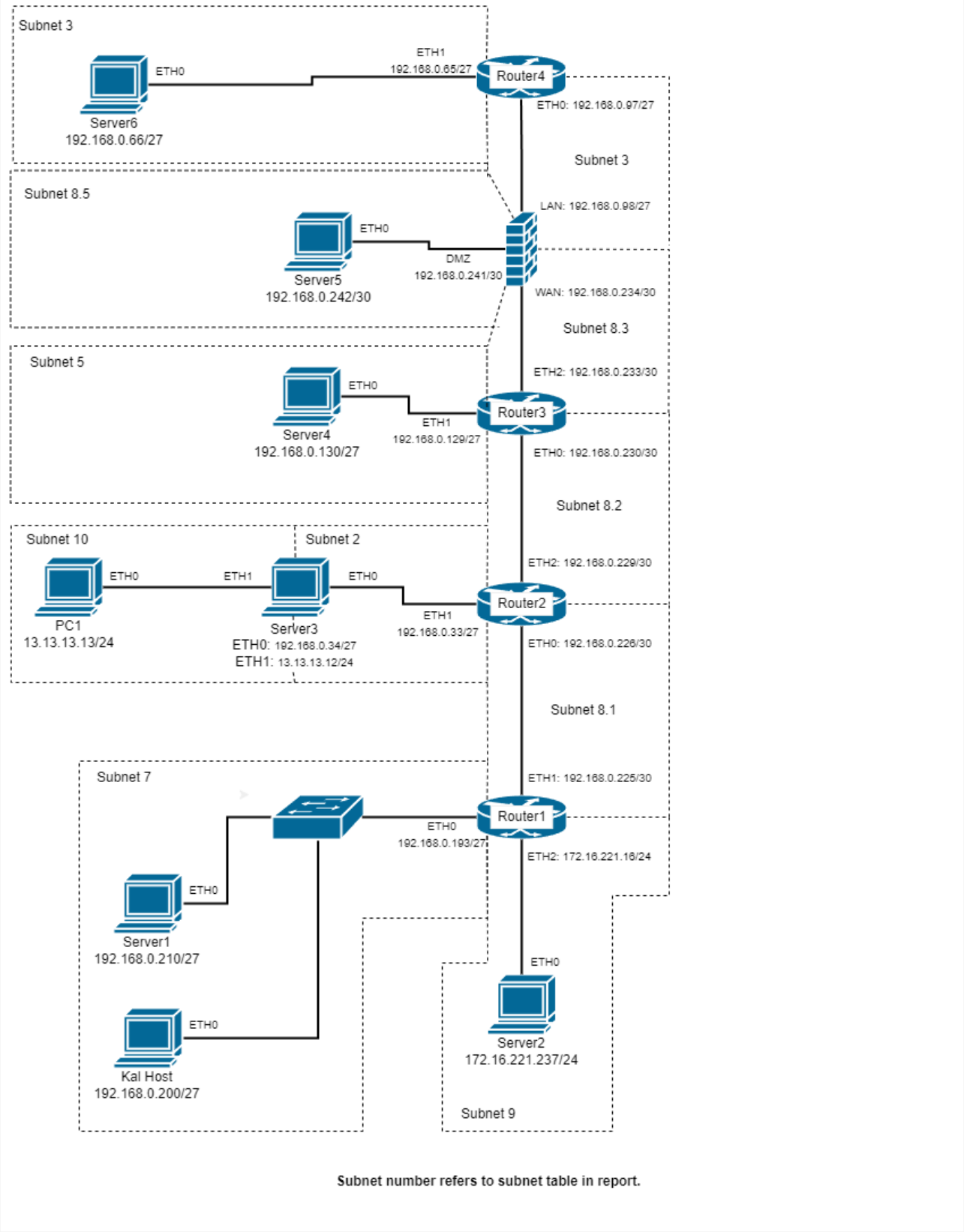


Figure 1:Network Diagram

Subnetting the network

The Network addresses found appears to start with 192.168.0.*/27 and /30 this suggested that the networks use VLSM technologies. The table below shows the finished subnets and in Appendix A, there is subnet calculations and a narrative to help follow along with the calculations. Further separate networks were found these were:

- 172.16.221.x
- 13.13.13.x

Both of these were deemed as class C and no subnetting calculations were required for them. The below table reveals how many active subnets are on the current network including subnets which aren't in use. The table further breaks down the subnets into the network address, the IP range, the broadcast address, and the IP addresses currently in use.

Network ID	Network address	Ip Range	Broadcast	Used IP addresses
1	192.168.0.0 /27	192.168.0.1 – 192.168.0.30	192.168.0.31	Not in use
2	192.168.0.32 /27	192.168.0.33 – 192.168.0.62	192.168.0.63	192.168.0.33 192.168.0.34
3	192.168.0.64 /27	192.168.0.65 – 192.168.0.94	192.168.0.95	192.168.0.65 192.168.0.66
4	192.168.0.96 /27	192.168.0.97 – 192.168.0.126	192.168.0.127	192.168.0.98 192.168.0.97
5	192.168.0.128 /27	192.168.0.129 – 192.168.0.158	192.168.0.159	192.168.0.129 192.168.0.130
6	192.168.0.160 /27	192.168.0.161 – 192.168.0.190	192.168.0.191	Not in use
7	192.168.0.192 /27	192.168.0.193 – 192.168.0.222	192.168.0.223	192.168.0.193 192.168.0.200 192.168.0.210
8.1	192.168.0.224 /30	192.168.0.225 – 192.168.0.226	192.168.0.227	192.168.0.225 192.168.0.226
8.2	192.168.0.228 /30	192.168.0.229 – 192.168.0.230	192.168.0.231	192.168.0.229 192.168.0.230
8.3	192.168.0.232 /30	192.168.0.233 - 192.168.0.234	192.168.0.235	192.168.0.233 192.168.0.234
8.4	192.168.0.236 /30	192.168.0.237 - 192.168.0.238	192.168.0.239	Not in use
8.5	192.168.0.240 /30	192.168.0.241 - 192.168.0.242	192.168.0.243	192.168.0.242 192.168.0.241
8.6	192.168.0.244 /30	192.168.0.245 -192.168.0.246	192.168.0.247	Not in use
8.7	192.168.0.248 /30	192.168.0.249 - 192.168.0.250	192.168.0.251	Not in use
8.8	192.168.0.252 /30	192.168.0.253 - 192.168.0.254	192.168.0.255	Not in use
9	172.16.221.0 /24	172.16.221.1 – 172.16.221.254	172.16.221.255	172.16.221.16 172.16.221.237
10	13.13.13.0/24	13.13.13.1 – 13.13.13.254	13.13.13.255	13.13.13.12 13.13.13.13

Table 1:Subnet table

Network mapping

Analysis on host

During the first startup of kali the host IP had to be confirmed. This was achieved using "ifconfig" shown below, Figure 2. This allowed the engineer to identify that the host has an IP address of 192.168.0.200 and a subnet mask of 255.255.255.224 on the "eth0" interface. This provides information that the "first thought to be a Class C address" is in fact a classless address. Earlier in the report the examiner covered how the network in submitted.

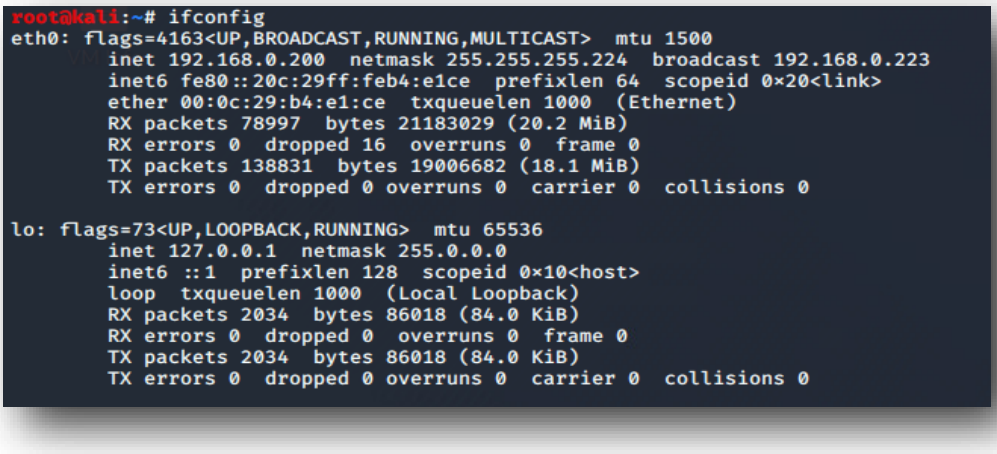


Figure 2:Kali Ifconfig command

First initial scanning

The first process for mapping the network was to use Nmap a network mapping tool. This allowed the new network engineer to figure out how the overall network is laid out. Scanning the subnet 192.168.0.200/27 as found on the host was the first task for mapping the network. This revealed 3 devices, Shown below, Figure 3.

- 192.168.0.193 (VyOS device)
- 192.168.0.200 (Host machine)
- 192.168.0.210 (Ubuntu device)

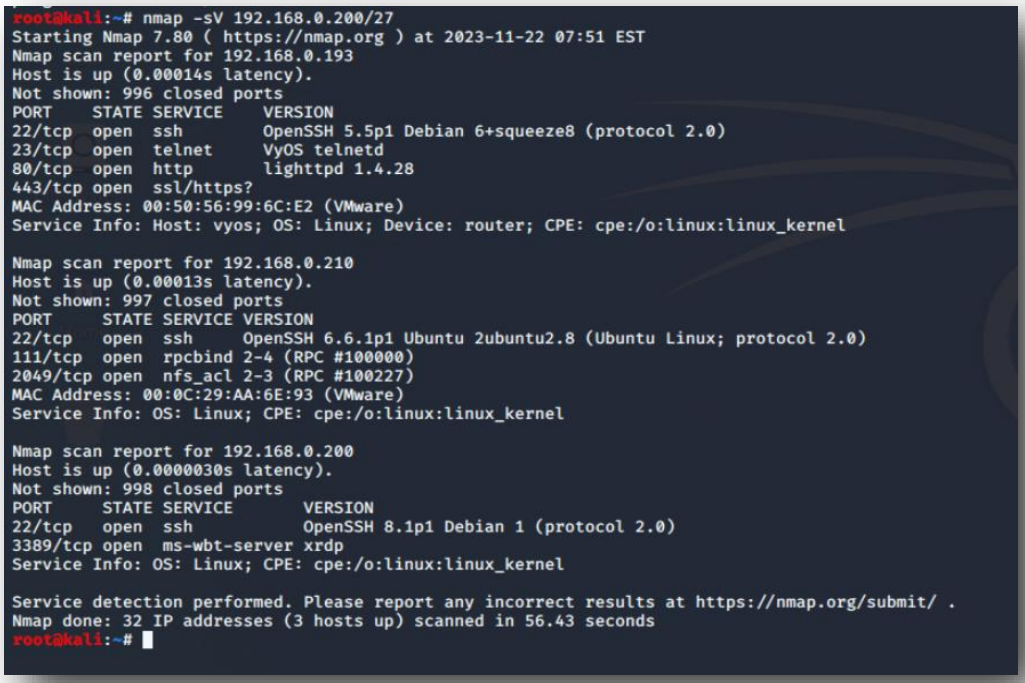


Figure 3:Nmap 192.168.0.200/27

These devices were identify as shown above. The VyOS device was identified to be using default credentials, this is discussed later in the report. The network engineer was able to telnet into 192.168.0.193(Router1) This allowed the engineer to use the command “show interfaces” shown below, Figure 4. This identified 3 networks. A further command used was “show ip route” This helped to find all the networks in which the router knew about. Shown below, Figure 5.

- 192.168.0.192/27 (Current network)
- 192.168.0.224/30 (New network)
- 172.168.221.0/24 (Separate network)

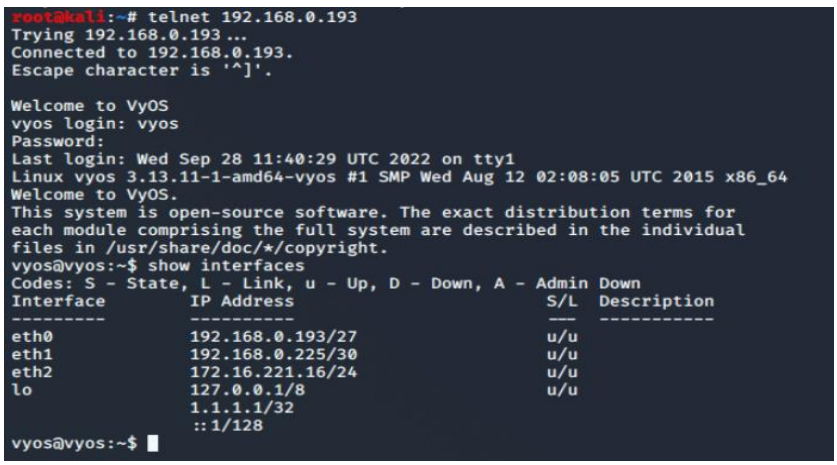


Figure 4:Telnet into 192.168.0.193 and using show interfaces

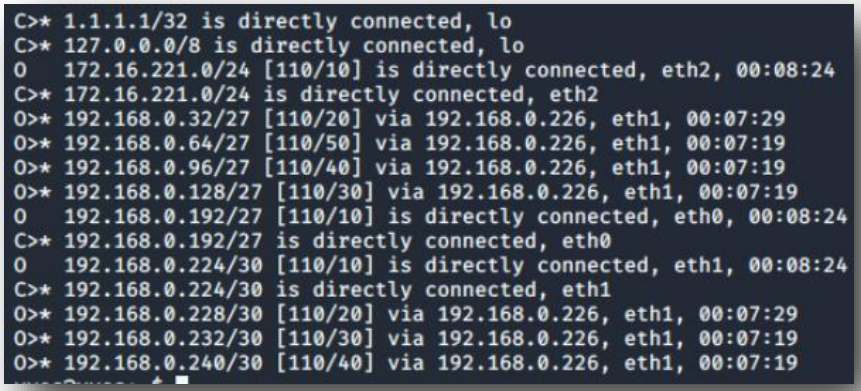


Figure 5:Show IP route 1.

The Network 172.168.221.0 will be discussed later in this report. The IP 192.168.0.224/30 was the next to be investigated. Using the nmap command "nmap -sV 192.168.0.224/30" This identified that another router on the IP, 192.168.0.226/30 was using the same software. This router was also vulnerable to default credentials.

The devices found are:

- 192.168.0.225/30 (Current network)
- 192.168.0.226/30 (New network)

Using the open telnet port, the tester connected to the router on the newly discovered IP using the credentials "vyos" for both the username and password. Using the same two commands as previously mentioned "show interfaces" and "show ip route" The network engineer managed to map more of the network. Shown below Figure 6 and Figure 7.

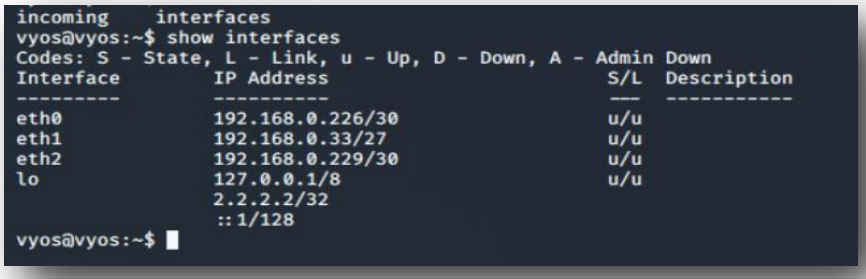


Figure 6:Telnet into 192.168.0.226 and use show interfaces

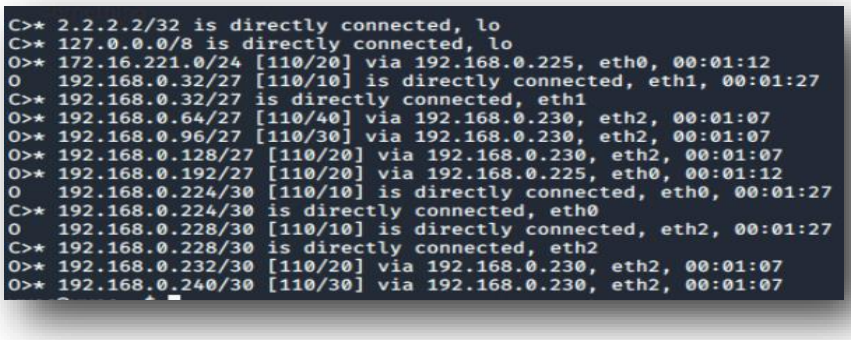


Figure 7:Show IP route 2.

The next part was the use another Nmap scan on the IP, 192.168.0.33/27. Shown below, Figure 8. This identified a new host device which will be examined further in the report.

Devices found are:

- 192.168.0.33/27 (Current Network VyOS device)
- 192.168.0.34/27 (New Host device)

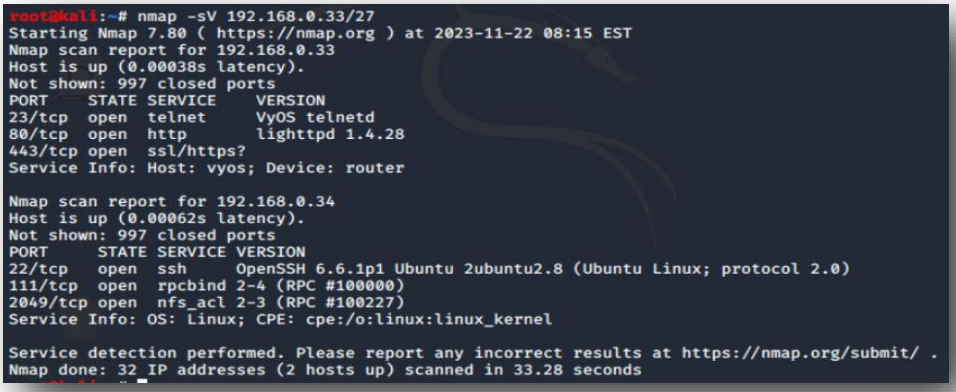


Figure 8:Nmap scan on 192.168.0.33/27

Another Nmap scan was conducted on 192.168.0.229/30, this revealed a further router running the same versions as the previous two routers. Shown below, Figure 9. The nmap revealed a total of two devices:

- 192.168.0.229/30 (Current network)
- 192.168.0.230/30 (New network)

Figure 9:Nmap scan on 192.168.0.229/30

This also allowed the tester to telnet into 192.168.0.230 with the default credentials and use the same previous commands. Shown below, Figure 10 and Figure 11.

This revealed 3 total networks which are connected to the router:

- 192.168.0.230/30 (Current network)
- 192.168.0.129/27 (New network)
- 192.168.0.233/30 (New network)

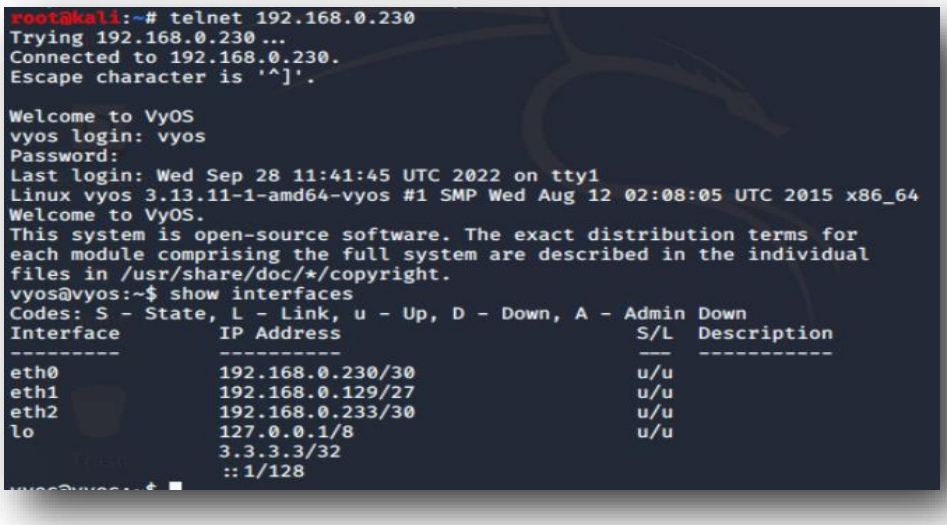


Figure 10:Telnet into 192/168.0.230 and use show interface

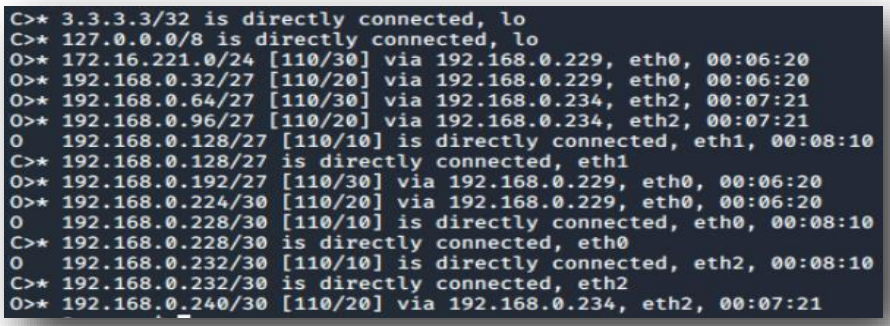


Figure 11:Show IP route 3

A Nmap scan was conducted on 192.168.0.129/27. This allowed the tester to find a host on the network scanned. Shown below, Figure 12. Total devices found:

- 192.168.0.129/27 (Current network)
- 192.168.0.130/27 (New host device)

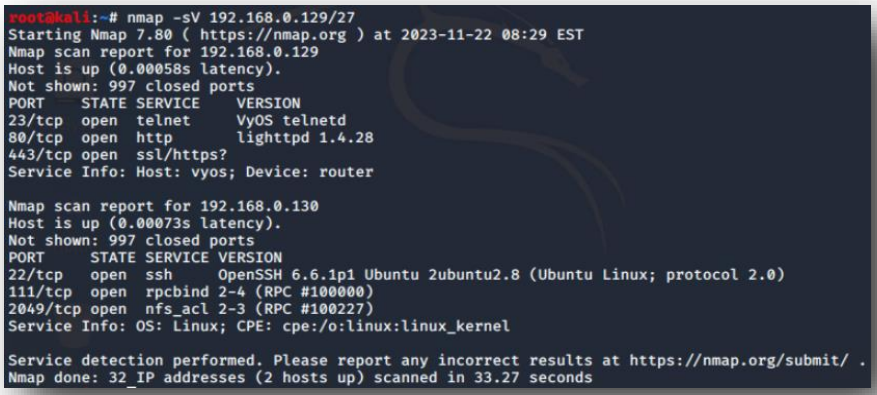


Figure 12:Nmap scan on 192.168.0.129/27

The other nmap scan conducted was on the IP 192.168.0.233/30. Shown below, Figure 13. This revealed no further devices just the interface already discovered:

- 192.168.0.233/30 (Current network)

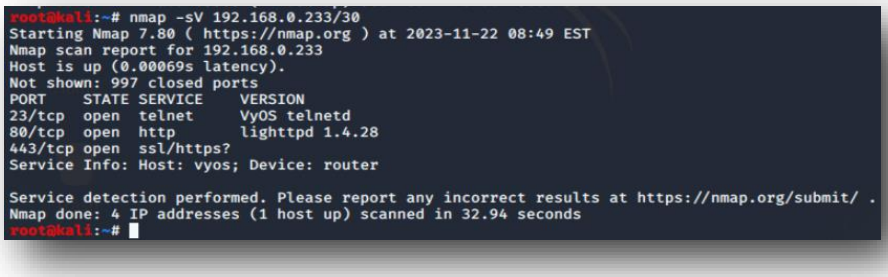


Figure 13:Nmap scan on 192.168.0.233/30

This brought the tester to a dead end on mapping the network as there were no further routers. However, with the command "show ip route" this revealed that the following networks exists through the IP 192.168.0.233. This might indicate a firewall within the network. Shown previously.

- 192.168.0.64/27
- 192.168.0.96/27
- 192.168.0.240/30

To examine these the Nmap tool was further used with the traceroute command, to find a route to these networks. Using the command "nmap -sV <IP> -traceroute" the results for the networks 192.168.0.64 and 192.168.0.96 revealed nothing. Shown below, Figure 14.

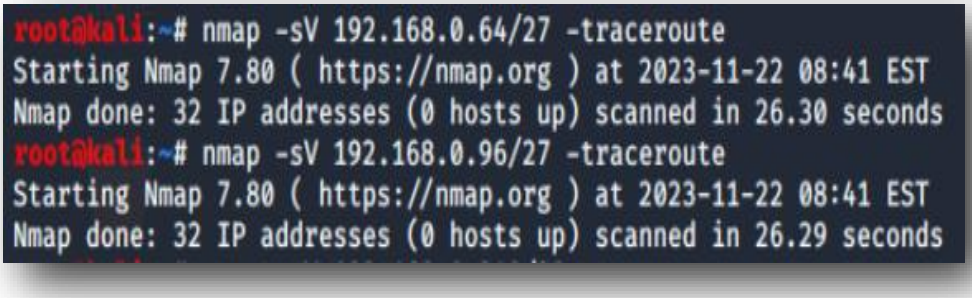


Figure 14:Nmap scan on 192.168.0.64/27 and 192.168.0.96/27

However, the Nmap scan on 192.168.0.240/30 revealed a hidden webserver and the traceroute to the service going through the interface 192.168.0.234. This resulted in the examiner possibly identifying a firewall which will need further analysis to prove this theory. Shown below, Figure 15.

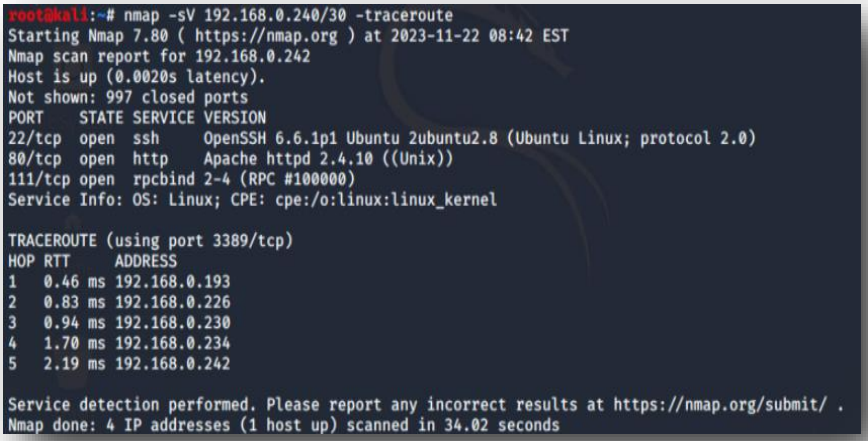


Figure 15:Nmap scan on 192.168.0.240/30

Below is a summary of the devices found with ports that are currently open and what needs to be further analysed and mapped. This does not include 172.168.221.X or 13.13.13.X as this was separate from the original VLSM network. Discussed and examined later. The colour code is used to differentiate the subnets.

IP	Ports	Version
192.168.0.33 (Router)	23 Telnet	VyOS telnetd
	80 Http	Lighttpd 1.4.28
	443 SSL/HTTPS	
192.168.0.34 (Linux)	22 SSH	OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8
	111 rpcblind	2-4 (RPC #100000)
	2049 nfs_acl	2-3 (RPC #100227)
192.168.0.129 (Router)	23 Telnet	VyOS telnetd
	80 HTTP	Lighttpd 1.4.28
	443 SSL/HTTPS	
192.168.0.130 (Linux)	22 SSH	OpenSSh 6.6.1p1 Ubuntu 2ubuntu2.8
	111 rpcblind	2-4 (RPC #100000)
	2049 nfs_acl	2-3 (RPC #100227)
192.168.0.225 (Linux Debian Router)	22 SSH	OpenSSH 5.5p1 Debian 6+squeeze8
	23 telnet	VyOS telnetd
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.226 (Router)	23 telnet	VyOS telnetd
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.229 (Router)	23 telnet	VyOS telnetd
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.230 (Router)	23 telnet	VyOS telnetd 1.14.0 or later
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.233 (Router)	23 telnet	VyOS telnetd 1.14.0 or later
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.242 (Linux)	22 SSH	OpenSSH 6.6.1p1 Ununtu 2ubuntu2.8
	80 http	Apache httpd 2.4.10 (Unxi)
	111 rpcblind	2-4 (RPC #100000)
192.168.0.193 (Linux Router)	22 SSH	OpenSSH 5.5p1 Debian 6+squeeze8
	23 telnet	VyOS telnetd
	80 http	Lighttpd 1.4.28
	443 ssl/https	
192.168.0.210 (Linux)	22 SSH	OpenSSH 6.6.1p1 Ubuntu
	111 rpcblind	2-4 (RPC #100000)
	2049 nfs_acl	2-3 (RPC)
192.168.0.200 (Kali Host)	22 SSH	OpenSSH 8.1p1 Debian 1
	3389 ms-wbt server xrdp	

Table 2:Initial Nmap scan

Further IP addresses were discovered later on in this report, their respective ports are discussed along with the discovery of them. Furthermore, UDP scan took place and can be seen in Appendix D

(UDP Scanning). These scans revealed nothing that was used for the procedures. Hence, they weren't added to the above table.

Router mapping

During the analysis of the IP addresses that are running Telnet with the version "VyOS", these all appear to be routers according to the Vyos website. (VyOS, 2023) during reading this website for further information, default credentials were mentioned these were as follows: Username:vyos, Password:vyos. (Vyos, 2023)

Below is a table of 3 routers which are currently known. Table 3:Identified routers. These routers also define several networks and their corresponding subnet mask which will help to subnet the network and identify areas of interest. The naming scheme of the routers are based on the closest connection to the host machine. (Kali machine)

Device	Interface	Network Address	IP Address	Subnet Mask
Router 1	Eth0	192.168.0.192	192.168.0.193	255.255.255.224
	Eth1	192.168.0.224	192.168.0.225	255.255.255.252
	Eth2	172.16.221.0	172.16.221.16	255.255.255.0
Route 2	Eth0	192.168.0.224	192.168.0.226	255.255.255.252
	Eth1	192.168.0.32	192.168.0.33	255.255.255.224
	Eth2	192.168.0.228	192.168.0.229	255.255.255.252
Router 3	Eth0	192.168.0.228	192.168.0.230	255.255.255.252
	Eth1	192.168.0.128	192.168.0.129	255.255.255.224
	Eth2	192.168.0.232	192.168.0.233	255.255.255.252

Table 3:Identified routers

This has proven to be beneficial in helping to map out the network and identify other devices needing to be analysed. Please note a further router was discovered and will be discussed later.

192.168.0.210 (Server1)

The first device on the network connected to the kali machine was 192.168.0.210/27, this device will be classed as Server1 from this point forward. The Nmap scan on Server1 showed that ports 111 and 2049 are open indicating a file share server and this allowed for the kali machine to mount onto the drive. The first test was initiated using the command "showmount -e 192.168.0.210" this allowed the tester to identify the directory that was shared. The root directory for this PC appears to be shared and this is a critical vulnerability, discussed in the section security weakness.

The next task was to mount the NFS share and identify if the passwd and shadow files contains available passwords

Once the engineer could find the files the next step was to copy the "shadow" and "passwd" files to the kali machine and use the "unshadow" command to create another file which will be used with a tool called John the Ripper to crack the passwords in the files. Shown below, Figure 16, Figure 17 and Figure 18.

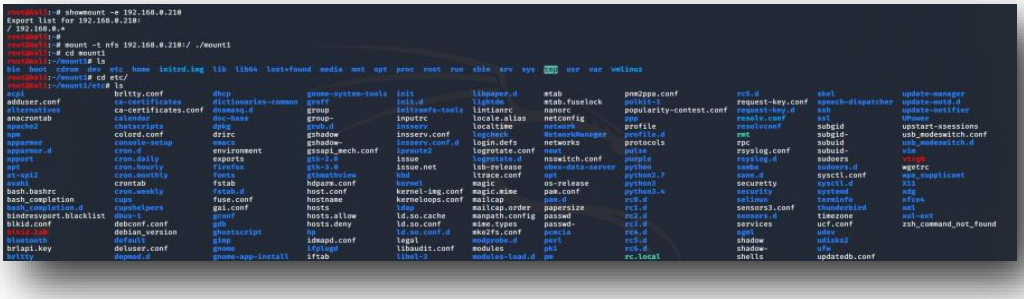


Figure 16:NFS analysis on 192.168.210

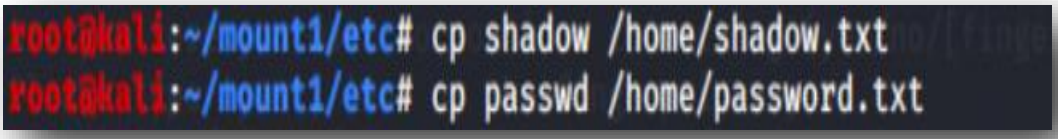


Figure 17:Copying passwd and shadow from 192.168.0.210

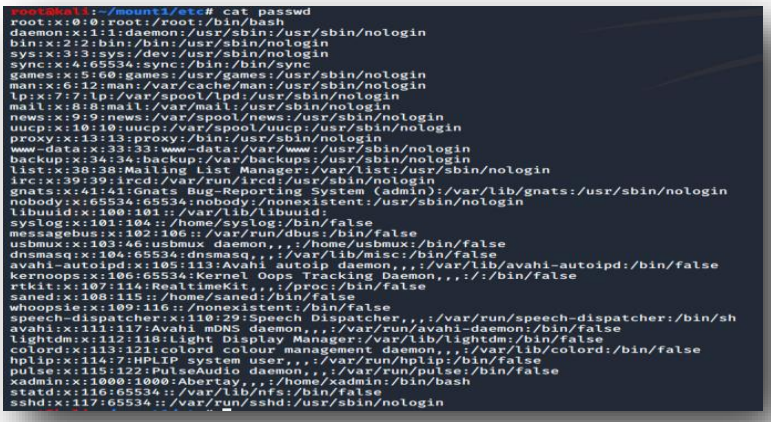


Figure 18:Passwd content

The username found was called “xadmin” the password found in correlation to this account was “plums”. These credentials were checked using the SSH protocol on Server1. The engineer managed to gain access to the system. Shown below, Figure 19. Furthermore, the vulnerable password could

indicate that a brute force attack could be executed for an effective way of gaining access without using the NFS shares.

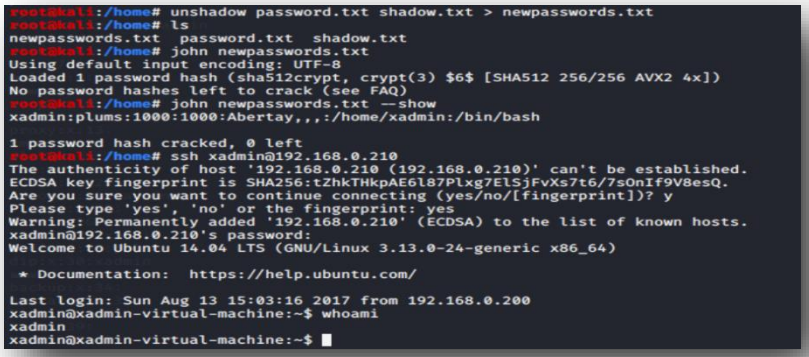


Figure 19:Using the tool John and testing the password

172.16.221.237 (Server2)

On Router 1, Eth2 another network was discovered. Using the IP address of 172.16.221.16/24, the tester used another Nmap scan on the corresponding network. This provided details of another device. Shown below, Figure 20. The IP 172.16.221.16 relates to the routers interface, whereas 172.16.221.237 relates to another device. On this device the nmap scan identified some ports and versions.

- Port 80 HTTP running Apache httpd 2.2.22 (Ubuntu)
- Port 443 HTTPS running Apache httpd 2.2.22

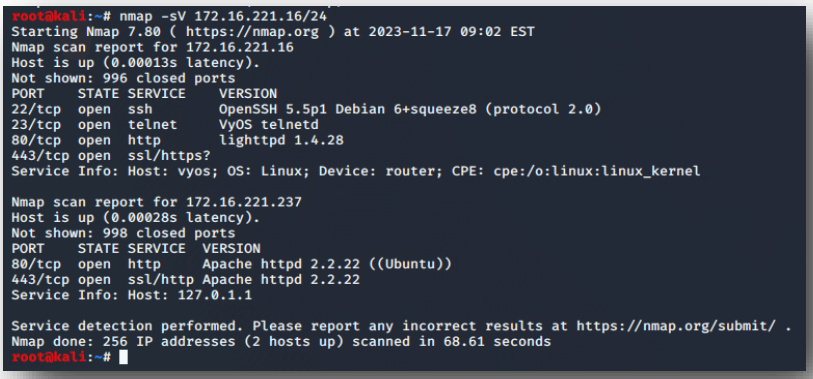


Figure 20:Nmap scan on 172.16.221.16/24

This indicates that 172.16.221.237 is in fact a webserver. From this point this device will be called Server 2. Two tools that were initially used on the webserver, included “Nikto” and “Dirb” These

allowed for two types of enumeration. Nikto was used to automate vulnerability scanning. Little came from this. Shown below, Figure 21.

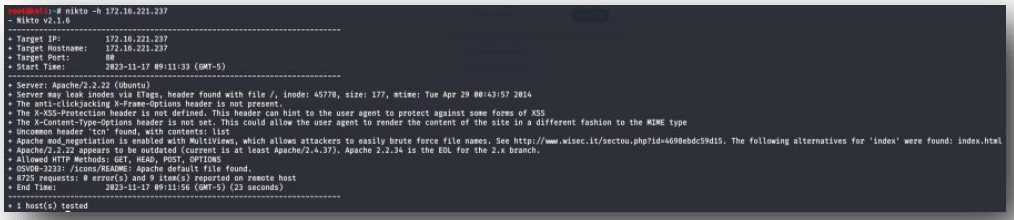


Figure 21:Nikto scan on 172.16.221.237

The next tool used was Dirbuster. This allowed to brute force directories to help find hidden directories. The results came back showing that the webserver was created with WordPress according to the directories found. Furthermore, an admin panel was discovered in the results. This could possibly be brute forced. Shown below, Figure 22.

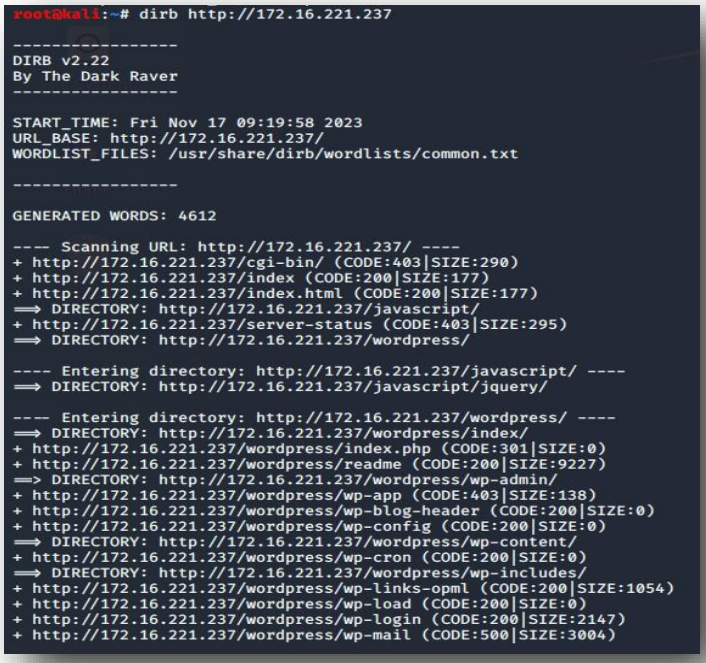


Figure 22:Dirb on 172.16.221.237

Attempting the brute force technique, the examiner had to first discover the username. This was achieved by using Metasploit and the payload “auxiliary/scanner/http/wordpress_login_enum”. This allowed the examiner to brute force usernames. The username that appears valid was “admin” Shown below, Figure 23.

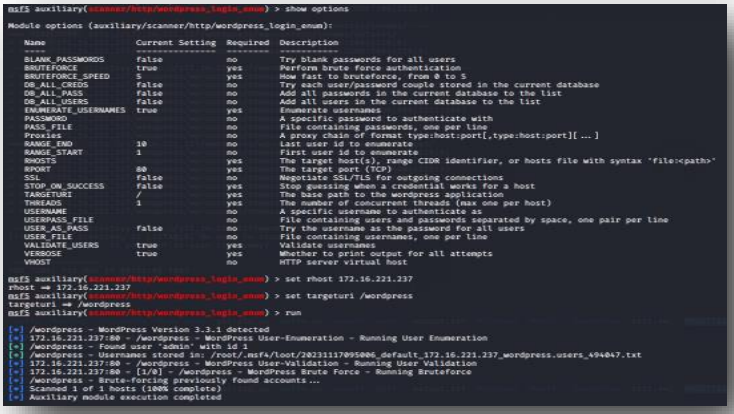


Figure 23:WordPress login attempt, username gathering

The next step was to add "admin" to the "USERNAME" field and add a password list to brute force a login. Shown below, Figure 24. The password list used was password.lst found in the "/usr/share/wordlists/Metasploit" directory.

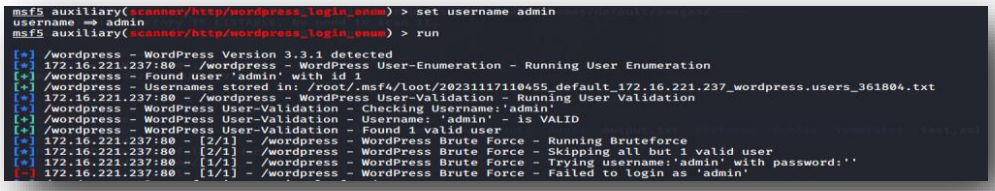


Figure 24:WordPress login attempt, password cracking

Once the options were set, the program was executed. The results took some time, but a password was found, the admin password is "zxc123". Shown below, Figure 25. This indicates that the network engineer could create a PHP script on the webserver using the WordPress admin panel and attempt a shellshock attack to gain a shell.

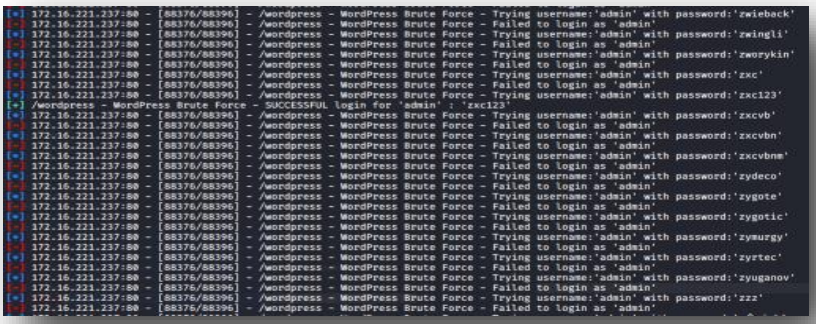


Figure 25: WordPress login attempt, found password

To demonstrate this the examiner first logged into WordPress with the given credentials which was successful. Show below, Figure 26.

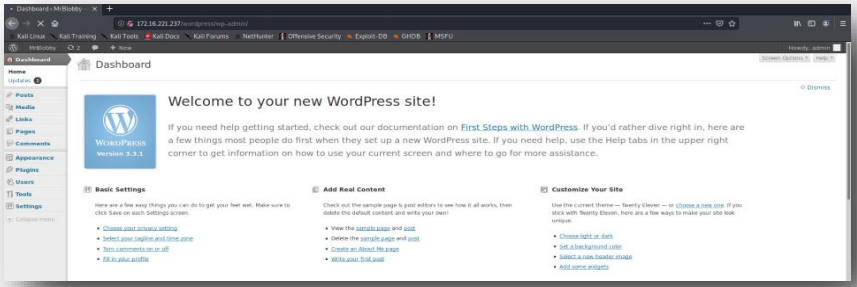


Figure 26:WordPress login attempt, password testing

An article informed the examiner where to identify an area in which to insert reverse shell script (Chandel, 2019). The area was the 404 page which allowed the admin account to add a PHP script. A payload on the kali machine was used, this was found in the following location "/usr/share/webshells/php/php-reverse-shell.php" The payload was created by Pentestmonkey (Pentestmonkey, n/a).

Inserting the payload is shown below, Figure 27.

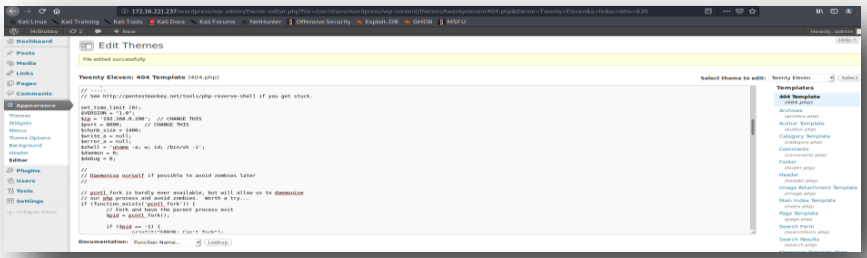


Figure 27:Wordpress, uploading a reverse PHP shell

The changes were then saved, and the next step was to create a URL that access page 404, this performs a type of URL injection attack to create the reverse TCP handler. Before inserting the URL to direct towards the script, the kali must first be listening on the intended port using Netcat. 8000 in this case. The URL injected was simply accessing 404.php page. As shown below, Figure 28.

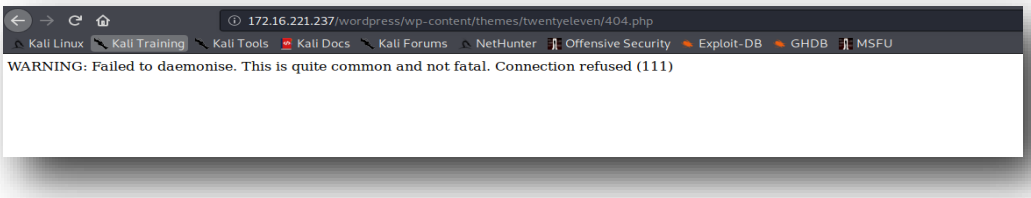


Figure 28:WordPress, connecting to reverse shell

The exploit was successful, and a shell was obtained. As seen below, Figure 29.

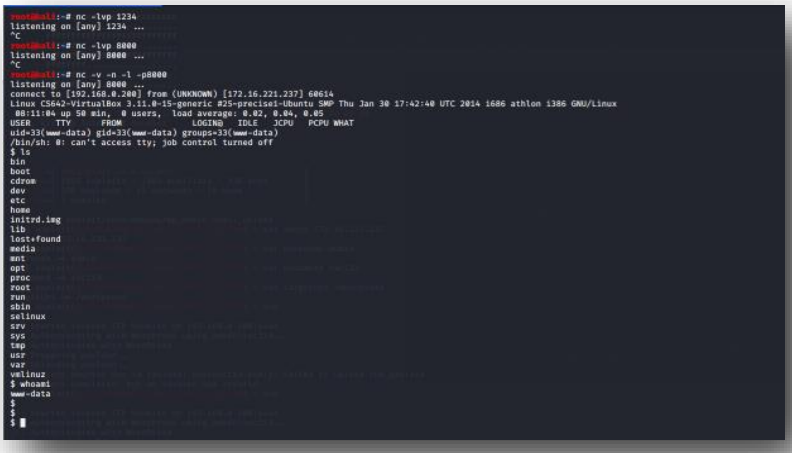


Figure 29:Nc listener to capture reverse shell

Another approach is to obtain a meterpreter this can be seen below, Figure 30. This allows for further attacks however IP 172.16.221.237 does not have further use as the machine is not multi-homed, and no machines exists behind it.

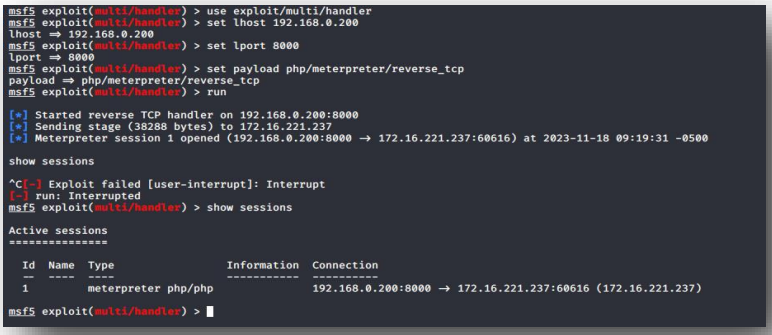


Figure 30:WordPress obtaining a meterpreter

The vulnerabilities of this machine will be discussed in Security Weakness section.

A further 2 ways were identified for username enumeration. This included the error response given from the website when you entered a username. The response would tell you if the password or username was incorrect. This is a common vulnerability of the WordPress application. Another tool which was trialled was WPScan. This tool basically tests WordPress applications. Since this Penetration / network mapping test does not resolve around web application hacking no more further tests were conducted on the website service.

192.168.0.34(Server3)

This machine according to the initial Nmap scan reveals that 192.168.0.34 is a server and will be named Server 3 in this report.

The first test conducted on this machine involved checking the NFS share to find if the share reveals any vulnerabilities, The share was shared at the "xadmins" home directory. Although this is not as vulnerable as the root directory that was shared on Server1 this could still be exploited. Shown below, Figure 31 and Figure 32.

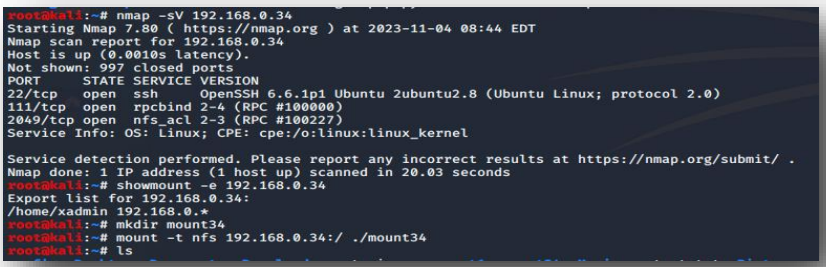


Figure 31:Nmap scan on 192.168.0.34 and using show mount command

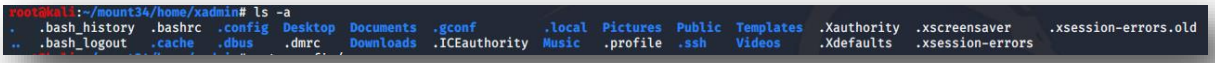


Figure 32:Show mount directory content

Attempting to validate the host with the password was a tedious process and an attempt to copy the kali machine credentials to authorised keys on Server3 was made. This was achieved by generating a RSA key, shown below, Figure 33.

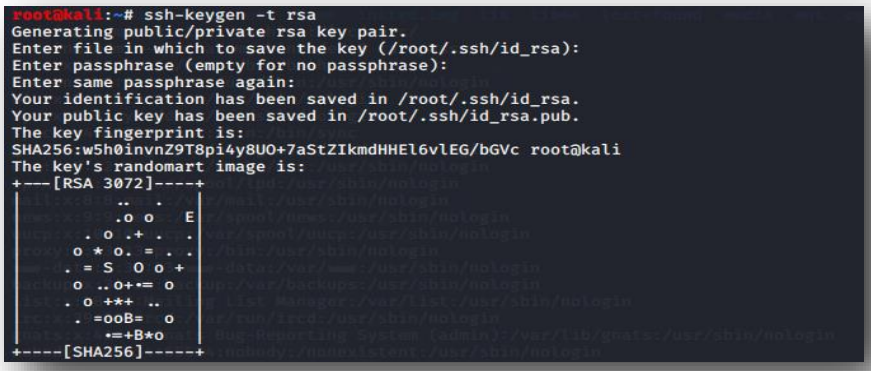


Figure 33:RSA key generation

Using the "SCP" function in Linux the tester was able to copy the key over to the "authorised_keys" file found using the NFS share. Shown below, Figure 34. Xadmins password was needed, and the password used was "plums" found earlier in this report.

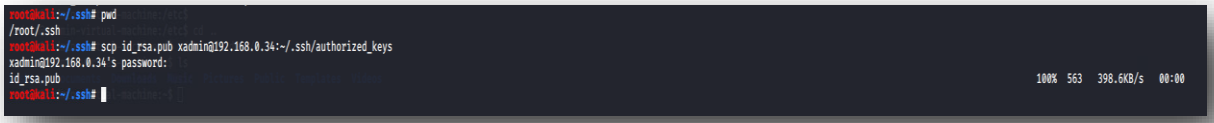


Figure 34:Secure copy of public key to 192.168.0.34/.ssh/authorized keys

To verify that this technique worked the tester had to SSH into Server3. The attempt was successful, and the network engineer did not need to authenticate. Shown below, Figure 35.

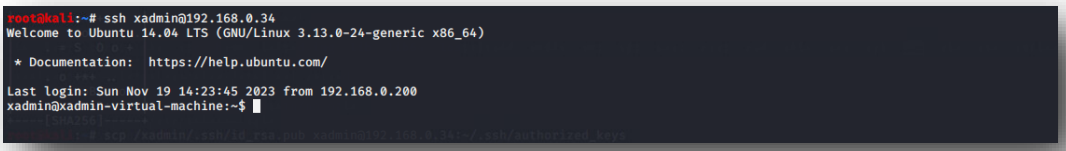


Figure 35:Instant access to 192.168.0.34

Using the "ifconfig" command this showed that the device 192.168.0.34 was a multi-homed device and had a network on a different interface (13.13.13.12/24) Shown below, Figure 36.

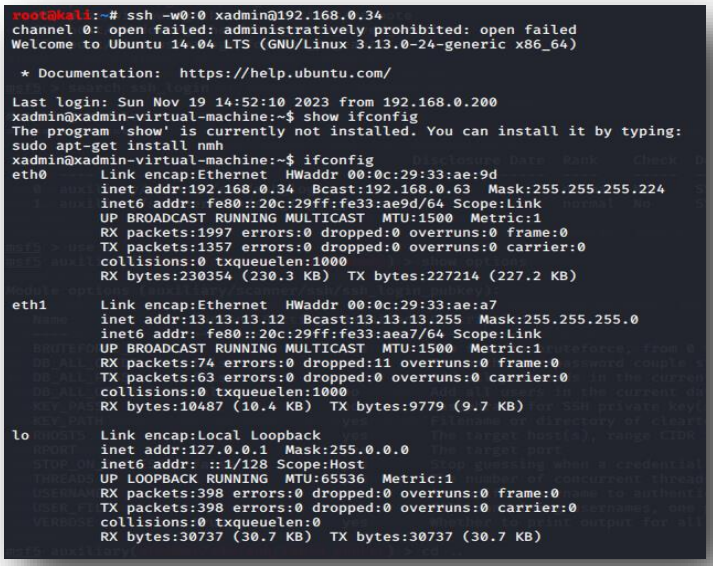


Figure 36:Ifconfig and SSH into 192.168.0.34

To identify if any hosts were on this network, a scripted ping command was used to ping 13.13.13.0-255. This revealed that 13.13.13.13 was active on the other network. Shown below, Figure 37.

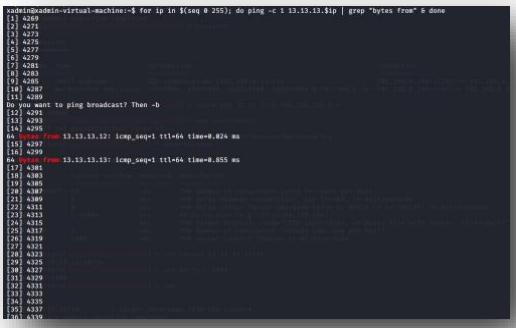


Figure 37:Ping sweep on 13.13.13.X network

The network engineer used Metasploit and the exploit “auxiliary/scanner/ssh/ssh_login” to login to the SSH protocol on 192.168.0.34. Shown below, Figure 38.

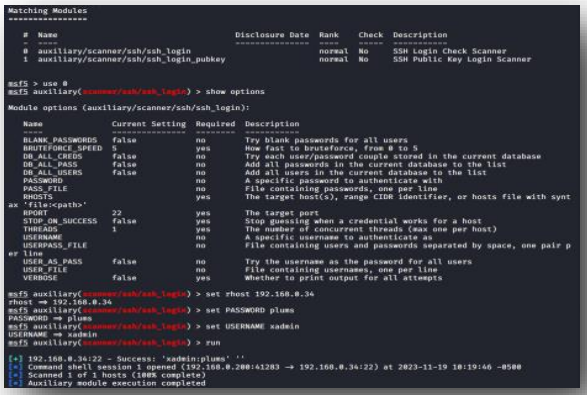


Figure 38:SSH login using meterpreter on 192.168.0.34

A post exploit was used after to upgrade the session to a meterpreter. The exploit used was “post/multi/manage/shell_to_meterpreter” this gave the engineer more privileges. Shown below, Figure 39.

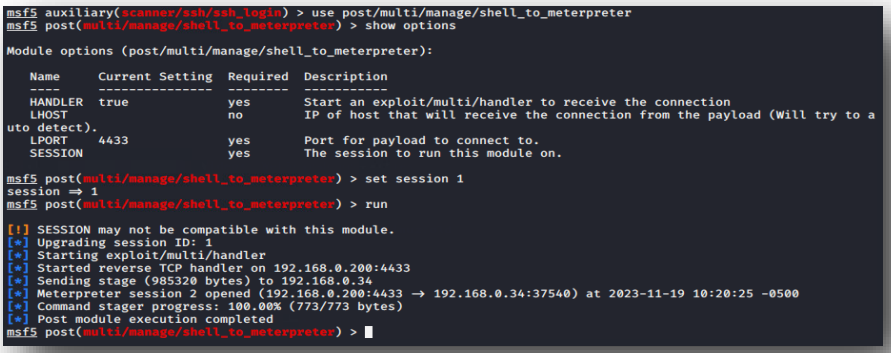


Figure 39:Privilage escalation using meterpreter

13.13.13.13(PC1)

This device was found behind Server3. From Metasploit a TCP port scan was used on 13.13.13.13 using the meterpreter gained in the previous section, before running the payload the destination, ports, and the route to the 13.13.13.0 network had to be added, Shown below, Figure 40 and Figure 41.

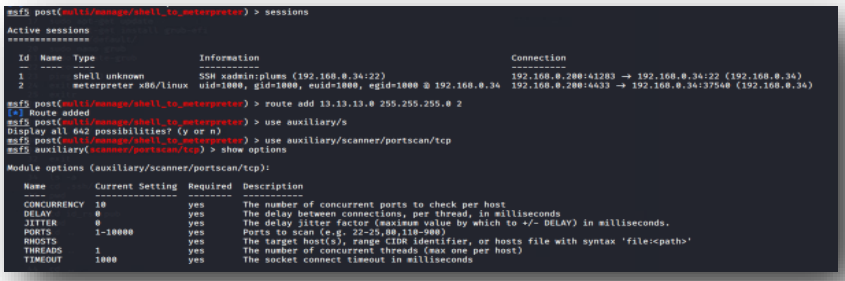


Figure 40:Adding a route towards the target

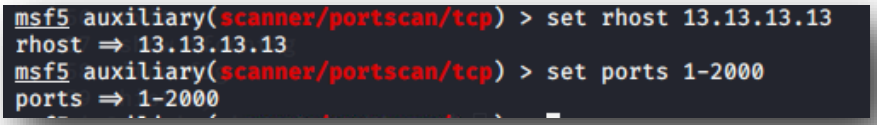


Figure 41:Setting target and port range

Once all the settings were set the application was executed. This revealed that port 22 (SSH) was the only common port open. Shown below, Figure 42.

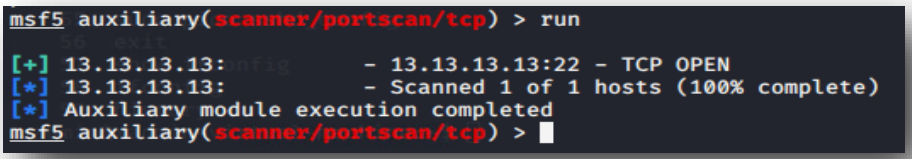


Figure 42:Ports identified on 13.13.13.13

This allowed the engineer to identify this device to be a PC since this was not providing a service. This device will be referred to as PC1 on the network diagram.

This allowed the engineer to use the “ssh_login” that was previous used in this report. The settings which had to be set were

- Remote host = 13.13.13.13
- Password = “plums”
- Username = “xadmin”

The payload was used and completed but no meterpreter was created this indicated that the password was incorrect for this device as meterpreter informed the username exists. As seen below, Figure 43.

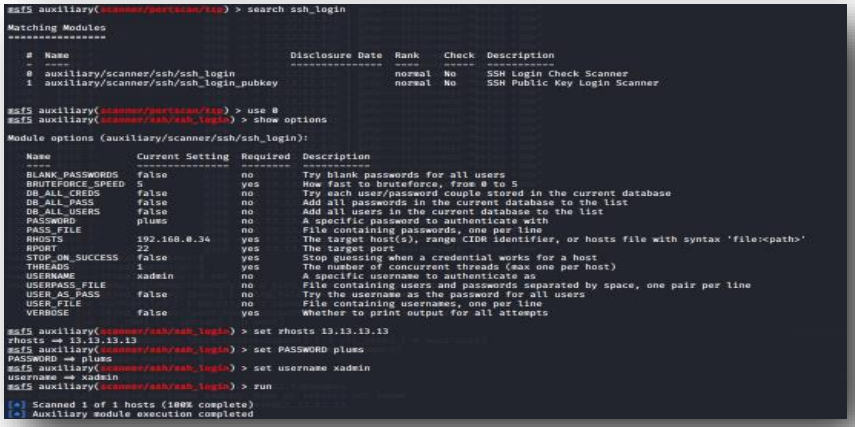


Figure 43:Failed SSH attempt

The next test was to brute force the login. Adding a password list found in "/usr/share/wordlist/Metasploit/password.lst" to the "PASS_FILE" field allowed the payload to brute force the login, the tester also turned the verbose mode to true to keep track of the progress. The brute force attempt was successful as a meterpreter was gained and the password was identified as "!gatvol". Shown below, Figure 44.

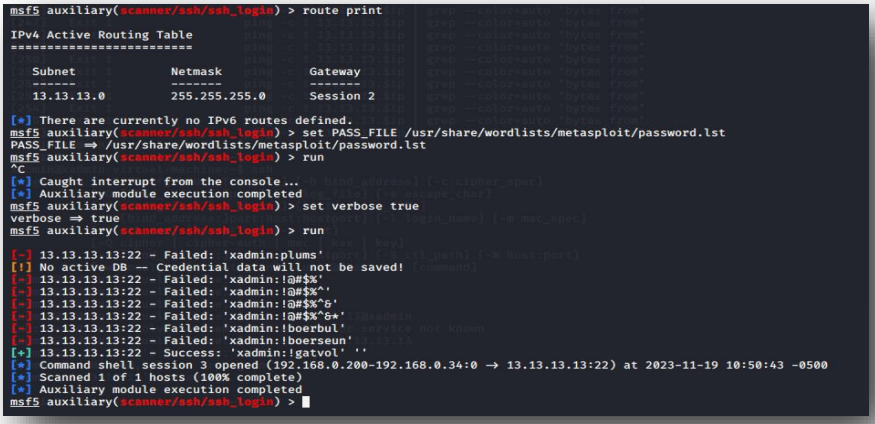


Figure 44:Brute force attempt on 13.13.13.13

The network engineer used the session gained and the command "ifconfig" within a shell to identify if there were any further devices on the network. This revealed that PC1 had only one interface and this concluded the end of the pivoting on Server3, shown below, Figure 45.

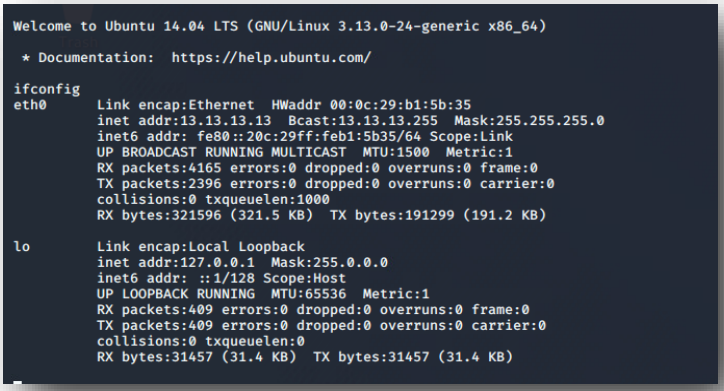


Figure 45:Ifconfig on 13.13.13.13

A security analysis on Server3 and PC1 will be conducted in the Security Weakness section of this report.

192.168.0.130 (Server4)

This device reveal that it was acting as a server as it was running ports 111 and 2049 indicating an NFS server. This device appears to be more robust than previous devices as the SSH protocol would not allow users to login with a password, and validated users based on the public key. Every username that the tester tried to SSH into, the device would be immediately rejected.

The examiner did identify one vulnerability. This was identified when the command "showmount -e 192.168.0.130" was used as the directory shared was /home/xadmin. Shown below, Figure 46.

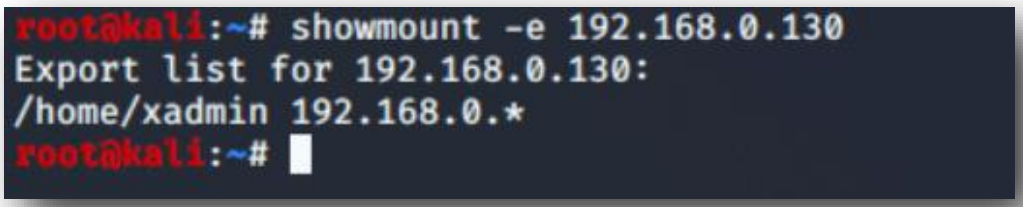


Figure 46:Show mount on 192.168.0.130

This is not as serious as the root folder being shared as shown on Server1, however this would allow a hacker to mount the device and try to copy their "public_key" to the device and gain instant access to the device.

A file called "authorised_keys" was found and this contained information about a host within the network who has permissions to access this device. Shown below, Figure 47.

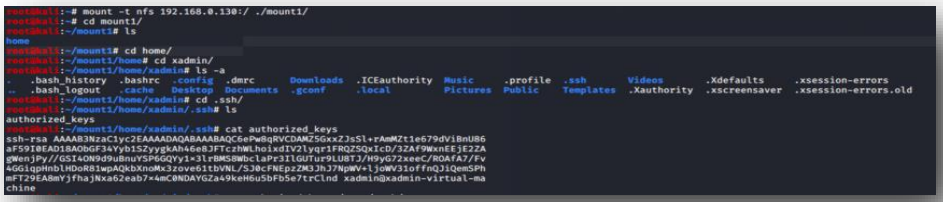


Figure 47:Authorized keys on 192.168.0.130

There are two methods of gaining access to this device, the first method which is more desirable because it causes less noise on the network and gives the hackers machine, ease of access. This method will be executed by copying the authorized private RSA key, to the kali machine.

The first task was to verify that a machine has access to the 192.168.0.130 machine via a private key. this can be seen above or in Appendix B, Figure 69. The next step is to copy the private key from the valid machine to the hacker's machine. This was achieved by a trial-and-error process, discussed later. The machine with valid credentials was 192.168.0.34. The private key had to be copied from this machine to the Kali Linux machine in the "/root/.ssh/" directory. As seen in Appendix B, Figure 70. Once the key was successfully copied the tester had to verify that this works. This can be seen in Appendix B, Figure 71. The Kali machine was able to authenticate without the use of a password.

Another way of gaining access is to identify who already has access to the machine and going through that machine to gain access via SSH. This was tested by going through all the compromised devices current found and compromised.

- 192.168.0.210 – Denied access
- 192.168.0.34 – Access granted

Luckly the process was faster than expected the machine with access is shown above. This was the second machine tested. This is more robust than weak or default credentials however this relies on the resilience of the machine 192.168.0.34 which was not a great idea for this network. Shown below, Figure 48.

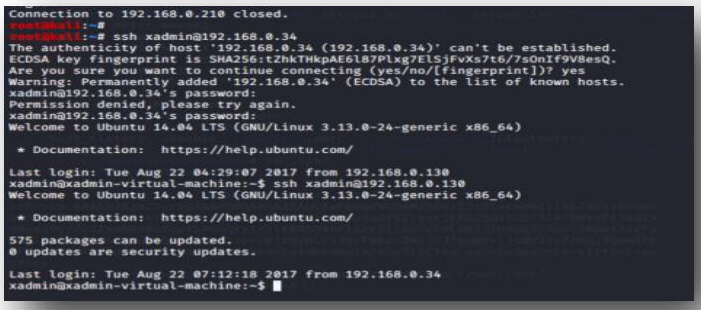


Figure 48:SSH into 192.168.0.34 then into 192.168.0.130

Once access was gained the command “ifconfig” was used and this identified that the machine was an endpoint, and no further attacks were needed. Shown below, Figure 49.

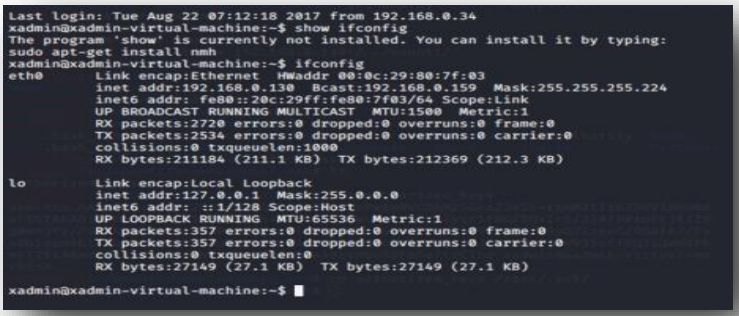


Figure 49:Ifconfig on 192.168.0.130

192.168.0.242 (Server5)

To this point a firewall could potentially sit between router 3 and device 192.168.0.242. This was based on the results from “show ip route” on Router3. The images shows that there are 192.168.0.64, 96 and 242 networks behind the 2nd interface of router3 (192.168.0.233) as determined by OSPF.

To better understand the network and verify that the path is correctly interpretated. Nmap with a firewall script, including the traceroute command was used. Shown below, Figure 50.

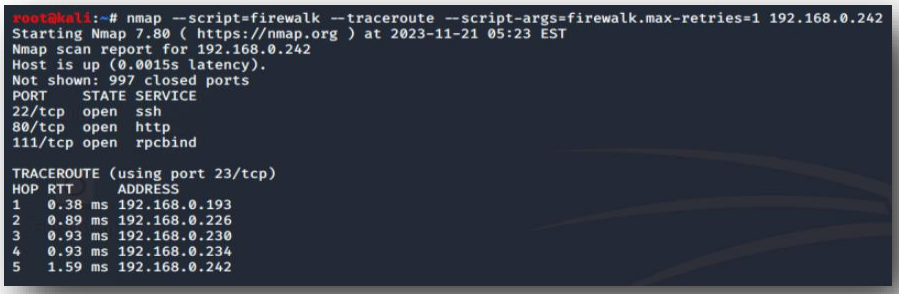


Figure 50:Nmap script firewalk on 192.168.0.242

This revealed the same ports originally found in the initial Nmap scan. This results in the device being called Server5 as it appears to be hosting a website service. The traceroute has also gone through the interface 192.168.0.234 which was not found on the original Nmap scan.

Since Server5 has been identified running a web service then the tool Nikto that was previously used was utilized again. The command “nikto -h 192.168.0.242” was used to try and identify any vulnerabilities within the website. Unfortunately, there is a major vulnerability called Shellshock. This vulnerability will be discussed in the security weakness section. The tester proved that this was

vulnerable by exploiting it. The nikto scan revealed a bash script in the /cgi-bin/status directory. Shown below, Figure 51. This information is needed to exploit the vulnerability.

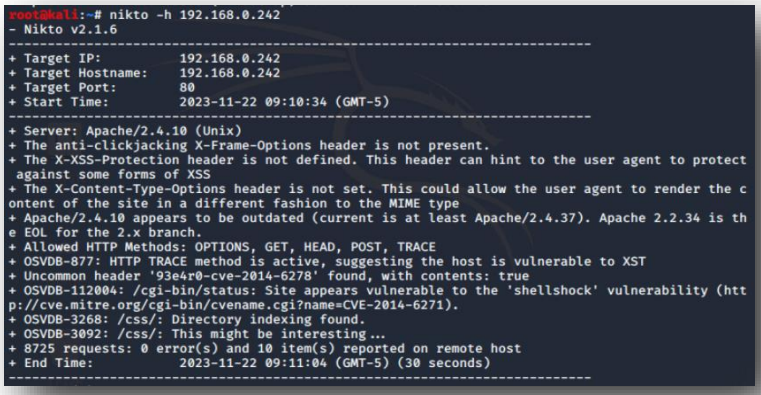


Figure 51:Nikto scan on 192.168.242

The network engineer used Metasploit and the exploit /exploit/multi/apache_mod_cgi_bash_env_exec. The options needing to be set were:

- Rhost = 192.168.0.242
- Targeturi = /cgi-bin/status

Once all options were set the exploit was able to be executed and was successful in obtaining a meterpreter. Shown below, Figure 52.

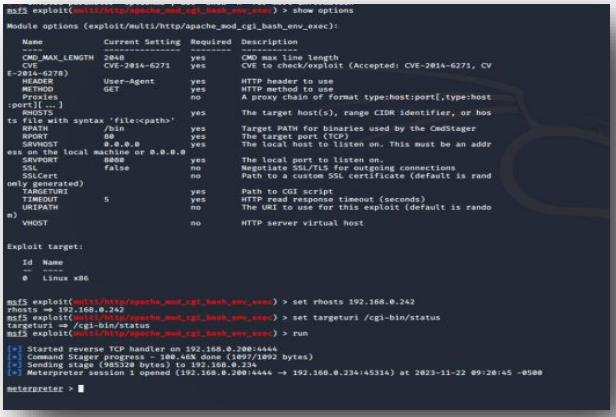


Figure 52:Exploiting using meterpreter

The Nmap scan also revealed that port 22 is open. This allows hackers to attempt to brute force a login. The first attempt was to try the credentials “xadmin” and “plums” however this attempt did not work. Thus, the tester has to identify new credentials. This was achieved using Metasploit and the auxiliary scanner to enumerate SSH usernames based on a wordlist given. The results were

promising as it identified that root user existed. There could possibly be more users, but the wordlist used was not perfect for the situation. However, this username would be ideal as this will give immediate full permissions. Shown below, Figure 53.

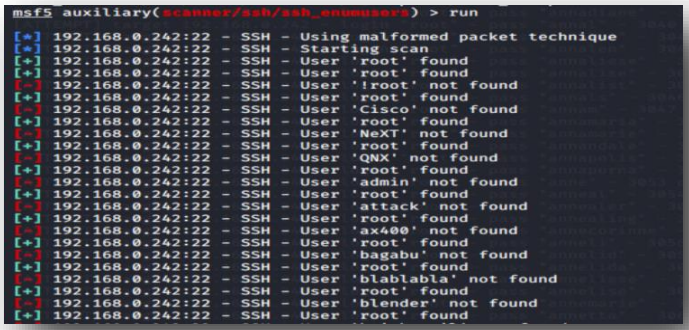


Figure 53:Username enumeration on 192.168.0.242

Given the username, a brute force attempt was then tested. The tool used for this was Hydra rather than Metasploit this was because the protocol used is SSH and this is encrypted, and the command used on hydra appeared to be faster than Metasploit. Furthermore, the process was slower automatically as the traffic is sent through 3 routers, a firewall while also being encrypted via SSH. Shown below, Figure 54.

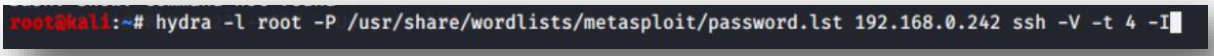


Figure 54:Hydra password attack

The results did take longer than previous attempts as predicted however an insecure password was found. This can be further used to prove if any firewalls exist. As seen below, Figure 55.

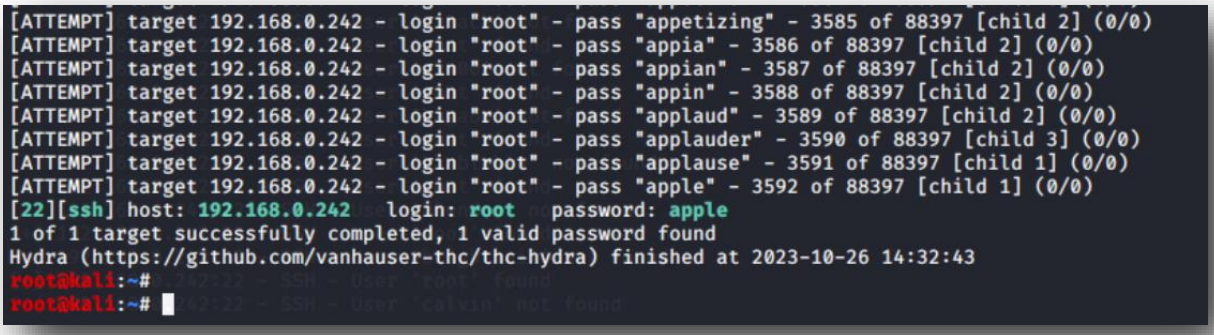


Figure 55:Hydra password identified

Firewall

The device 192.168.0.242 goes through the router 3 on the interface 192.168.0.233 however no other device can be found. This reveals that a firewall is quite possible. To test this theory the tester ran the shellshock exploit again, shown in the previous section. This allowed the examiner to gain a shell and run a ping test on the network from the machine past the 192.168.0.233 interface. This revealed that there were other machines running. As seen below, Figure 56. The machines of interest were as follows;

- 192.168.0.66
- 192.168.0.234
- 192.168.0.241

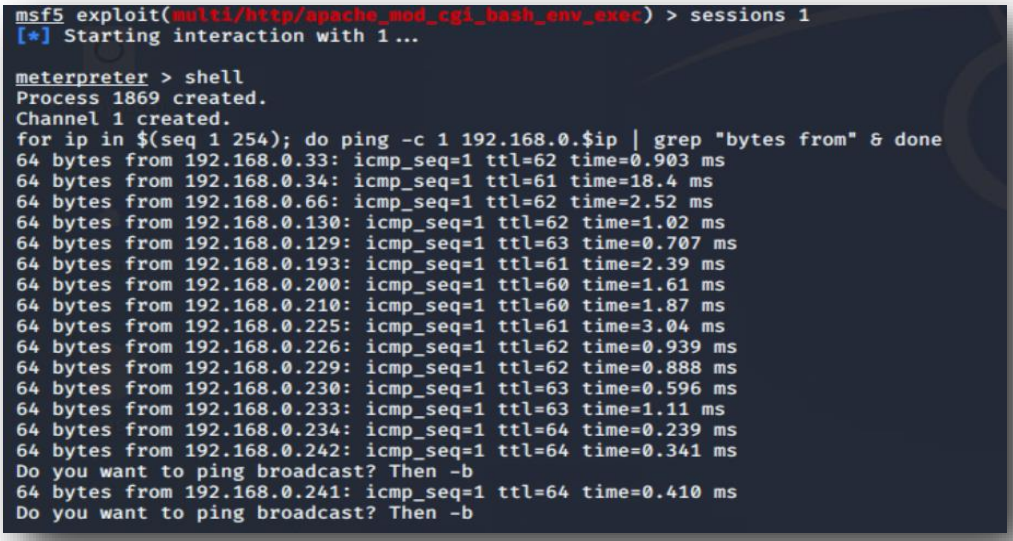


Figure 56:Ping sweep behind the firewall

Using simple subnetting the examiner identified that 192.168.0.241 was the other useable IP on the 192.168.0.240 subnet as there are only two useable addresses on a /30 subnet. The examiner used "auxiliary/scanner/portscan/tcp" payload on Metasploit to scan the active ports on the 192.168.0.241 device. Before running the scan, a route had to be added to tell meterpreter where to send the packets. This was accomplished by using the command "route add Network, Subnet, session" Shown below, Figure 57.

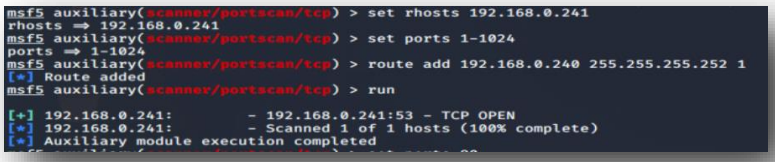


Figure 57:Add route to 192.168.0.241 and run a port scan

This revealed port 53 (DNS) this wasn't as helpful as thought it'd be. Using the same technique, the port scanner was then targeted at 192.168.0.234, This was on the other side of Router3 on interface 2.

Before scanning, the examiner had to direct the traffic to the network 192.168.0.232 through the open session on meterpreter using the route add function. Shown below, Figure 58.

Figure 58:Add a route towards 192.168.0.232 and run a port scan

This revealed that DNS was running on this device as well, but a further port was open. This was port 80 which could possibly be running http. To gain access to this port the examiner had to forward a local port through the meterpreter session to port 80 on the device 192.168.0.234. Shown below, Figure 59.

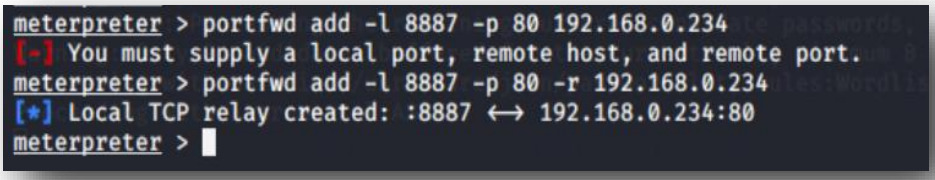


Figure 59:Forward port 80

When the examiner navigated to the localhost on port 8887 using a browser an interesting discovery was made. The IP 192.168.0.234 was an inline firewall as the page was a PFsense community edition login page. Shown below, Figure 60.

Figure 60:Firewall login

During a quick investigation this revealed that PFsense has default credentials which were tested. (Security Space, 2022) This revealed another vulnerability as the default credentials were valid. Discussed later in this report. Once the examiner was logged in this brought up a dashboard, shown below, Figure 61.

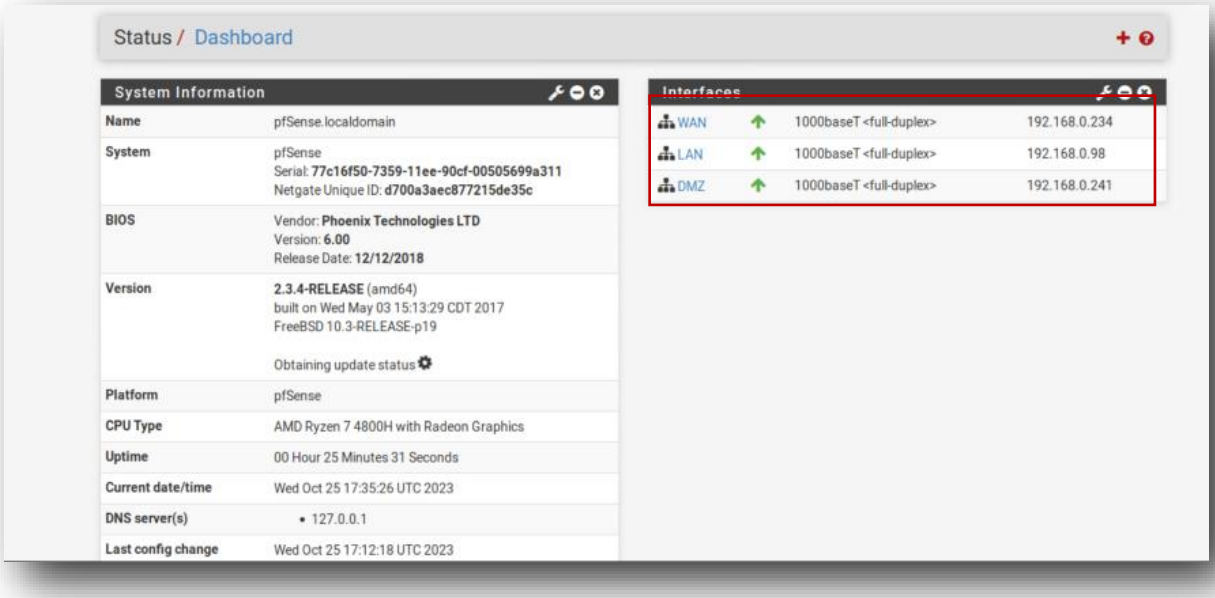


Figure 61:Dashboard for firewall

The examiner used some manual examination to check the tabs and rules to find any useful information. This helped to continue mapping the network as further IP addresses were identified. This can be seen above in the red box. This also revealed their zone.

During the examination the tester was able to find and examine the rules. These weren't as strict as they should've been. Furthermore, the examiner was able to create a new firewall rule and add it above all of the previous rules to allow any traffic through. Figure 62, shows a summary of the old rules for WAN, LAN rules were also applied.

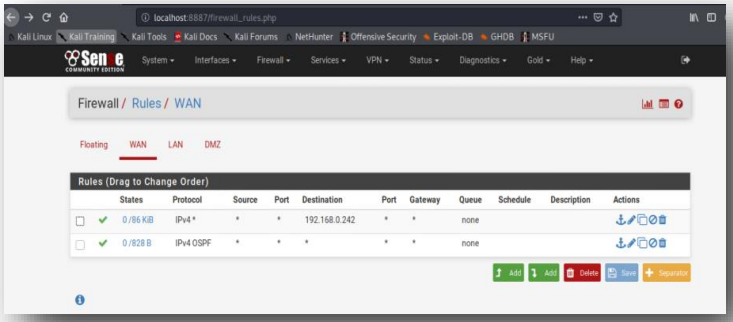


Figure 62:Old WAN rules

Figure 63, shows the new rule that was added to allow traffic through.

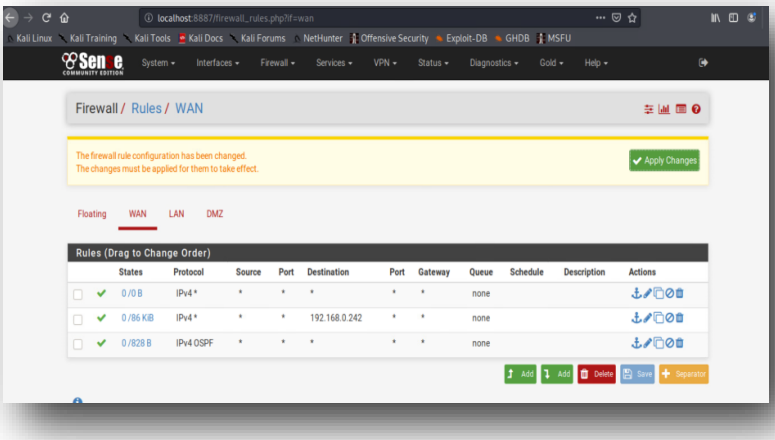


Figure 63:New WAN rules

These actions had rendered the firewall moot. As the examiner could now map the rest of the network through the firewall without any packets being dropped. The security weaknesses and firewall rules will be discussed in security weakness.

This technique generates a lot of noise on the network and there is another technique to perform a more silent attack. This uses the method of tunnelling; a later section will talk about how this can be conducted.

Devices behind the firewall

With the firewall being "practically" disabled this allowed the tester to continue the Nmap scans. The network 192.168.0.240/30 has already been identified as a DMZ with a webserver being hosted in the zone. The other interface on the network was connected to the firewall. A Nmap scan was used for completed thoroughness of the test. The results can be seen in Figure 64.

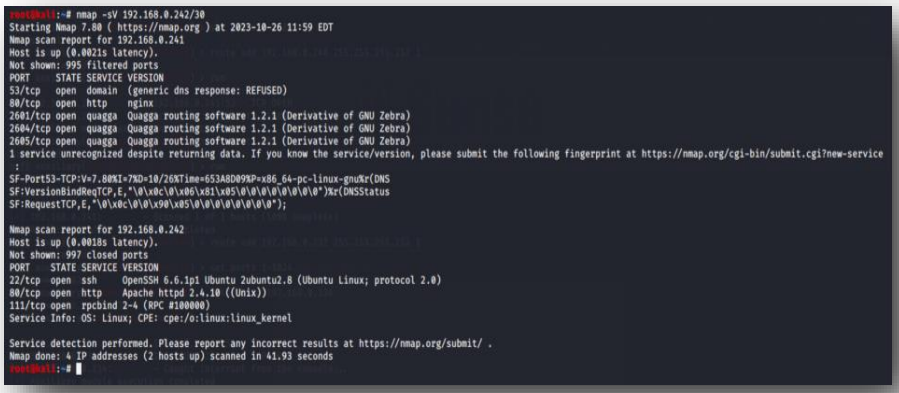


Figure 64:Nmap scan on 192.168.0.242/30

Another redundant scan was done on the network 192.168.0.232/30 for the same reason as before. This revealed the anticipated 192.168.0.233 and 192.168.0.234 IP addresses. Shown below, Figure 65.

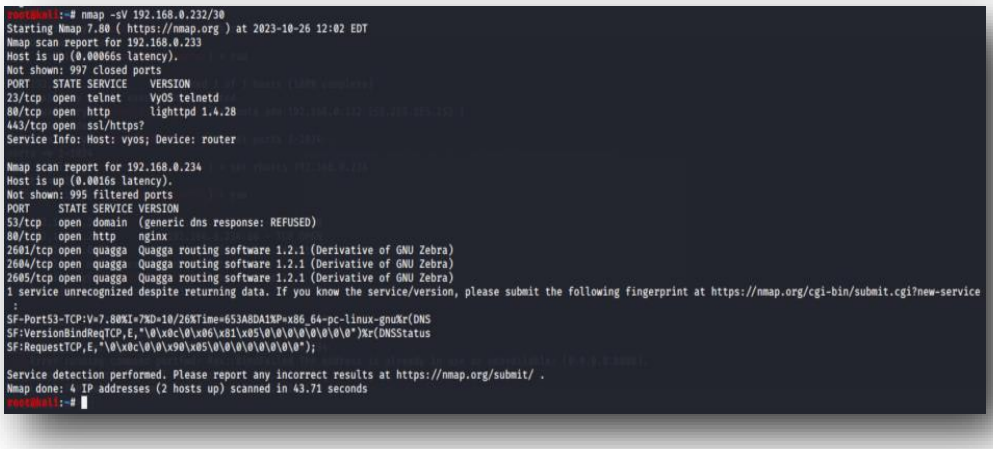


Figure 65:Nmap scan on 192.168.0.232/30

A scan which was based on the network 192.168.0.96/27 found in the IP route command. Which was used at the start of the report was tested as the firewall had an interface connected to this network. (192.168.0.96) This revealed that only one other device was connected to this network, apart from the firewall which was the IP address of 192.168.0.98. The Nmap scan revealed that this was a router. Shown below, Figure 66.

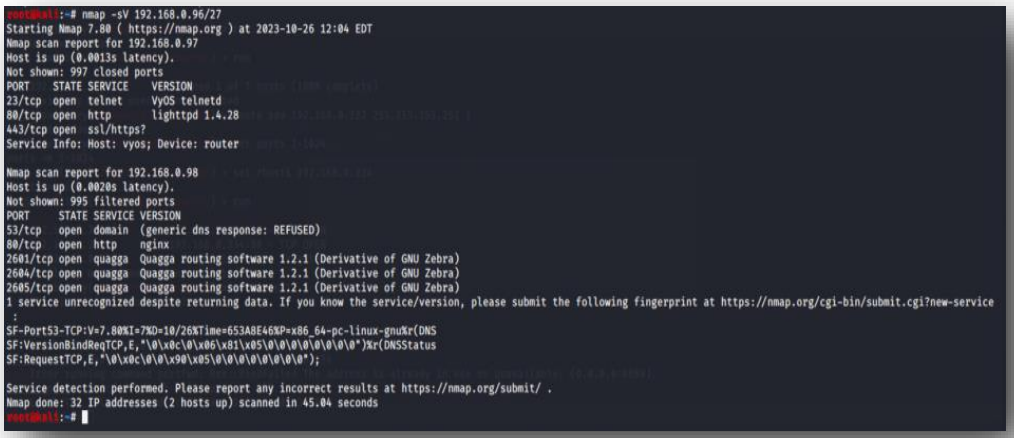


Figure 66:Nmap scan on 192.168.0.96/27

The diagram refers to this router as Router4 as it was the final router found. Shown above is the IP 192.168.0.97 running the VyOS service.

The examiner tested the known default credentials on this router using port 23(Telnet), Unfortunately the same security weakness was also confirmed as these credentials authorised the user. Using the command "show interfaces" while connected to the router revealed that there was still one network past this router. On the network 192.168.0.64/27. Shown below, Figure 67.

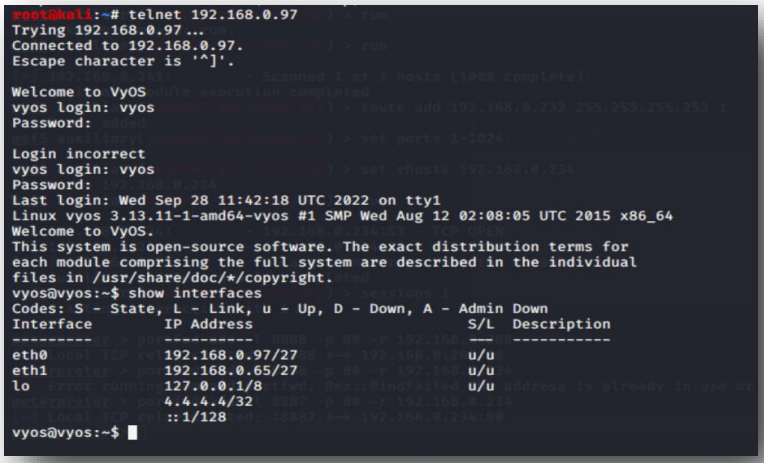


Figure 67:Telnet into 192.168.97 and use show IP interfaces

The final Nmap scan conducted was on the network 192.168.0.64/27. This revealed that only two devices were part of this network, one of which was the router previous found. The new IP address found was 192.168.0.66/27 which appears to be another NFS server. Shown below, Figure 68.

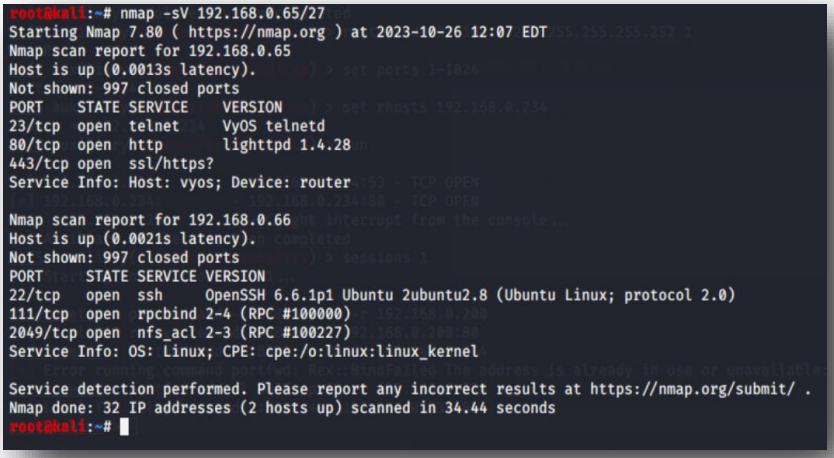


Figure 68:Nmap scan on 192.168.0.65/27

Tunnel Method

This section will discuss another way of manoeuvring around the firewall, without turning the firewall off. This section is more complicated than the previous section and won't include the use of meterpreter.

The first step is to use the protocol SSH to gain access into the machine behind the firewall. The machine behind the firewall is, 192.168.0.242. Once access is gained the hacker must configure the "sshd_config" file to allow tunnelling. Shown in Appendix C, Figure 72 and Figure 73. Allowing the permit tunnel option will allow the hacker to create an encrypted tunnel through the firewall to bypass any rules. The hacker must also restart the service, shown in Appendix C, Figure 74.

Now the hacker is able to create a SSH tunnel using the "-W0:0" flag. Shown in Appendix C, Figure 75. To verify that the machine has an open tunnel, the hacker used the "ip addr" command to check if a tunnel was in operation. This can be seen in Appendix C, Figure 76.

The tunnel is there but not in the up state, this is due to no connection at the other side and the link is also not configured to be up. To configure this the hacker must create an IP address for this interface and set the link to the up state. Furthermore, the hacker had to allow port forwarding. These configurations can be seen in Appendix C, Figure 77.

An IP had to be set on the Kali machine as well. The IP address had to be on the same subnet and the device had to be set to the same interface, tun0 in this case. The hacker used the "ip addr" command to verify that the tunnel is up. This can be seen in Appendix C, Figure 78.

Before creating any routes to devices behind the firewall the hacker had to edit iptables to allow NAT as the IP addresses aren't registered on any of the routers. This is shown in Appendix C, Figure 79. Once IP tables were configured the hacker could add a route to an IP behind the firewall. The IP address most desired is the 192.168.0.234 as this is the other side of router 3. Using the command "route add -host <IP> tun0" this allowed the hacker to add the route. Shown in Appendix C, Figure

80. To verify that this worked a ping command was issued, which was successful. Using Nmap the hacker was able to map ports on this valid IP address. Shown In Appendix C, Figure 81.

Port 80 was found to be open. This indicated a website, using a browser the hacker was able to browse this page. As suspected, this was in fact the interface of the firewall found previously. Shown in Appendix C, Figure 82. An Nmap scan was conducted on the 192.168.0.234/30 network to finally prove that only two IP address are valid. Shown in Appendix C, Figure 83.

Another Nmap scan was conducted on the 192.168.0.240/30 network, this revealed that 192.168.0.241 was another interface of the firewall and that 192.168.0.242 was the webserver. This concluded this subnet investigation. Shown in Appendix C, Figure 84.

Navigating to the firewall on the 192.168.0.234 interface and using the credentials found previously (pfsense), allowed the hacker to verify the next interface and IP addresses to scan (Shown in Appendix C, Figure 85) which was in the 192.168.0.98/30 subnet. According to the "show ip route" command used earlier on in this report. However, the device 192.168.0.242 could not access that network. The hacker crafted a ping sweep to check for valid IP addresses. Shown in Appendix C, Figure 86 and Figure 87. This revealed the IP address 192.168.0.66 was active. To allow the Kali machine access to this machine, a route had to be added. Shown in Appendix C, Figure 88. This allowed access to the IP address as a successful ping was achieved. trying the same method on the other IP address was regrettably a waste of time, as none were reachable, shown in Appendix C, Figure 89.

The hacker used Nmap to once again map out the ports used by 192.168.0.66 this can be seen in Appendix C, Figure 90. This revealed vulnerable ports which can be used to gain access to the machine. Discussed in section 192.168.0.66(Server6).

Once access was gained the hacker attempted to form a tunnel, this was denied presumably because of the "sshd_config" file was ill configured for tunnelling. Shown in Appendix C, Figure 100. To solve this issue the configuration file had to be re-configured as well as the service restarted. These can be seen in Appendix C, Figure 101 and Figure 102. The attempt to gain a tunnel was successful this time around. The flag "-W1:1" had to be used as tunnel 0 was already in use. Show in Appendix C, Figure 103.

To enable the tunnel the same process as before had to be conducted.

1. Set IP address for the specific tunnel.
2. Turn the link to the up state
3. Configure the IP tables
4. Enable port forwarding.

These can be seen in Appendix C, Figure 104. Steps 1 and 2 needs to be conducted on the Kali machine as well. Once the settings were configured the ping command was issued to test connectivity. Shown in Appendix C, Figure 105.

The device 192.168.0.97 was unable to be pinged from the 192.168.0.242 machine, this was the first device to be examined. The route had to be added as seen in Appendix C, Figure 106. The next step was to test with a ping command which was successful, see Appendix C, Figure 107.

The next two devices still needing to be added to the routing table are, 192.168.0.65 and 192.168.0.98, once these were added the ping command was issued again to test connectivity. Shown in Appendix C, Figure 108. The test was successful and this proved that the network behind the firewall has been compromised.

Several Nmap scans were then used to verify no further devices were used the first was for the network 192.168.0.64/27, two devices were identified shown below and in Appendix C, Figure 109.

- 192.168.0.65
- 192.168.0.66

One appeared to be a router, and this was verified using the same technique earlier. telnetting into the port and using show interfaces. Shown in Appendix C, Figure 110. This provided the information to deem no further networks are on this internet.

Another Nmap scan was used for the network 192.168.0.97/27, this revealed the same two IP addresses as found previously. Shown in Appendix C, Figure 111. To prove that 192.168.0.98 was the LAN connection of the firewall a browser was used to navigate to port 80. Shown in Appendix C, Figure 112.

192.168.0.66(Server6)

To gain access to the machine 192.168.0.66 the hacker had to use ports 22, 111 and 2049. The first step was to identify what was shared. Shown in Appendix C, Figure 91. The root directory was shared. This is a major vulnerability.

The hacker wanted consistent access as the root account to continue mapping the network. This was easily done by copying the RSA key but before this was enabled some things had to be set. The first was that authorised keys had to allowed. This was achieved by editing the "sshd_config" file on the public share. Shown in Appendix C, Figure 92

From this the hacker identified that the device allowed root login and from the authorised keys. The hacker had to generate RSA keys, shown on Appendix C, Figure 93.

The next set of steps are as follows;

1. Verify generation, shown in Appendix C, Figure 94.
2. Copy the public key to the target at the /root/.ssh/ directory, shown in Appendix C, Figure 95.
3. Navigate to the directory, shown in Appendix C, Figure 96.
4. Rename the public key to the authorised keys, shown in Appendix C, Figure 97.
5. Verify that the key works, shown in Appendix C, Figure 98.

This proves that the machine is vulnerable as access was gained. Using the "ifconfig" command this provided information to show that no further devices are passed 192.168.0.66. Shown in Appendix C, Figure 99.

Security weakness

Overview

To conclude the evaluation of the vulnerability analysis, the devices within the network appear to be highly vulnerable. The majority of the vulnerabilities are due to poor configuration or staff training, furthermore, some software is outdated causing some common vulnerabilities. During this security weakness section, each identified vulnerability will be mentioned and analysed. Additionally, the overall risk of the vulnerability will be discussed and how to mitigate the risk.

Outdated software

Outdated software can have severe consequences as vulnerabilities that are publicly accessible can be used to exploit the system and, in some cases, gain a shell due to the outdated software. Some of the software used in this network was found to be outdated. A further analysis identified some common vulnerabilities these were revealed through the website Mitre, See **Error! Reference source not found.**.

OpenSSH 6.6.1p1 is vulnerable to a variety of common vulnerabilities (60 vulnerabilities) which range from low to high, with one vulnerability being critical. This critical vulnerability is referred to as CVE-2023-38408. This vulnerability is caused by an untrustworthy file path which could lead to remote code execution. See **Error! Reference source not found.**.

Apache 2.2.22 has 44 weaknesses which have been published from 2006 to 2021. These vulnerabilities have been rated from low to hight according to Mitre. Furthermore, another version of Apache was identified. Version 2.4.10 which has 60 recorded vulnerabilities, rated from low to high once again. See **Error! Reference source not found.**.

Lighttpd, running the version 1.4.28 has 34 common vulnerabilities according to the Mitre website. The risk of these ranges from low to high.

The above technologies have numerous weaknesses, and these were found on several machines throughout the network.

Shellshock vulnerability

Shellshock vulnerability is also known as bashdoor. This vulnerability occurs in a bash shell which allows execution of malicious code through the environment. This type of vulnerability is another critical vulnerability that has been identified on Server 5. The script found which was used was in the location `/cgi-bin/status`.

Mitigations

To resolve these issues the only solution is to maintain the software and update the software when new patches and versions are released.

Further mitigation for shellshock

As consistent patching can be strenuous, a few other methods could be implemented to further protect against shellshock. These include correct user sanitisation; this prevents hackers from

escaping the shell's environment. Finally monitoring logs for indication of attempted or successful command execution. (Gergen, 2014)

Weak and default credentials

Accounts and logins were identified to be vulnerable to brute force attempts due to weak or default credentials. This essentially would allow hackers ease to the network as obtaining these credentials weren't much effort. This vulnerability is quite critical as no further authentication on any device was identified. Additionally, no lockout mechanism was found which theoretically would allow an unethical hacker to consistency attempt password attempts until access is gained.

The passwords found on the network that are vulnerable are shown below:

- Vyos (Default credentials)
- Pfsense (Default credentials)
- Plums (Weak credential)
- Apple (Weak credentials)
- !gatvol (Weak credentials)
- Zxc123 (Weak credentials)

Weak credentials, they are passwords and usernames which are guessable, short, non-complex, and common. These passwords are dangerous because hackers could brute force the passwords, shown throughout the report.

The term, "default credentials" is used for credentials which can be obtained online, as they weren't changed prior to production. These are seriously dangerous as the hacker will immediately gain access without having failed attempts, which could've be logged and stopped.

Mitigation

Enforce a password policy which ensures passwords are longer than 8 characters, that includes complexity ($£%^&*!?) and numerical characters. Although this will slow down hackers it won't stop them. To resolve the issue of brute force attacks, lockouts should be utilized. With a strong password policy, the lockout mechanism could allow somewhere between 10 attempts. Furthermore, monitoring the password attempts using a SOC analyst would be highly beneficial as real time attacks could be stopped.

Changing all default credentials is a must and should be done before a production environment.

The final issues were the use of the same password throughout the full network. Each password for each device should be different. This is to ensure a malicious hacker is not able to jump from machine to machine with the use of one password. This comes to staff training which is highly important. The only real way of securing a network is to have competent staff via training. This will ensure that they use secure passwords and do not use the same password for other devices.

NFS shares

NFS (Network file system) is a service that runs on servers that allow files to be shared to devices on the network. If the NFS server is ill configured this can lead to a data leak of sensitive files. Furthermore, the sensitive files could be altered if the set up allows this. (poor configuration) This

damages the integrity of the service and files. Causing data loss, data leaks and possibly full control of the service, as shown in the report.

During the test Servers 1, 3, 4 and 6 were all using NFS. Servers 1 and 6 were sharing the full root directory which allowed access to the passwords, RSA keys, and files of these devices, This is highly insecure as this allows for hackers to gain access to the device with the use of RSA keys or passwords.

Server 3 only allowed access to "xadmin's" directory. This allowed hackers to steal the private key as it was openly stored on "xadmin's" directory in .ssh. This ultimately weakened the full network as the tester was able to gain access Server 4 after obtaining the private key.

Server 4 only allows access to "xadmin's" directory this is not as insecure but allowed hackers to identify that a device on the network had direct access to server 4.

Mitigation

To secure NFS the use of proper configuration is mandatory. The NFS service should be updated to the last stable version. Once the version is stable there is a list of good practices.

- Encrypt the NFS traffic to ensure the traffic cannot be seen and stolen
- Use VLANs to ensure only the correct IP addresses can access this service. This would've ultimately prevented the kali machine gaining access.
- Secure firewall rules to prevent NFS traffic leaving a secure LAN
- Configuration of permissions set correctly
    - Authorized use only.
        - Set secure passwords
    - Specific files or folders
- Monitoring the NFS traffic and audit.

Once all of these are implemented the NFS service will becomes more robust.

RSA certificates

The use of RSA keys did not prove to be vulnerable by itself however, the private key was easily found and forged. This is a reason on why the testers was able to gain access to several accounts.

Mitigation

Store the private key in a safe and secure manner and not shared on NFS.

WordPress

During the test the network engineer noticed that Server 2 was running a website that was created and maintained using the application WordPress. The version that was running was outdated. Furthermore, the default username was not changed, and the password set for the application was weak.

This allowed for the tester to gain access as an administrator using the easily obtainable credentials and upload a reverse shell script which ultimately allows the hacker to gain a shell.

Mitigation

To secure WordPress there are a few steps:

- Update the outdated version to a stable version
- Use strong passwords
- Change the username from “admin” to remove default credentials
- Limit the amount of login attempts
- Enable multi-factor authentication
- Monitor the login attempts to detect real time attacks
- Remove unnecessary themes and plugins, this prevents any known vulnerabilities being used to exploit the application

Cryptography (HTTP & Telnet)

HTTP stands for hypertext transfer protocol and is used to transfer website data to clients. This is fine however HTTP does not encrypt any data for transfer which hold a vulnerability as data could be intercepted and stolen or even changed. This could damage the integrity of the network data. Additionally, Telnet operates as the same as SSH, but Telnet does not encrypt any traffic. This is similar to the weakness in HTTP. As this vulnerability could allow a hacker to intercept the login request for any router and steal the login credentials and then use the same credentials to access all the routers.

Throughout the test HTTP and Telnet were constantly used. HTTP was used for website traffic; the protocol was identified on Server 2 and 5. Server 2 had port 443 which is an encrypted version for HTTP. Server 5 however, only operates on port 80 using HTTP. Telnet is in operation for all 4 routers, these are all proven to be vulnerable. Traffic sent to and from these devices could be copied or even changed by a malicious hacker. These types of attacks are commonly known as man-in-the-middle attacks.

Mitigations

To secure the network from unencrypted traffic its best to adopt different and more secure protocols. Establishing a connection with HTTP is fine but once the connection is established the protocol used after should be HTTPS as this will encrypt traffic using TLS (Most current encryption) This will prevent hackers from reading information they have intercepted.

The protocol Telnet is depreciated and has been replaced with SSH. This is because SSH allows for a more secure type of authentication and once the authentication has been approved all traffic sent back and forth is encrypted similar to HTTPS.

Firewall rules

The network contained a Pfsense firewall. This firewall added a certain level of security by preventing access to some machines behind it from the DMZ and LAN interface, however the rules for the firewall were ill-configured. This is due to the firewall allowing access from the DMZ to the WAN allowing the tester to gain access to the firewall interface from Server 5. Another concern is the rules set for the DMZ zone. The rules for outside machines to connect to Server 5 were quite

loose. These rules allowed any type of traffic on any port through to Server 5. This is generally a bad practice.

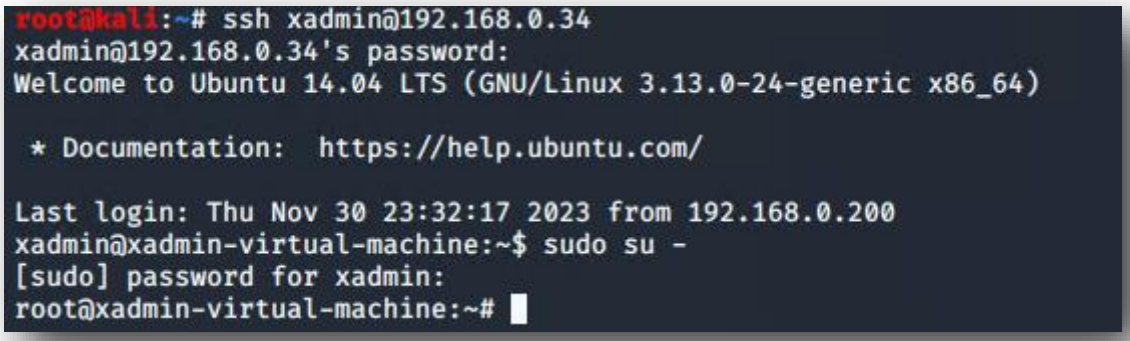
Mitigations

To resolve these issues, rules should be configured securely, only allowing resources to devices which need them (least privilege). An important rule which will prevent access to the firewall interface from the DMZ would be blocking all the traffic from the DMZ to the WAN, apart from mandatory protocols.

Furthermore, only allowing specific traffic to Server5 in the DMZ is important. An example of this would be; only allowing port 80 traffic through as this is all that is needed for a webserver. This would've prevented the tester, using SSH to breach the firewall.

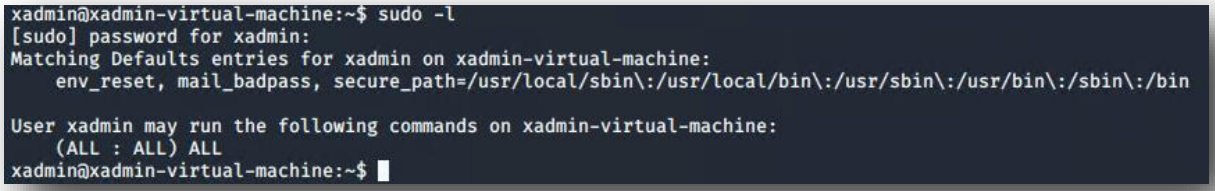
Improper SUDO permissions

The configuration of the /etc/sudoer file was misconfigured. The file allowed for any user to execute commands with the use of the current user's password as a super user. A super user allows for processes or programs to be executed as a privileged user. This is highly insecure as this allows unprivileged users to gain access to a root user through the command "sudo su -" As shown below.



SUDO 1

The configuration of the file can be seen below.



SUDO 2

This is a major security risk as all the machines on the network allow regular users to elevate to privileges users, without the use of correct authentication. This ultimately weakens the full network.

Mitigations

To resolve this issue the best practice is to correctly configure the /etc/sudoer file. The best practice is to disable the ability to run programs as root with the use of "sudo". Additionally, the password needed to execute a "sudo" command should be that of a privileged user, not the current user.

Network critical evaluation

Overview

The network was deemed to be poor and highly vulnerable. The network could be improved by implementing VLANS and better use of subnets to separate resources, improvements for fault tolerance and firewall security.

Subnet evaluation

The subnets created for this network is sufficient but impractical for expansion and further security concerns exist from IP addressing.

The majority of subnets use the /27 prefix, this is poorly configured, this is because majority of the subnets only use 2 IP addresses, apart from the subnet which includes the kali machine, this leaves 28 unused IP addresses on each subnet. This reduces the available networks. This can be improved upon by utilizing better subnetting.

An example of an improved version would be the use of 192.168.0.X/26 This will create 4 subnets two of them, will be from 192.168.0.0 to 192.168.0.127, these could be reserved for later use, while 192.168.0.128 to 223 could be further submitted into a /28 which will allow for 6 subnets, each with 14 useable hosts, more than enough for the current 2 and future expansion. The final addresses from 192.168.0.224 to 192.168.0.255 could be further submitted with the /30 prefix to create enough subnets for serial links with extra for expansion. This will heavily increase the reserved IP addresses for future use. This example might not be perfect, but this entirely depends on the network needs.

Additionally, most of the IP addresses are guessable, this is because when a device is assigned an IP address, the previous engineer assigned the first available address. This could potentially allow a hacker to guess used IP addresses. To better improve IP assignation, would be implementation of random IP allocation. The protocol DHCP would be ideal for this.

Firewall improvements

Although the rules were improper and could be dramatically improved, further firewalls could also prevent cyber breachers. As a web server was present in the DMZ, a website application firewall (WAP) could be utilized to improve web security. Furthermore, an internal firewall could exist after the main boarder firewall to further protect 192.168.0.66. Although these will improve security its paramount to correctly configure the rules.

Improved fault tolerance

The current network does not appear to follow any specific network topology, from the diagram it appears to contain several star topologies with the use of routers. Subnet 7 follows this approach with the use of a switch. The only connection, which does not follow this approach is subnet 2 and 10, this is connected using a point-to-point connection. The network appears to be acceptable at this current rate, as the network is quite small. However, if a connection between the routers goes down the network will be split.

This could be improved by increasing fault tolerance and redundancy. If possible, a switch or a router should connect router 1 with router 3 this will create a more reliable network, because if any connection between any of the routers outside the firewall fail, there will be an alternative route to send traffic.

Security evaluation

The security of the network mostly depends on the security of the devices on the network. This has been discussed to be quite poor. This network if targeted would indeed be breach by any competent hacker. This report has also mentioned how to resolve the issues and prevent a cyber breach.

To enhance the security to be superior there is two types of methods which could be implemented to secure the network further. The first is proper configuration of a switch. The switch on subnet 7 allowed for any device to connect to the network without authentication. This ultimately allowed the engineer to connect and start attacking. Ports on the switch should be set to up only when they are being used by an authenticated device.

The last method is the use of IPS (Intrusion prevention system), alternatively IDS (Intrusion detection system) could be used if cost is an issue. However, IPS are far superior as these can prevent attacks. The choice between these devices also relies on how much traffic goes around. As the network appears small, the recommendation would be an IPS. If the network suffers from large amounts of traffic this an IDS would be more beneficial as this won't slow the network down.

These devices will increase the security of the network as real time attacks can be identified before they become a serious issue, the use of an IPS could also prevent the attack. The placement of this device should be at the entrance of the network. Either inline, with the use of an IPS to prevent real time attacks, or if an IDS is used then the traffic should be mirrored and sniffed

Conclusion

To conclude, the network in question has some strengths, for example the firewall does stop traffic from accessing 192.168.0.66 and the network topology is good for the current needs of the network. However, there is several issues which makes this network vulnerable.

The report has outlined several issues about this network and has also discussed solutions to mitigate the risks and secure the network. If the recommendations are adhered to, the network's security will greatly improve. ACME Inc should be aware that no matter how secure a network is there is always a chance of a cyber breach and implementing a cyber resilience plan, will always be recommended to ensure attacks are kept to a minimum. Furthermore, although the solutions in this report will increase the security, the network will need monitored and audited correctly to ensure real time attacks are detected and stopped. This requires competent personnel and great staff training.

Appendices

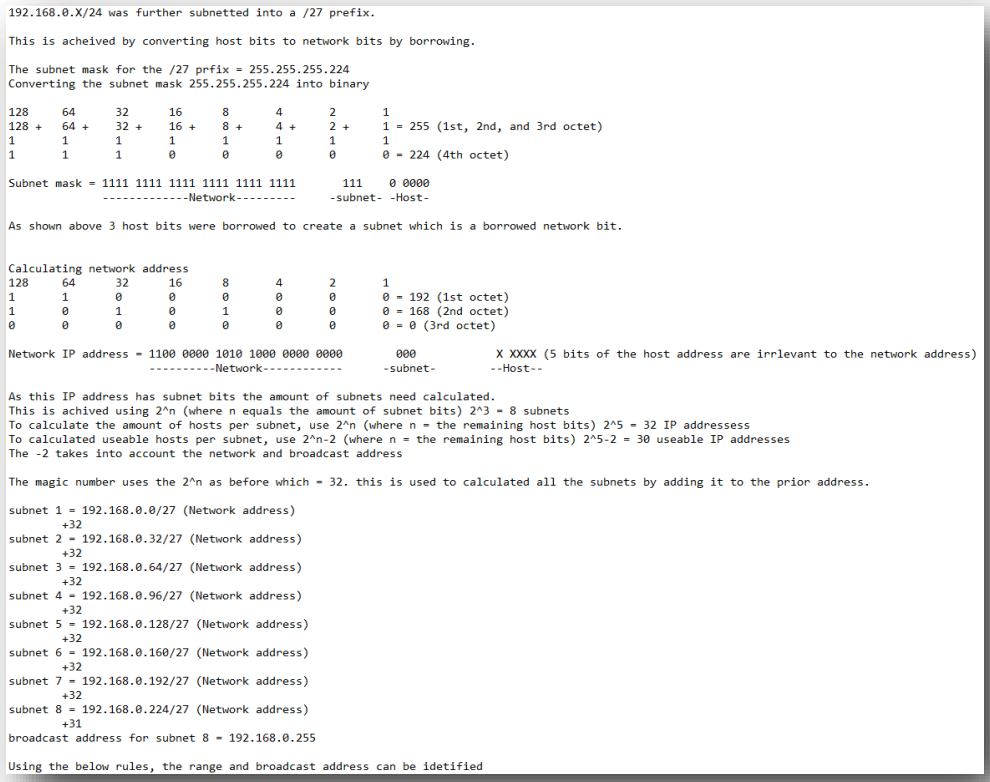
Appendix A (Subnetting)

```
During the test, 3 major IP addresses were found:

13.13.13.X/24, /24 = a subnet mask of 255.255.255.0
172.16.221.X/24, /24 = a subnet mask of 255.255.255.0

192.168.0.X/24
This IP address was further subnetted into 192.168.0.X/27 then into 192.168.0.X/30

The equation 2^n had to be used to calculate the number of avalible subnets, hosts and (2^n-2 for useable hosts).

Calcualating 13.13.13.X/24 network.

converting the subnet mask 255.255.255.0 into binary

128     64      32      16      8       4       2       1
128 +   64 +    32 +    16 +    8 +     4 +     2 +     1 = 255 (1st, 2nd, and 3rd octect)
1       1       1       1       1       1       1       1
0       0       0       0       0       0       0       0 = 0 (4th octect)

Subnet mask = 1111 1111 1111 1111 1111 1111 0000 0000
              -------------Network--------- ---Host--

128     64      32      16      8       4       2       1
0       0       0       0       1       1       0       1 = 13

Network IP address = 0000 1101 0000 1101 0000 1101 XXXX XXXX (4th octect is irrevant as this refers to the host)
                     ----------Network------------ --Host---

Calculating the amount of hosts on the 13.13.13.X/24 network

2^n (Where n = host bits) = 2^8 = 256 host IP addresses
2^n-2 (Where n = host bits) = 2^8-2 = 254 useable host IP addresses
As /24 is a class C IP address no subnet calculations are needed.

Using the magic number which is calculated using 2^n (Where n = host bits) = 2^8 = 256
This will be used to calculate the range, network address and broadcast address.

The network address is needed first. This is achived by turing all of the hosts bits to 0, as shown below.

0000 1101 0000 1101 0000 1101 0000 0000 = 13.13.13.0

Network address = 13.13.13.0
First useable address = 13.13.13.1 (Add one to the network address)
Broadcast address = 13.13.13.255 (Add magic number to network address 0 + 256 = 255 (0 includes a valid IP address))
Last useable address = 13.13.13.254 (minus 1 from broadcast address, 255 - 1 = 254)
IP range = 13.13.13.1 to 13.13.13.254
```

Subnet calculations 1

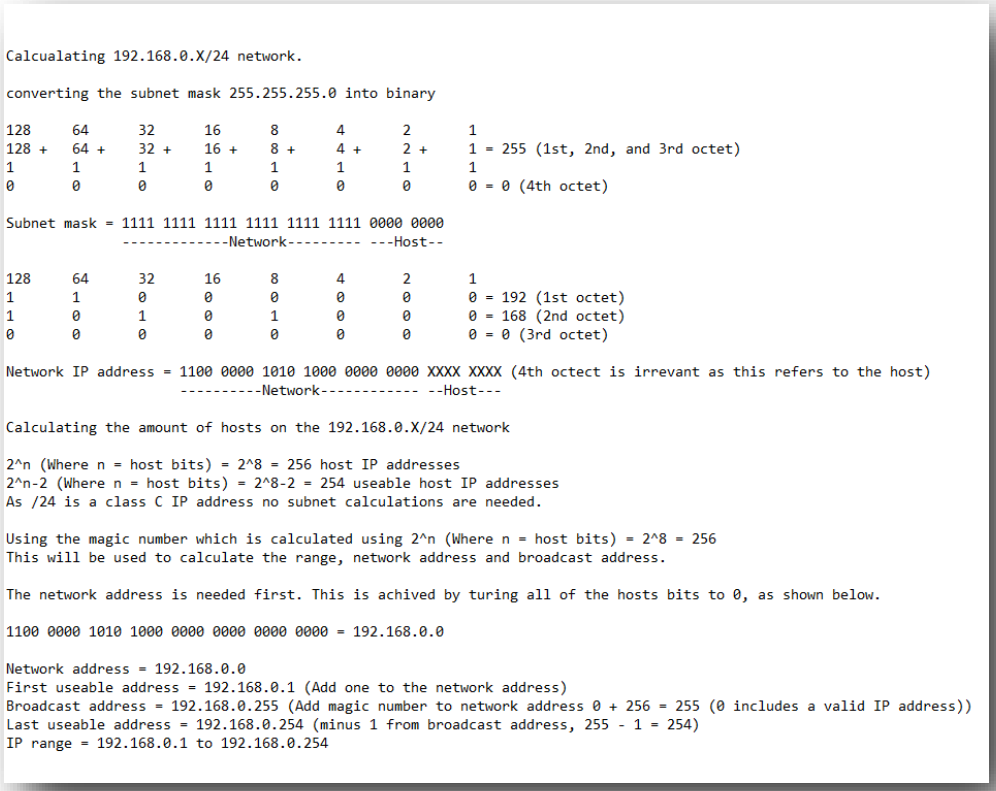
```
Calcualating 172.16.221.X/24 network.

converting the subnet mask 255.255.255.0 into binary

128     64      32      16      8       4       2       1
128 +   64 +    32 +    16 +    8 +     4 +     2 +     1 = 255 (1st, 2nd, and 3rd octet)
1       1       1       1       1       1       1       1
0       0       0       0       0       0       0       0 = 0 (4th octet)

Subnet mask = 1111 1111 1111 1111 1111 1111 0000 0000
              -------------Network--------- ---Host--

128     64      32      16      8       4       2       1
1       0       1       0       1       1       0       0 = 172 (1st octet)
0       0       0       1       0       0       0       0 = 16 (2nd octet)
1       1       0       1       0       0       0       1 = 221 (3rd octet)

Network IP address = 1010 1100 0001 0000 1101 0001 XXXX XXXX (4th octect is irrevant as this refers to the host)
                     ----------Network------------ --Host---

Calculating the amount of hosts on the 172.16.221.X/24 network

2^n (Where n = host bits) = 2^8 = 256 host IP addresses
2^n-2 (Where n = host bits) = 2^8-2 = 254 useable host IP addresses
As /24 is a class C IP address no subnet calculations are needed.

Using the magic number which is calculated using 2^n (Where n = host bits) = 2^8 = 256
This will be used to calculate the range, network address and broadcast address.

The network address is needed first. This is achived by turing all of the hosts bits to 0, as shown below.

1010 1100 0001 0000 1101 0001 0000 0000 = 172.16.221.0

Network address = 172.16.221.0
First useable address = 172.16.221.1 (Add one to the network address)
Broadcast address = 172.16.221.255 (Add magic number to network address 0 + 256 = 255 (0 includes a valid IP address))
Last useable address = 172.16.221.254 (minus 1 from broadcast address, 255 - 1 = 254)
IP range = 172.16.221.1 to 172.16.221.254
```

Subnet calculations 2



```
192.168.0.X/24 was further subnetted into a /27 prefix.

This is acheived by converting host bits to network bits by borrowing.

The subnet mask for the /27 prfix = 255.255.255.224
Converting the subnet mask 255.255.255.224 into binary

128     64      32      16      8       4       2       1
128 +   64 +    32 +    16 +    8 +     4 +     2 +     1 = 255 (1st, 2nd, and 3rd octet)
1       1       1       1       1       1       1       1
1       1       1       0       0       0       0       0 = 224 (4th octet)

Subnet mask = 1111 1111 1111 1111 1111 1111       111    0 0000
              ------------Network---------     -subnet- -Host-

As shown above 3 host bits were borrowed to create a subnet which is a borrowed network bit.


Calculating network address
128     64      32      16      8       4       2       1
1       1       0       0       0       0       0       0 = 192 (1st octet)
1       0       1       0       1       0       0       0 = 168 (2nd octet)
0       0       0       0       0       0       0       0 = 0 (3rd octet)

Network IP address = 1100 0000 1010 1000 0000 0000        000           X XXXX (5 bits of the host address are irrlevant to the network address)
                     ----------Network-----------      -subnet-        --Host--

As this IP address has subnet bits the amount of subnets need calculated.
This is achived using 2^n (where n equals the amount of subnet bits) 2^3 = 8 subnets
To calculate the amount of hosts per subnet, use 2^n (where n = the remaining host bits) 2^5 = 32 IP addressess
To calculated useable hosts per subnet, use 2^n-2 (where n = the remaining host bits) 2^5-2 = 30 useable IP addresses
The -2 takes into account the network and broadcast address

The magic number uses the 2^n as before which = 32. this is used to calculated all the subnets by adding it to the prior address.

subnet 1 = 192.168.0.0/27 (Network address)
        +32
subnet 2 = 192.168.0.32/27 (Network address)
        +32
subnet 3 = 192.168.0.64/27 (Network address)
        +32
subnet 4 = 192.168.0.96/27 (Network address)
        +32
subnet 5 = 192.168.0.128/27 (Network address)
        +32
subnet 6 = 192.168.0.160/27 (Network address)
        +32
subnet 7 = 192.168.0.192/27 (Network address)
        +32
subnet 8 = 192.168.0.224/27 (Network address)
        +31
broadcast address for subnet 8 = 192.168.0.255

Using the below rules, the range and broadcast address can be idetified
```

Subnet calculations 3



```
Calcualating 192.168.0.X/24 network.

converting the subnet mask 255.255.255.0 into binary

128     64      32      16      8       4       2       1
128 +   64 +    32 +    16 +    8 +     4 +     2 +     1 = 255 (1st, 2nd, and 3rd octet)
1       1       1       1       1       1       1       1
0       0       0       0       0       0       0       0 = 0 (4th octet)

Subnet mask = 1111 1111 1111 1111 1111 1111 0000 0000
              ------------Network--------- ---Host--

128     64      32      16      8       4       2       1
1       1       0       0       0       0       0       0 = 192 (1st octet)
1       0       1       0       1       0       0       0 = 168 (2nd octet)
0       0       0       0       0       0       0       0 = 0 (3rd octet)

Network IP address = 1100 0000 1010 1000 0000 0000 XXXX XXXX (4th octect is irrevant as this refers to the host)
                     ----------Network------------ --Host---

Calculating the amount of hosts on the 192.168.0.X/24 network

2^n (Where n = host bits) = 2^8 = 256 host IP addresses
2^n-2 (Where n = host bits) = 2^8-2 = 254 useable host IP addresses
As /24 is a class C IP address no subnet calculations are needed.

Using the magic number which is calculated using 2^n (Where n = host bits) = 2^8 = 256
This will be used to calculate the range, network address and broadcast address.

The network address is needed first. This is achived by turing all of the hosts bits to 0, as shown below.

1100 0000 1010 1000 0000 0000 0000 0000 = 192.168.0.0

Network address = 192.168.0.0
First useable address = 192.168.0.1 (Add one to the network address)
Broadcast address = 192.168.0.255 (Add magic number to network address 0 + 256 = 255 (0 includes a valid IP address))
Last useable address = 192.168.0.254 (minus 1 from broadcast address, 255 - 1 = 254)
IP range = 192.168.0.1 to 192.168.0.254
```

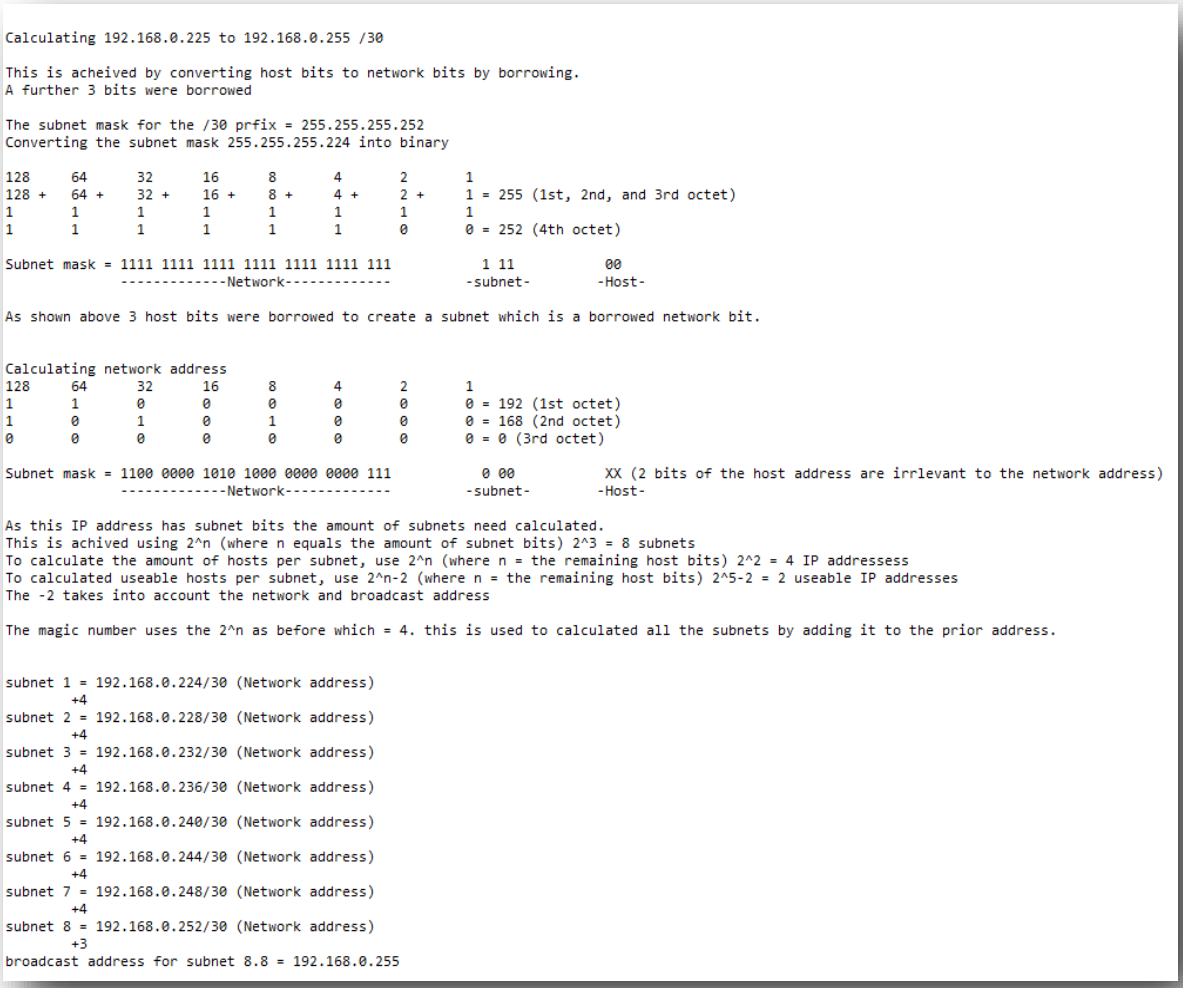
Subnet calculations 4

```
1st subnet
Network address = 192.168.0.0
First useable address = 192.168.0.1 (Add one to the previous network address)
Broadcast address = 192.168.0.31 (Minus one from the next subnets network address)
Last useable address = 192.168.0.30 (minus 1 from broadcast address, 31 - 1 = 30)
IP range = 192.168.0.1 to 192.168.0.30

2nd subnet
Network address = 192.168.0.32
First useable address = 192.168.0.33
Broadcast address = 192.168.0.63
Last useable address = 192.168.0.62
IP range = 192.168.0.33 to 192.168.0.62

3rd subnet
Network address = 192.168.0.64
First useable address = 192.168.0.65
Broadcast address = 192.168.0.95
Last useable address = 192.168.0.94
IP range = 192.168.0.65 to 192.168.0.94

4th subnet
Network address = 192.168.0.96
First useable address = 192.168.0.97
Broadcast address = 192.168.0.127
Last useable address = 192.168.0.126
IP range = 192.168.0.97 to 192.168.0.126

5th subnet
Network address = 192.168.0.128
First useable address = 192.168.0.129
Broadcast address = 192.168.0.159
Last useable address = 192.168.0.158
IP range = 192.168.0.129 to 192.168.0.158

6th subnet
Network address = 192.168.0.160
First useable address = 192.168.0.161
Broadcast address = 192.168.0.191
Last useable address = 192.168.0.190
IP range = 192.168.0.161 to 192.168.0.190

7th subnet
Network address = 192.168.0.192
First useable address = 192.168.0.193
Broadcast address = 192.168.0.223
Last useable address = 192.168.0.222
IP range = 192.168.0.192 to 192.168.0.222

8th subnet
Network address = 192.168.0.224
First useable address = 192.168.0.225
Broadcast address = 192.168.0.255
Last useable address = 192.168.0.254
IP range = 192.168.0.225 to 192.168.0.254
```

Subnet calculations 5

```
The above subnets are how the network subnets worked out. Subnet 8 was further subnetted.
The test found that IP addresses found within the current 8th subnet were used with a /30 prefix
This idicates that subnet 8 was further subnetted ussing VLSN
```

Subnet calculations 6



```
Calculating 192.168.0.225 to 192.168.0.255 /30

This is acheived by converting host bits to network bits by borrowing.
A further 3 bits were borrowed

The subnet mask for the /30 prfix = 255.255.255.252
Converting the subnet mask 255.255.255.224 into binary

128     64      32      16      8       4       2       1
128 +   64 +    32 +    16 +    8 +     4 +     2 +     1 = 255 (1st, 2nd, and 3rd octet)
1       1       1       1       1       1       1       1
1       1       1       1       1       1       0       0 = 252 (4th octet)

Subnet mask = 1111 1111 1111 1111 1111 1111 111           1 11           00
              -------------Network-------------         -subnet-        -Host-

As shown above 3 host bits were borrowed to create a subnet which is a borrowed network bit.


Calculating network address
128     64      32      16      8       4       2       1
1       1       0       0       0       0       0       0 = 192 (1st octet)
1       0       1       0       1       0       0       0 = 168 (2nd octet)
0       0       0       0       0       0       0       0 = 0 (3rd octet)

Subnet mask = 1100 0000 1010 1000 0000 0000 111           0 00           XX (2 bits of the host address are irrlevant to the network address)
              -------------Network-------------         -subnet-        -Host-

As this IP address has subnet bits the amount of subnets need calculated.
This is achived using 2^n (where n equals the amount of subnet bits) 2^3 = 8 subnets
To calculate the amount of hosts per subnet, use 2^n (where n = the remaining host bits) 2^2 = 4 IP addressess
To calculated useable hosts per subnet, use 2^n-2 (where n = the remaining host bits) 2^5-2 = 2 useable IP addresses
The -2 takes into account the network and broadcast address

The magic number uses the 2^n as before which = 4. this is used to calculated all the subnets by adding it to the prior address.


subnet 1 = 192.168.0.224/30 (Network address)
        +4
subnet 2 = 192.168.0.228/30 (Network address)
        +4
subnet 3 = 192.168.0.232/30 (Network address)
        +4
subnet 4 = 192.168.0.236/30 (Network address)
        +4
subnet 5 = 192.168.0.240/30 (Network address)
        +4
subnet 6 = 192.168.0.244/30 (Network address)
        +4
subnet 7 = 192.168.0.248/30 (Network address)
        +4
subnet 8 = 192.168.0.252/30 (Network address)
        +3
broadcast address for subnet 8.8 = 192.168.0.255
```

Subnet calculations 7

```
1st subnet
Network address = 192.168.0.224
First useable address = 192.168.0.225 (Add one to the previous network address)
Broadcast address = 192.168.0.227 (Minus one from the next subnets network address)
Last useable address = 192.168.0.226 (minus 1 from broadcast address, 227 - 1 = 226)
IP range = 192.168.0.225 to 192.168.0.226

2nd subnet
Network address = 192.168.0.228
First useable address = 192.168.0.229
Broadcast address = 192.168.0.231
Last useable address = 192.168.0.230
IP range = 192.168.0.229 to 192.168.0.230

3rd subnet
Network address = 192.168.0.232
First useable address = 192.168.0.233
Broadcast address = 192.168.0.235
Last useable address = 192.168.0.234
IP range = 192.168.0.233 to 192.168.0.234

4th subnet
Network address = 192.168.0.236
First useable address = 192.168.0.237
Broadcast address = 192.168.0.239
Last useable address = 192.168.0.238
IP range = 192.168.0.237 to 192.168.0.238

5th subnet
Network address = 192.168.0.240
First useable address = 192.168.0.241
Broadcast address = 192.168.0.243
Last useable address = 192.168.0.242
IP range = 192.168.0.241 to 192.168.0.242

6th subnet
Network address = 192.168.0.244
First useable address = 192.168.0.245
Broadcast address = 192.168.0.247
Last useable address = 192.168.0.246
IP range = 192.168.0.245 to 192.168.0.246

7th subnet
Network address = 192.168.0.248
First useable address = 192.168.0.249
Broadcast address = 192.168.0.251
Last useable address = 192.168.0.250
IP range = 192.168.0.249 to 192.168.0.250

8th subnet
Network address = 192.168.0.252
First useable address = 192.168.0.253
Broadcast address = 192.168.0.255
Last useable address = 192.168.0.254
IP range = 192.168.0.253 to 192.168.0.254

This is how the subnets worked out. See subnet table for reference.
```

Subnet calculations 8

Appendix B (RSA Verifying)

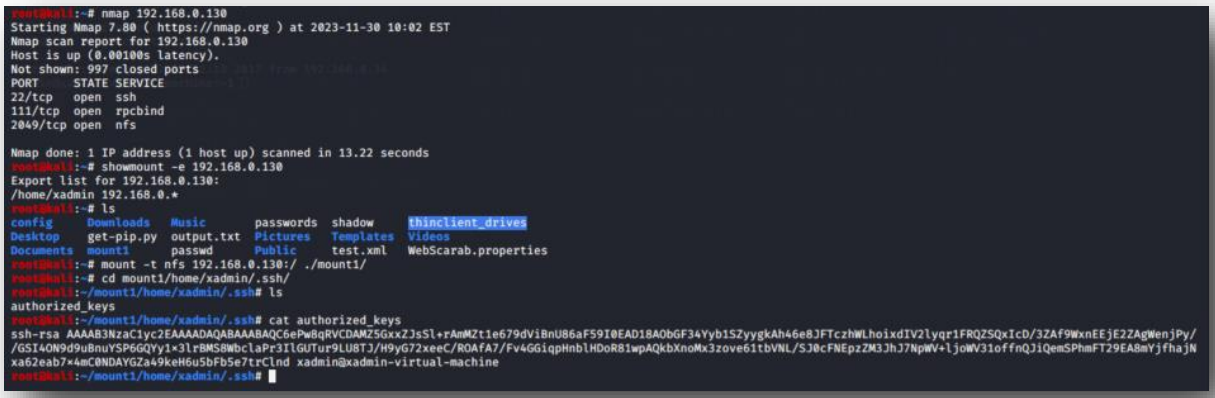


Figure 69:Authorized keys within 192.168.0.130

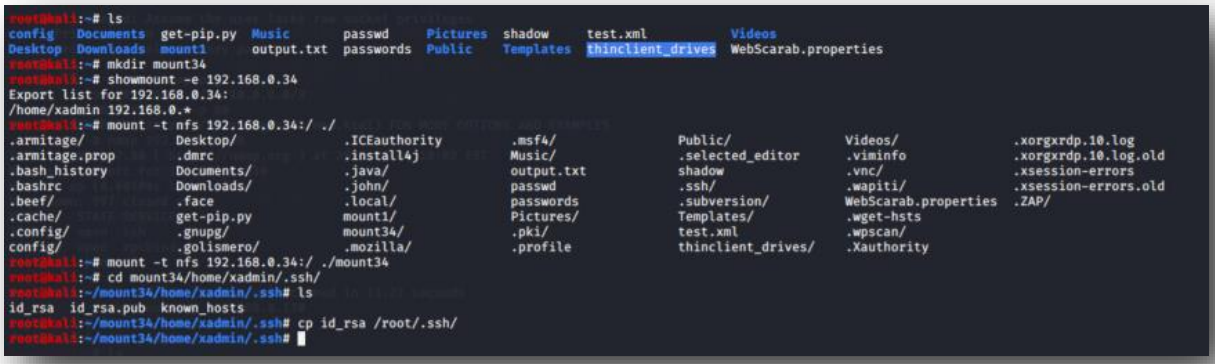


Figure 70:Copying private key from 192.168.0.34 to kali's root/.ssh folder

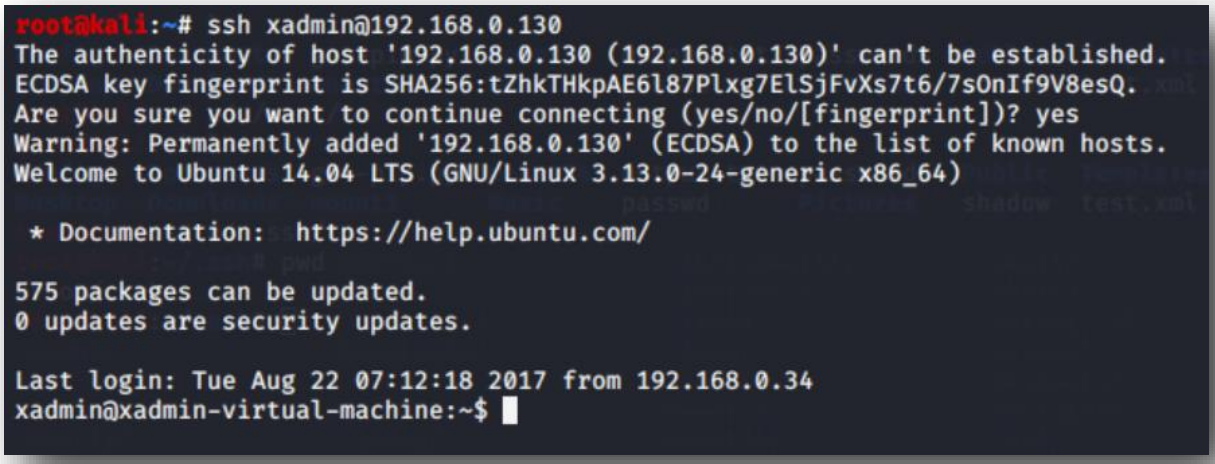


Figure 71:Testing private key by SSH into 192.168.0.130

Appendix C (Tunnelling)

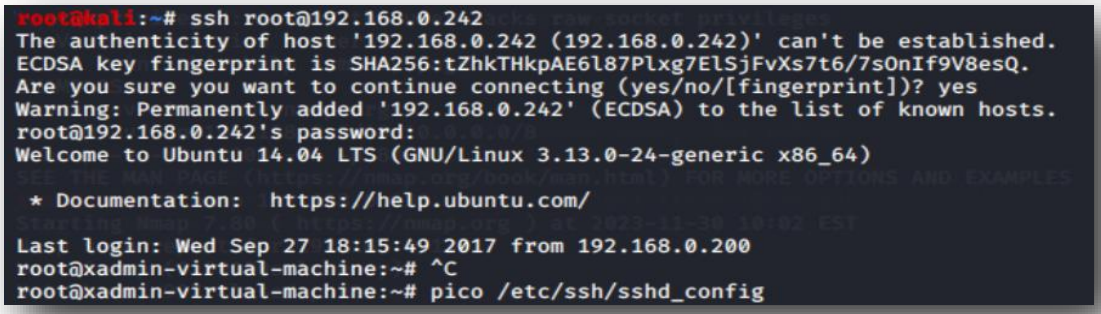


Figure 72:SSH into 192.168.0.242

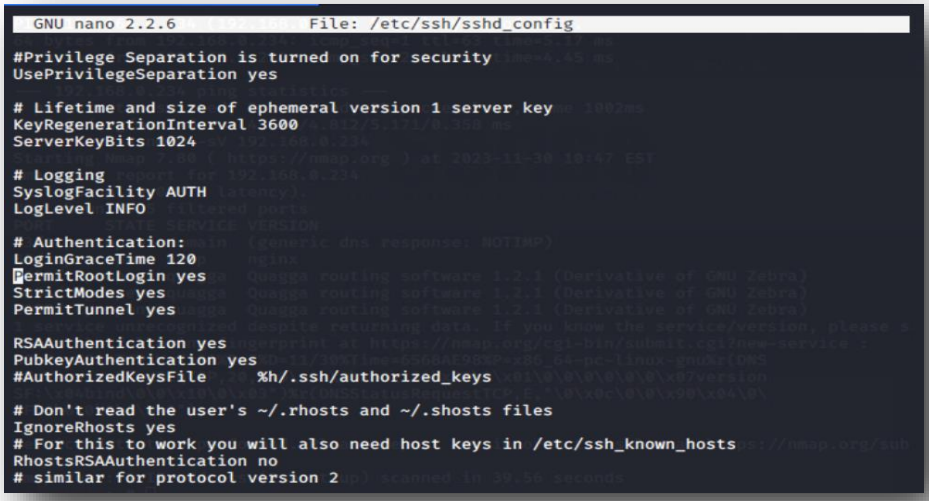


Figure 73:Permitting tunnel on 192.168.0.242 by editing the sshd_config file

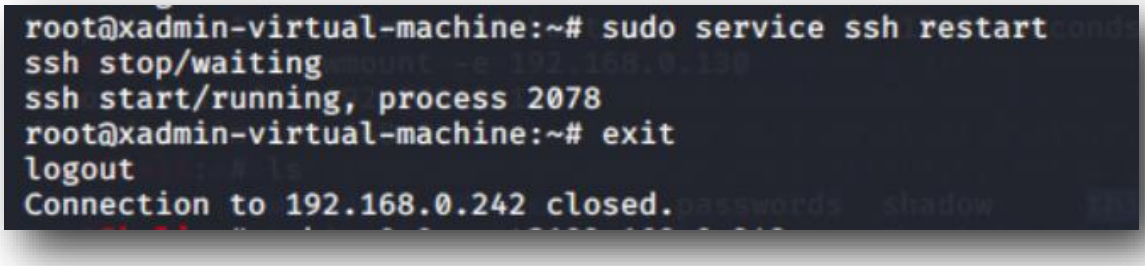


Figure 74:Restarting SSH service

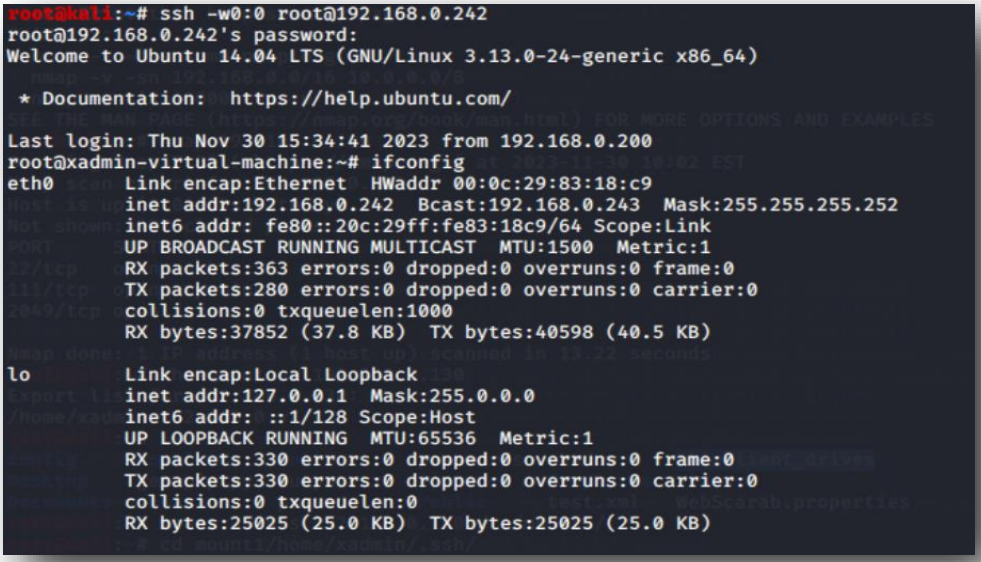


Figure 75:Create tunnel to 192.168.0.242

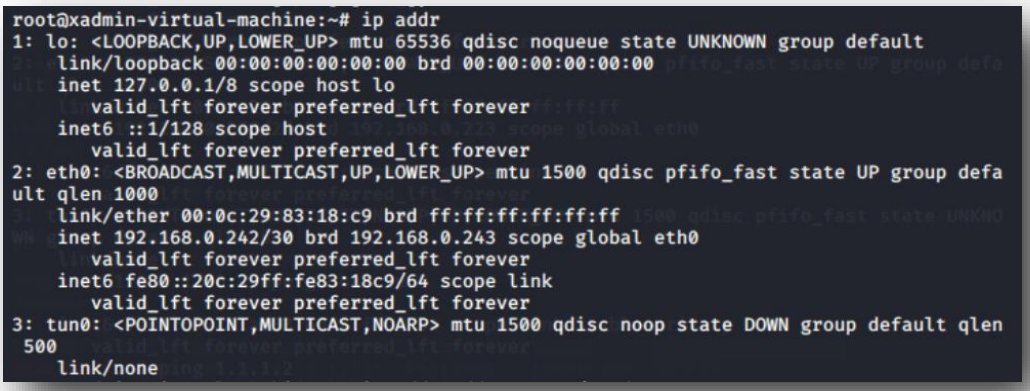


Figure 76:check tunnel status with ip addr

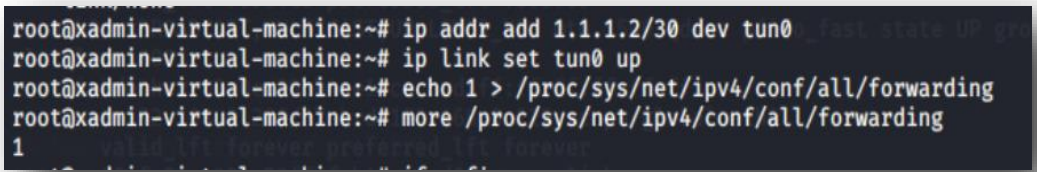


Figure 77:Add IP address, set tunnel up and allow port forwarding

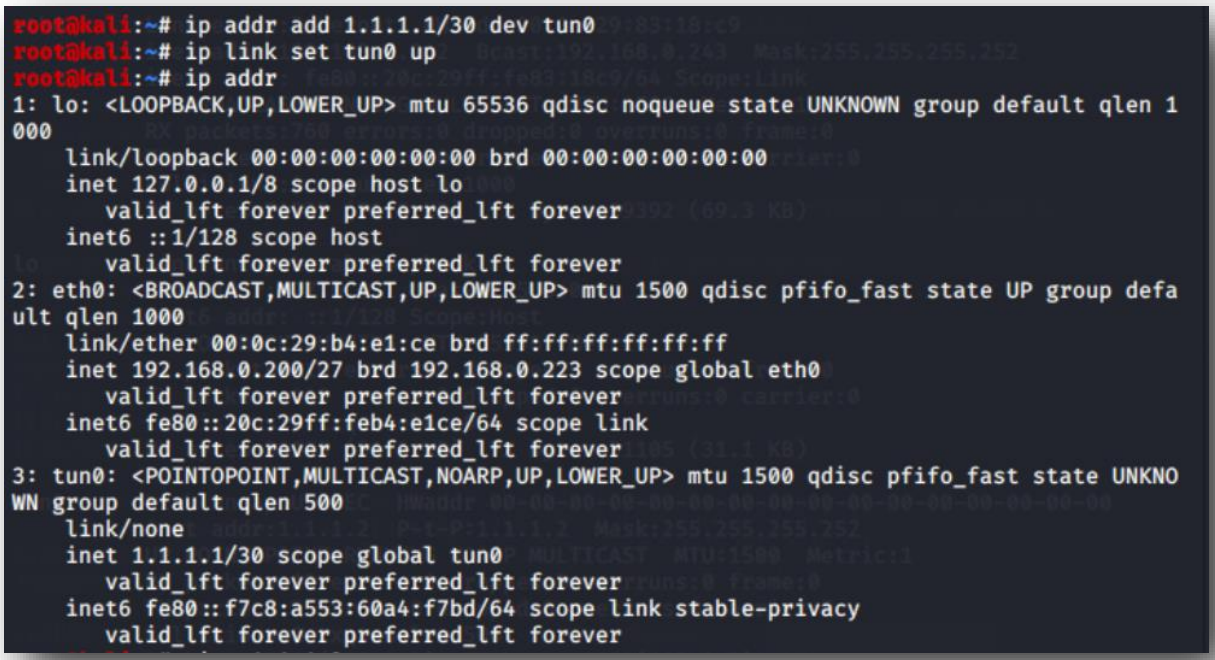


Figure 78:Set IP address and status up on the kali machine

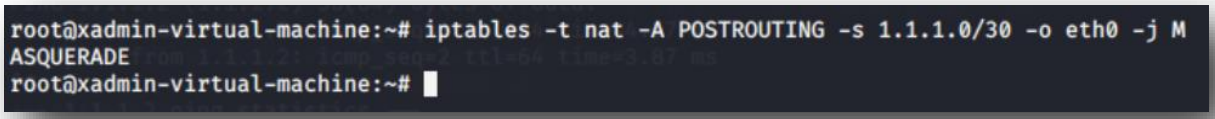


Figure 79:Edit IP tables and enable NAT

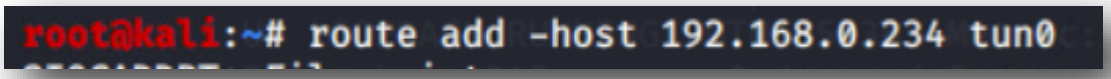


Figure 80:Add route to 192.168.0.234 via tun0

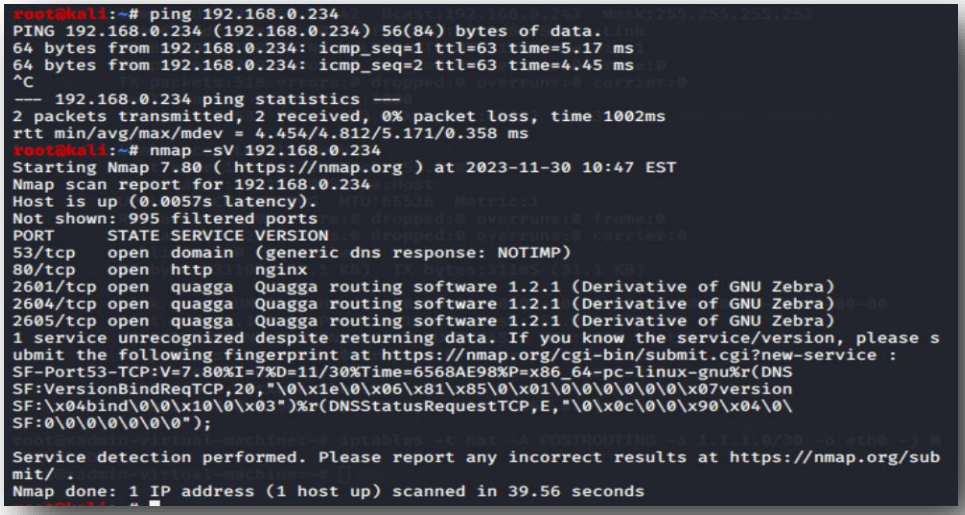


Figure 81:Ping and Nmap scan on 192.168.0.234

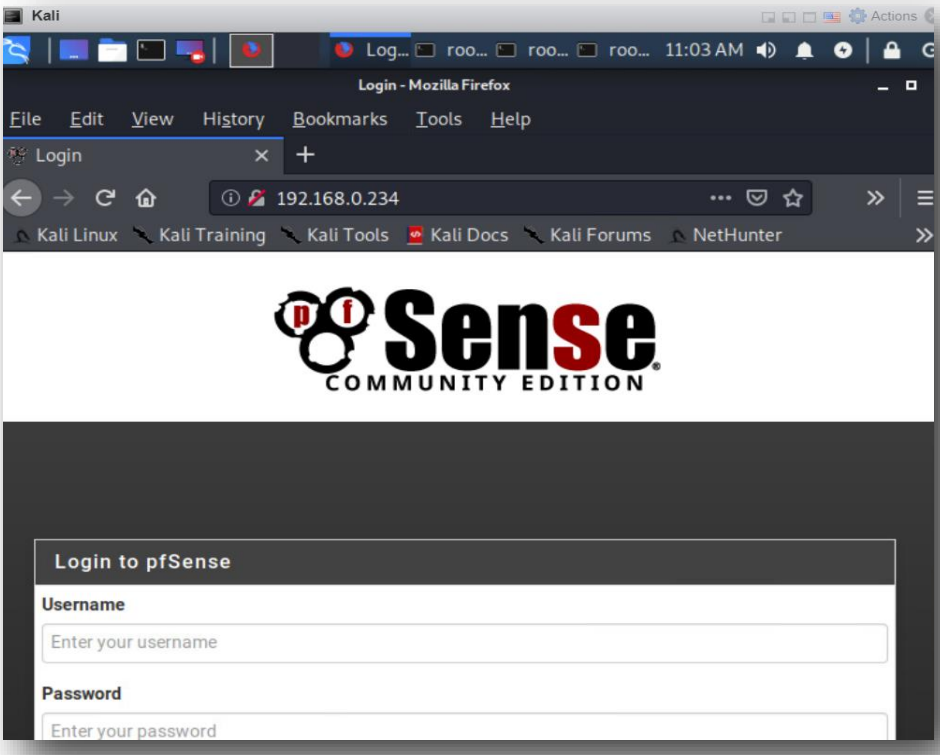


Figure 82:Access the interface via port 80

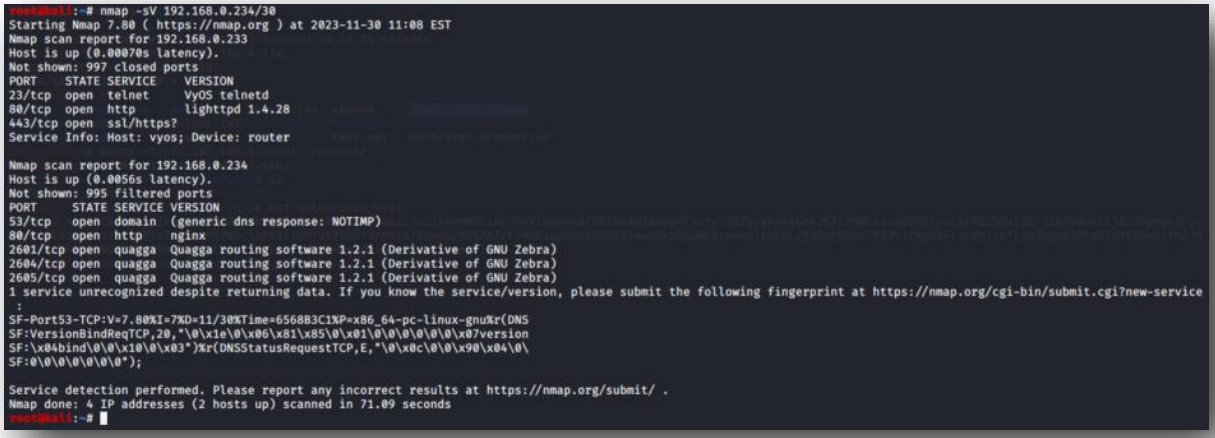


Figure 83:Nmap scan on 192.168.0.234/30

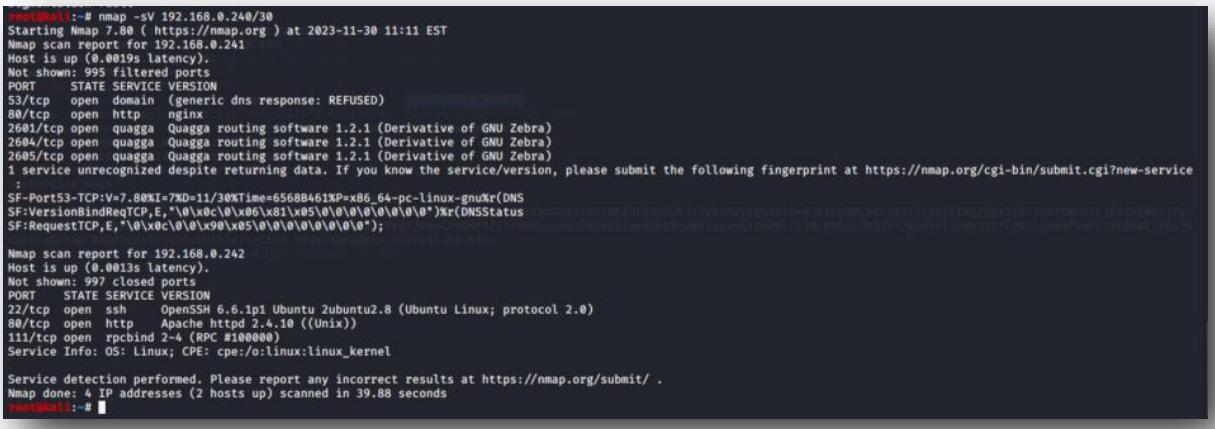


Figure 84:Nmap scan on 192.168.240/30

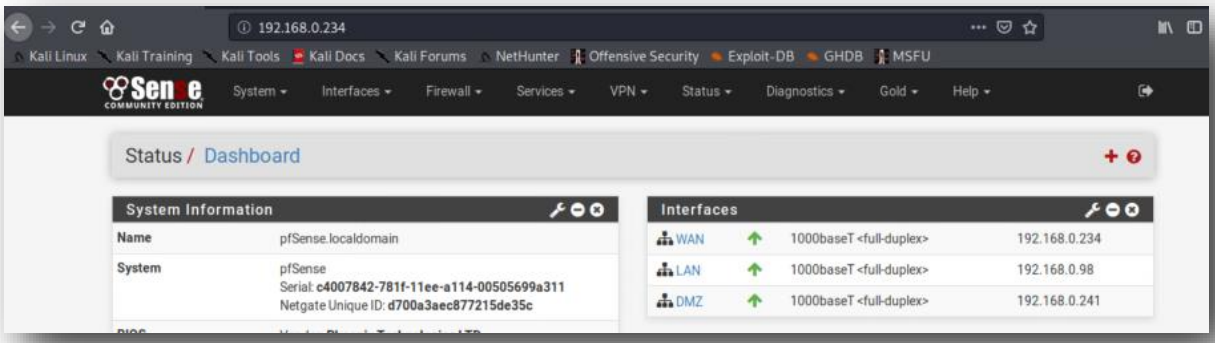


Figure 85:Firewall dashboard on 192.168.0.234

Figure 86:Ping sweep on 192.168.0.X/24 from behind the firewall

Figure 87:Device found via the ping sweep

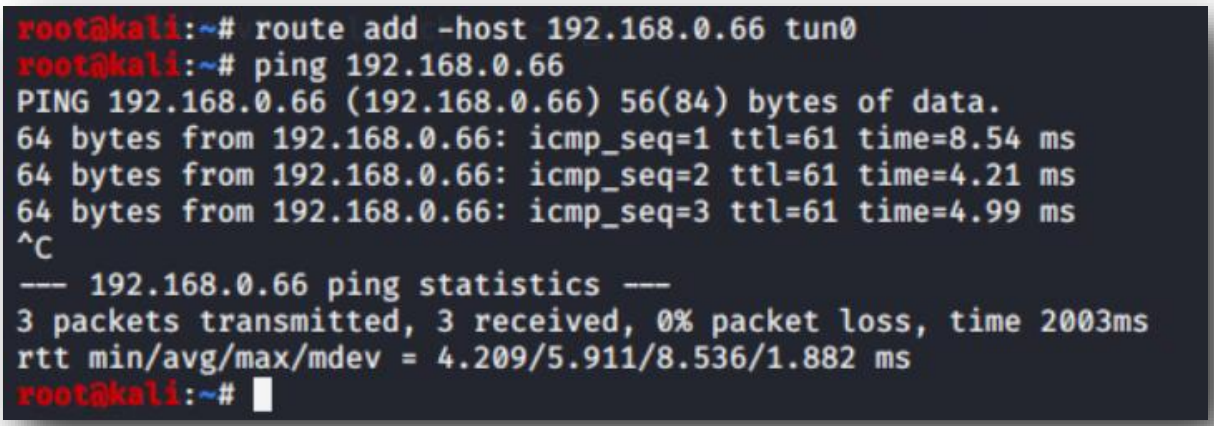


Figure 88:Adding the route towards 192.168.0.66 and pinging to test connection

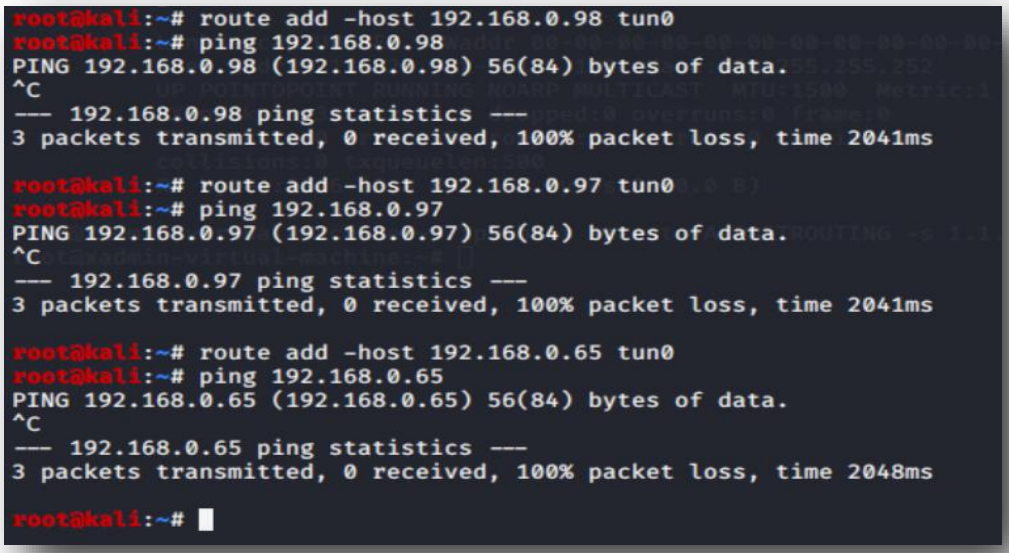


Figure 89:Adding route to known IP addresses and testing by using the ping command

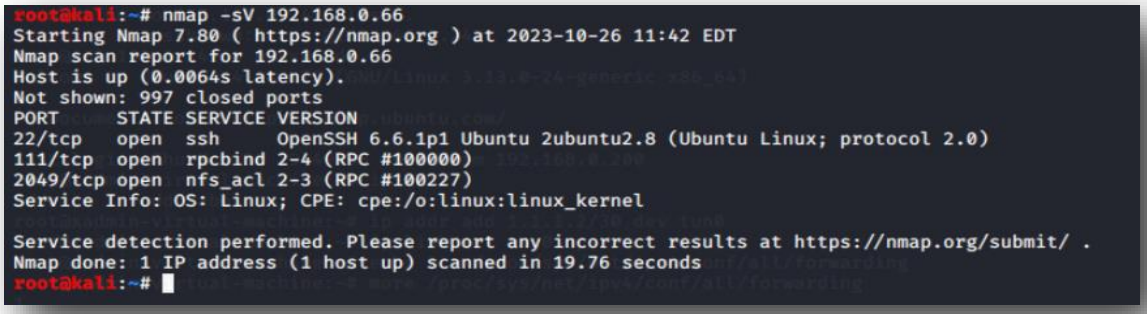


Figure 90: Nmap scan on 192.168.0.66

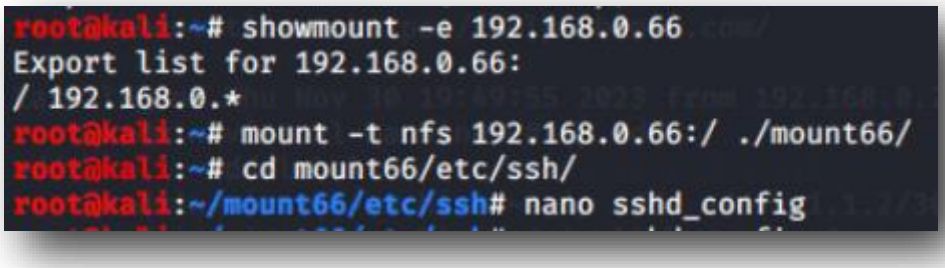


Figure 91:Show mount on 192.168.0.66 and nano the sshd_config file

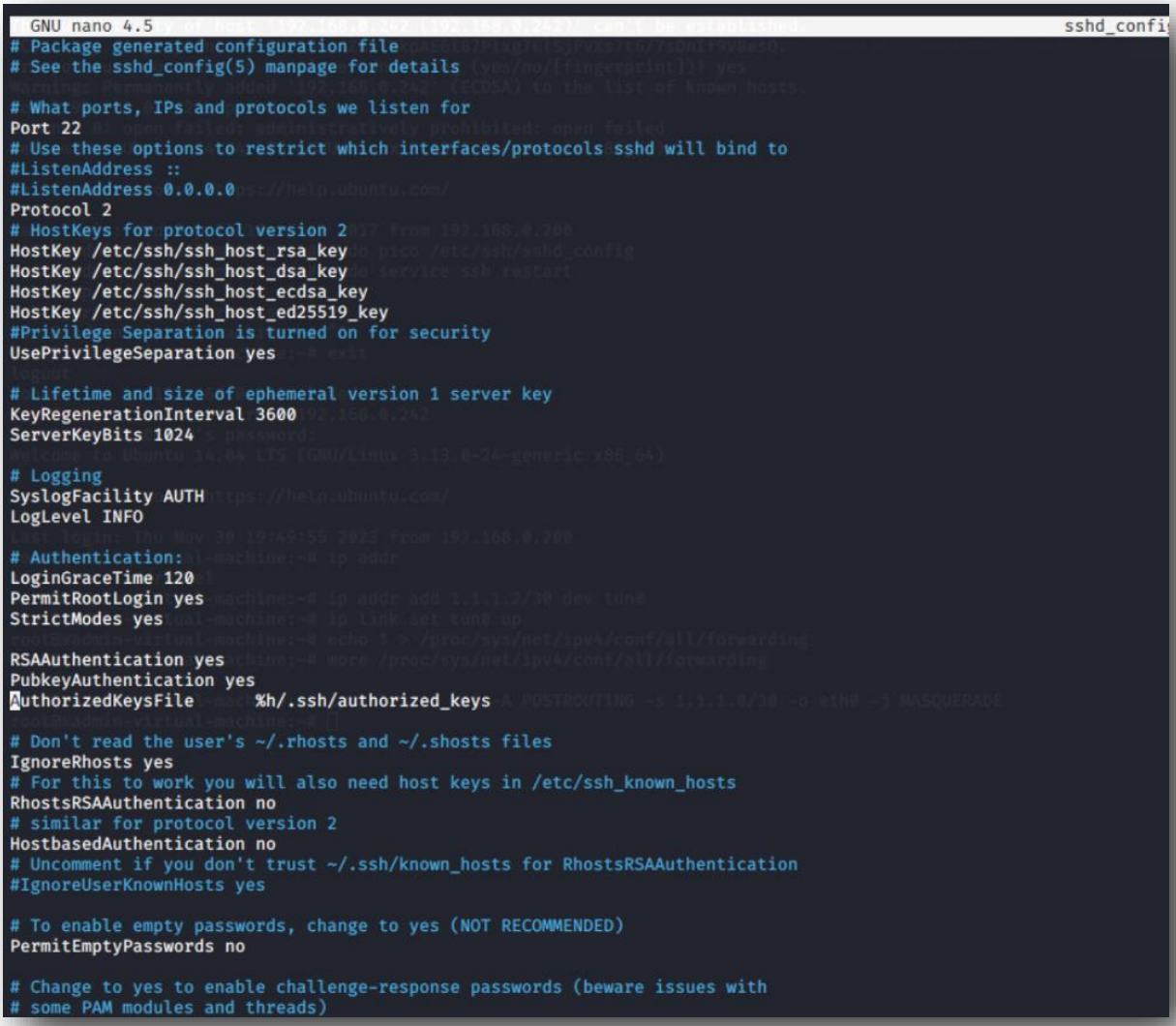


Figure 92:Permit authorized keys to gain access

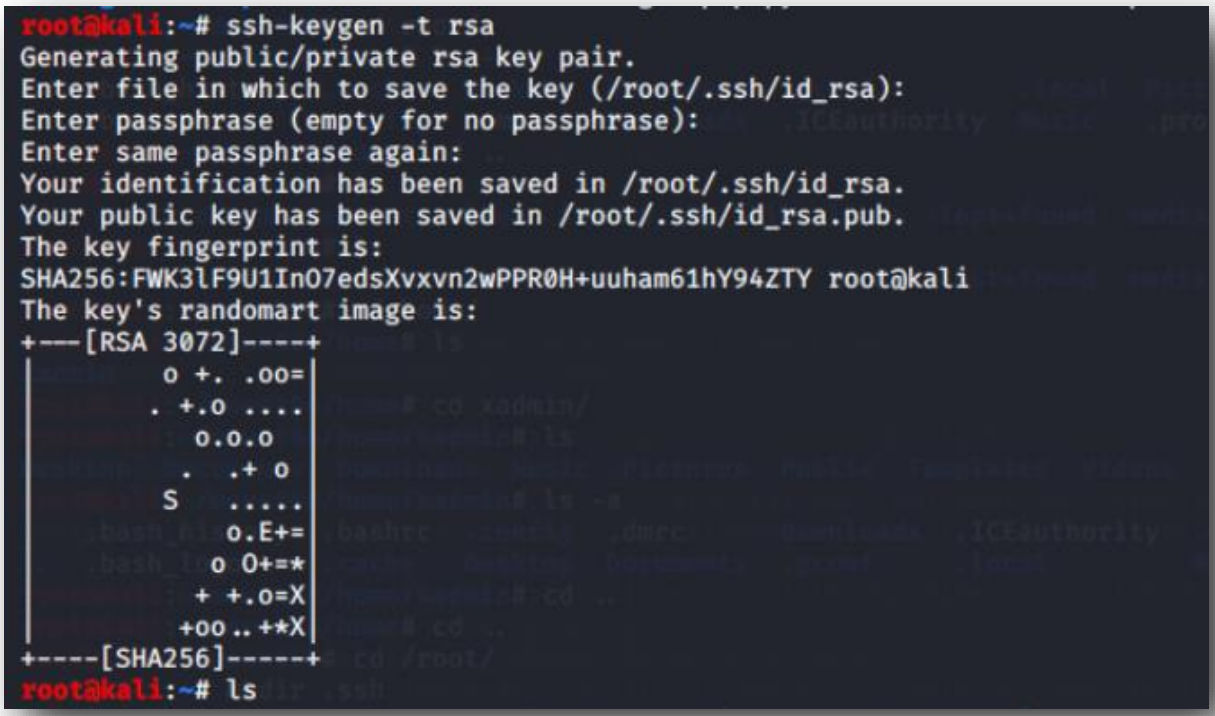


Figure 93:RSA key generation

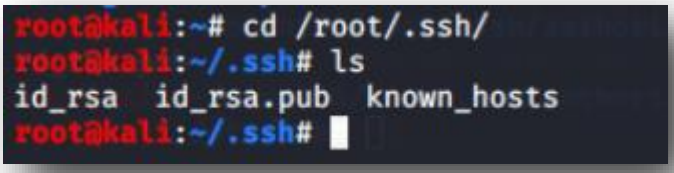


Figure 94:Checking RSA keys

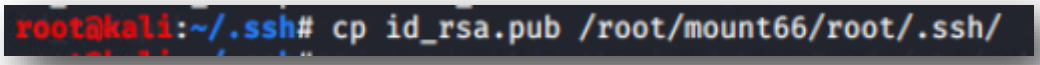


Figure 95:Copying public key to /root/.ssh directory on 192.168.0.66 with the use of NFS

```
root@kali:~/.ssh# cd /root/mount66/root/.ssh/
root@kali:~/mount66/root/.ssh#
```

Figure 96:changing directory to where the key should be stored

```
root@kali:~/mount66/root/.ssh# mv id_rsa.pub authorized_keys
root@kali:~/mount66/root/.ssh# ls
authorized_keys
```

Figure 97:changing name of public key to allow sshd_config file to identify the public key

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

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

Last login: Thu Nov 30 20:19:31 2023 from 192.168.0.242
root@xadmin-virtual-machine:~#
```

Figure 98:Using SSH to gain access to 192.168.0.66

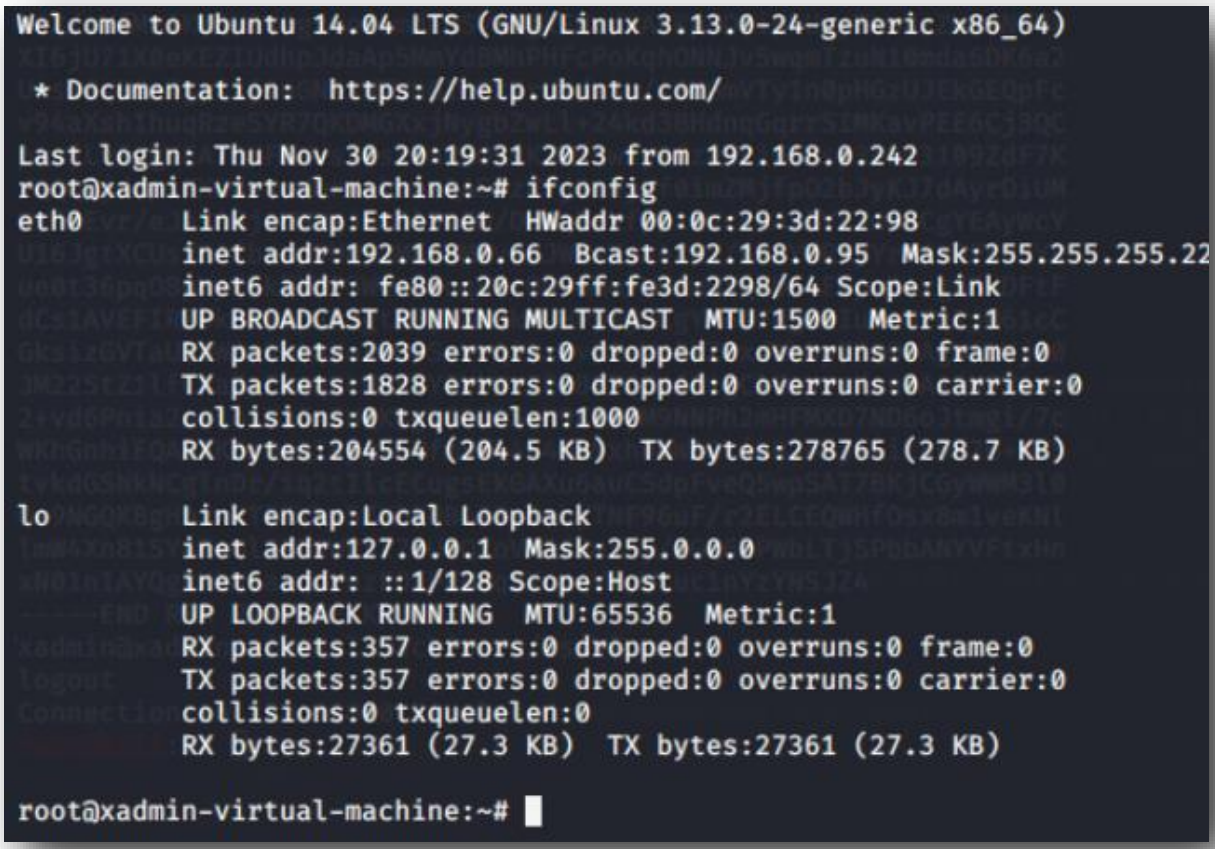


Figure 99:Ifconfig command on 192.168.0.66

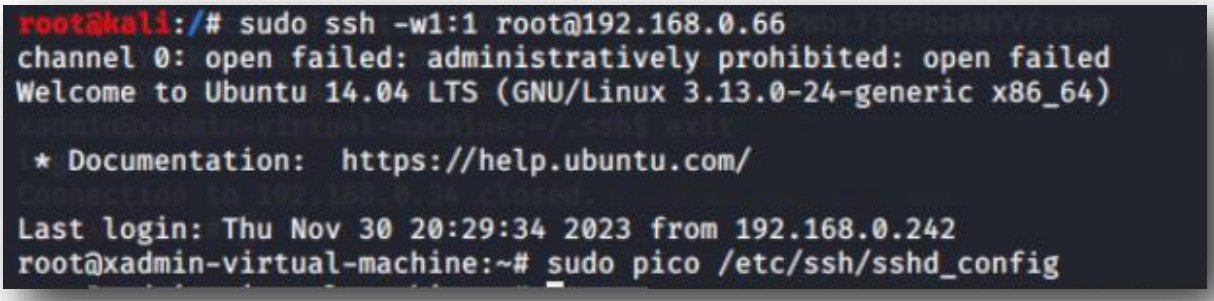


Figure 100:SSH into 192.168.0.66

Figure 101:Permitting tunnel in the sshd_config file on 192.168.0.66

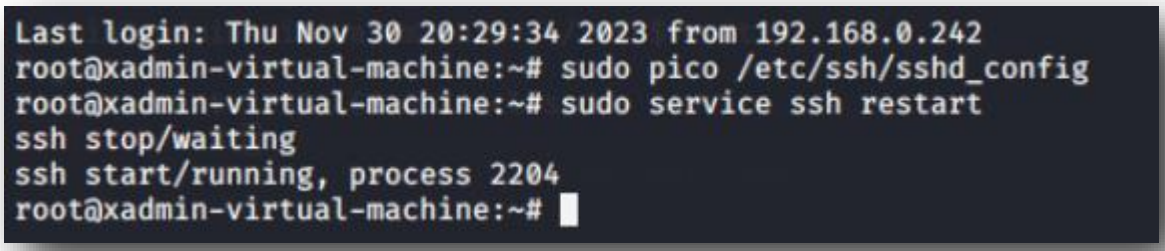


Figure 102:Restarting SSH service on 192.168.0.66

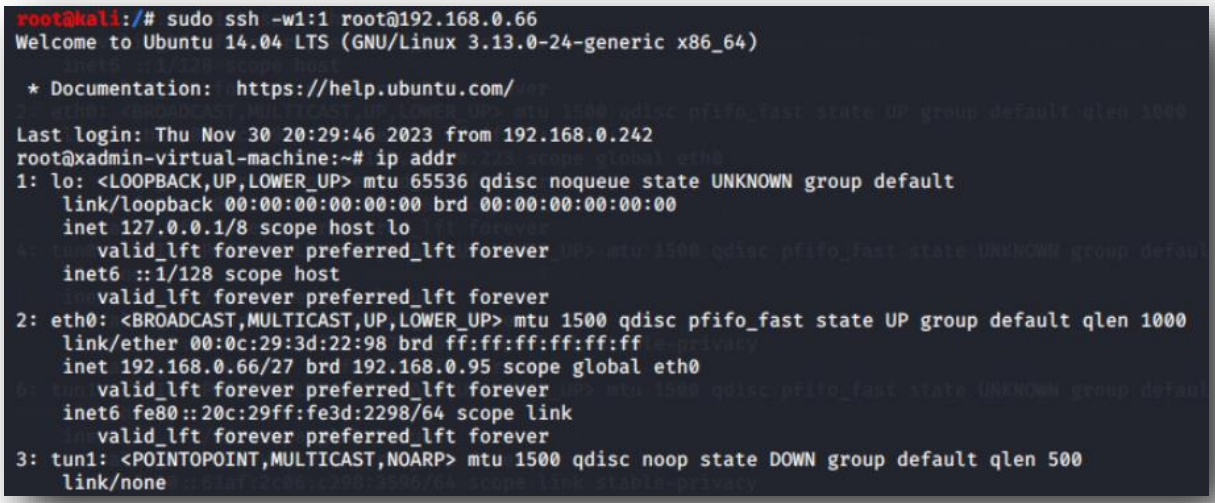


Figure 103:SSH into 192.168.0.66 and creating a tunnel

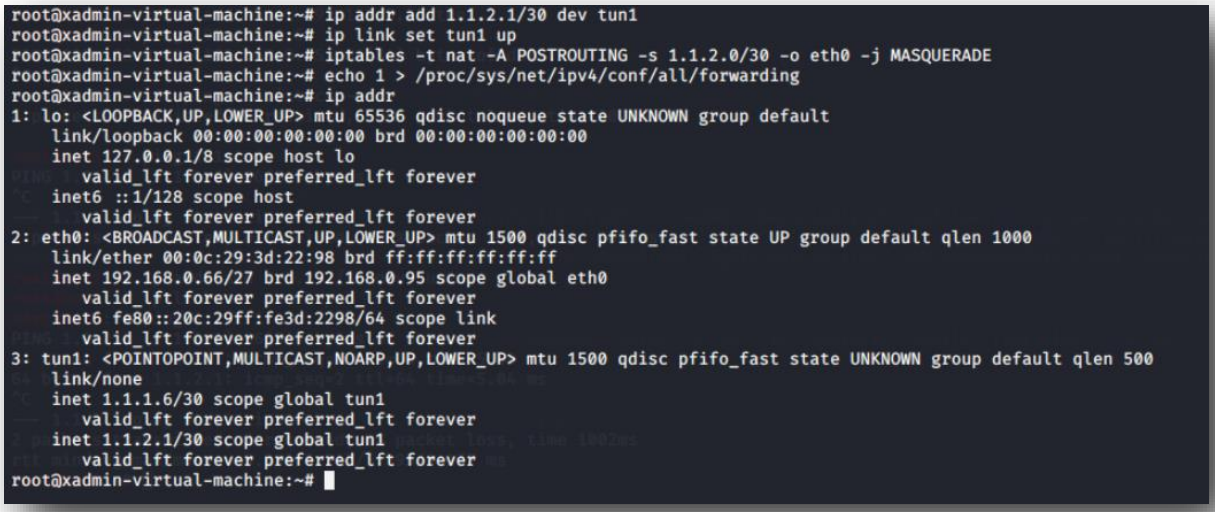


Figure 104:Adding an IP address and changing the state to up, further editing IP tables and allowing port forwarding. Checking using ip addr

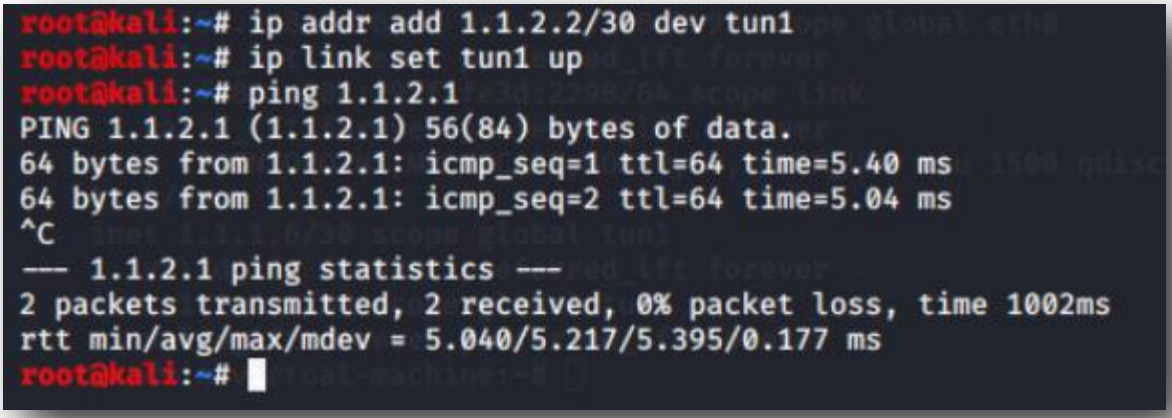


Figure 105:Adding an IP and changing state on the kali machine

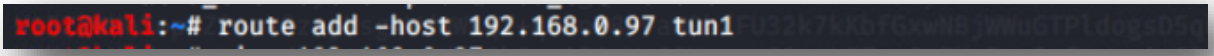


Figure 106:Adding a route towards 192.168.0.97 via tun1

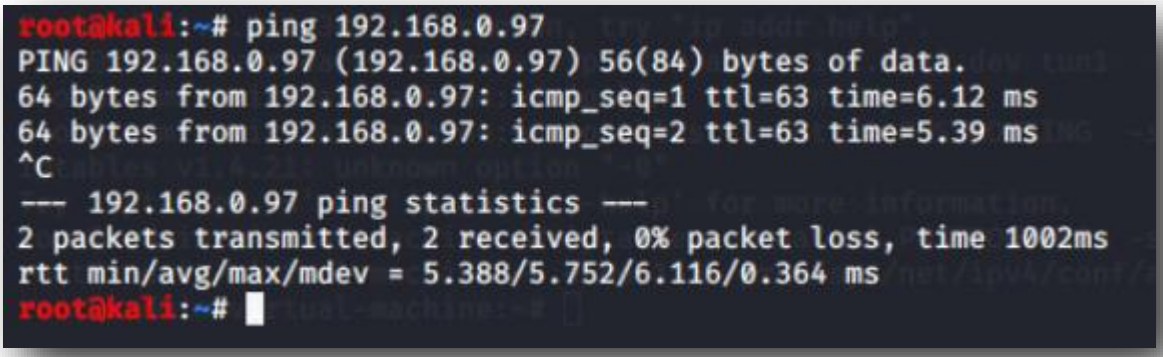


Figure 107:Testing connection via ping command

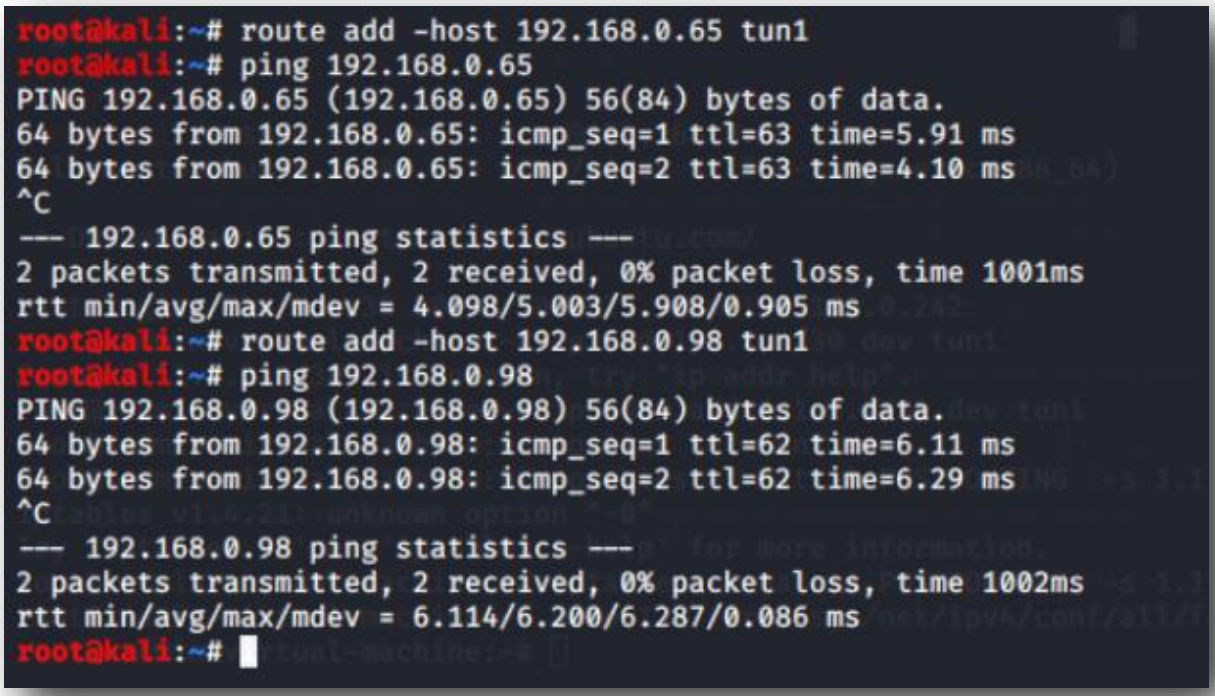


Figure 108Adding route towards known IP addresses and testing using a ping command

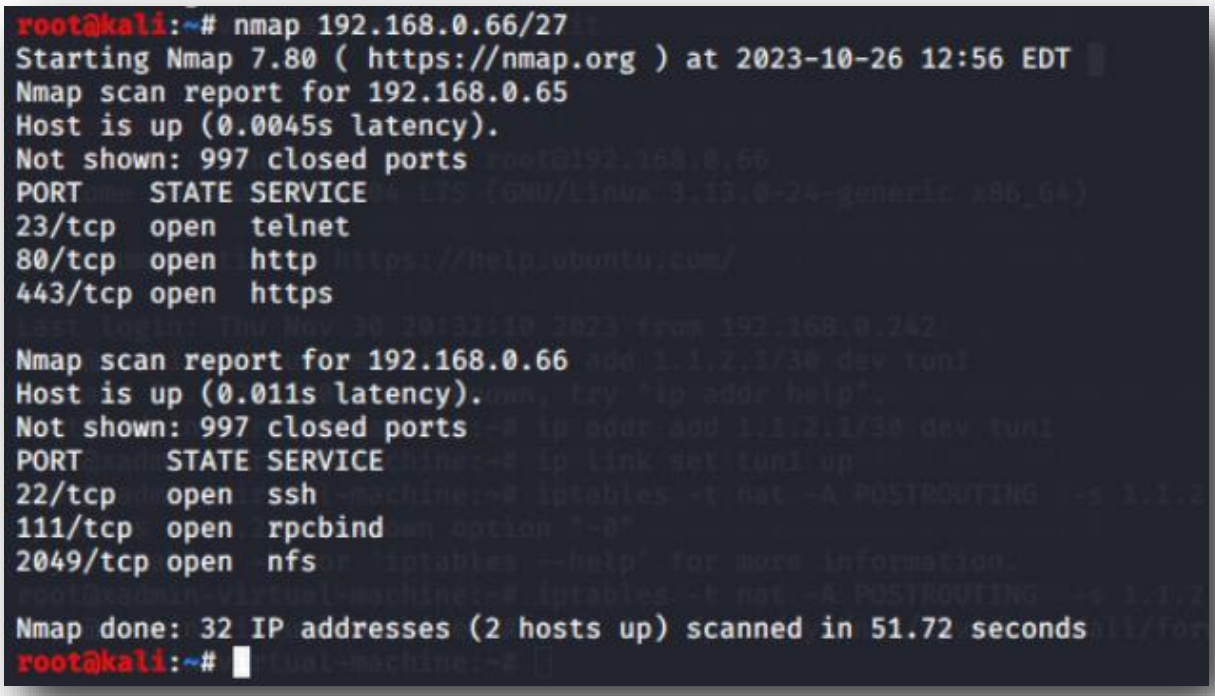


Figure 109:Nmap scan on 192.168.0.66/27

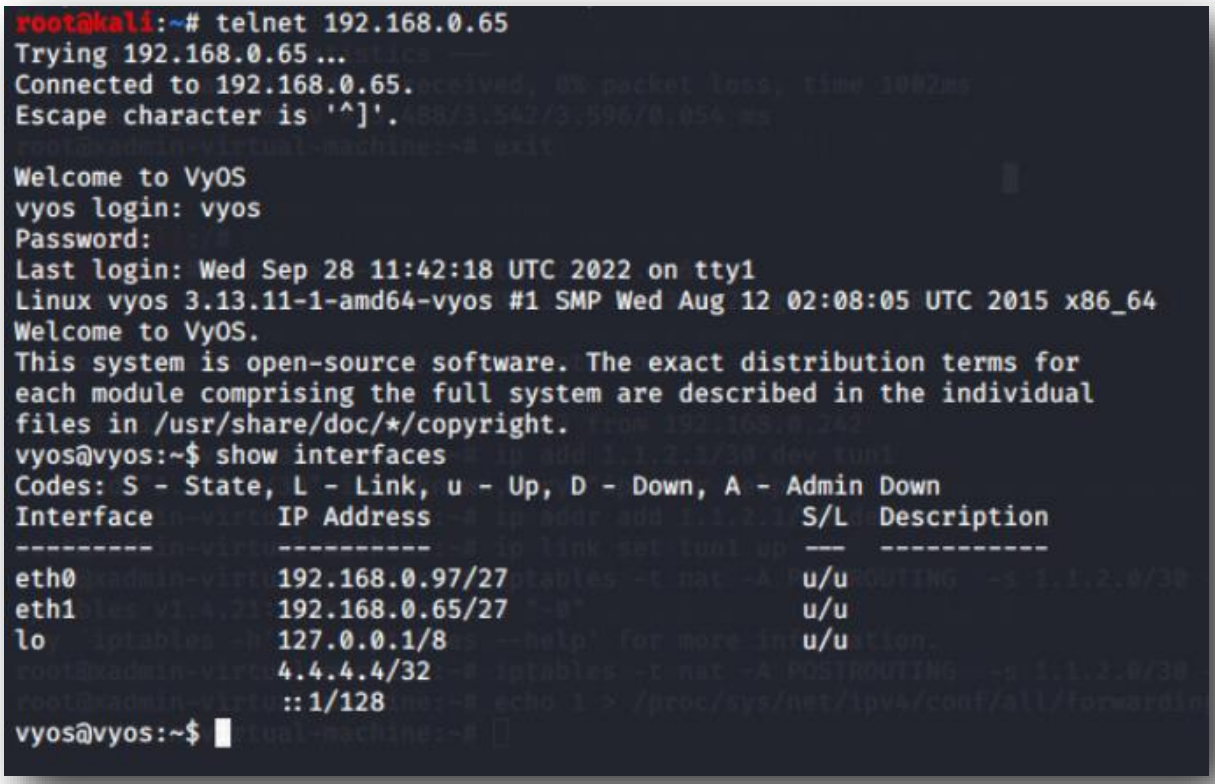


Figure 110:Telnet into 192.168.0.65 and use show IP interfaces

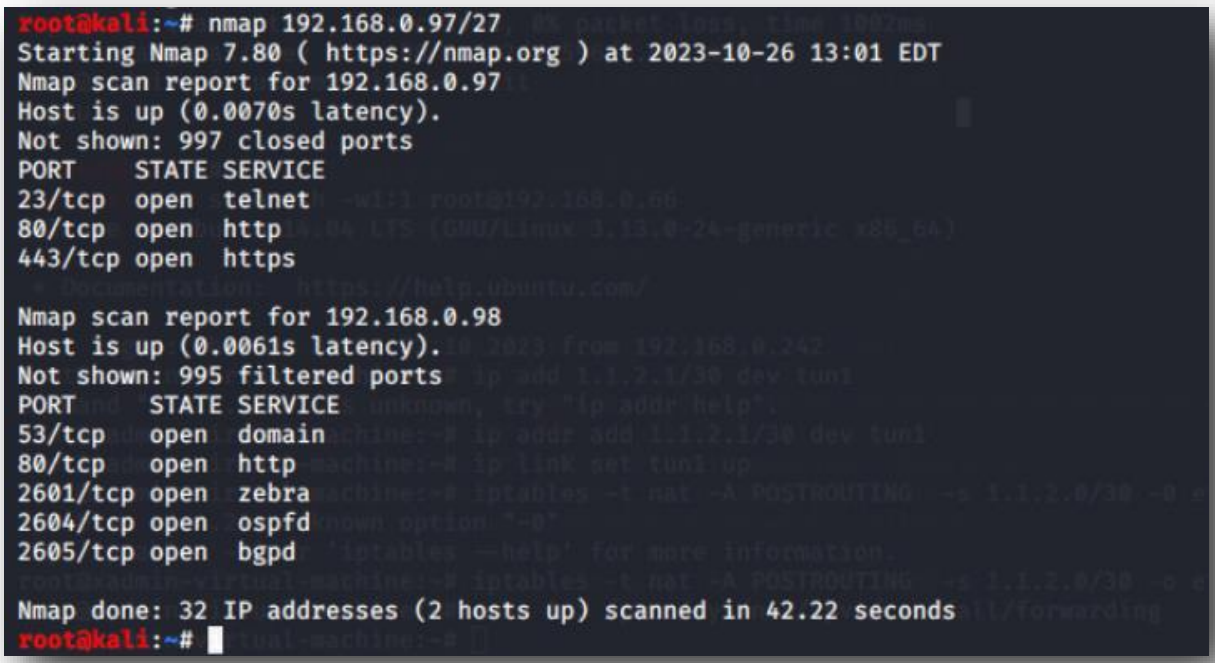


Figure 111:Nmap scan on 192.168.0.97/27

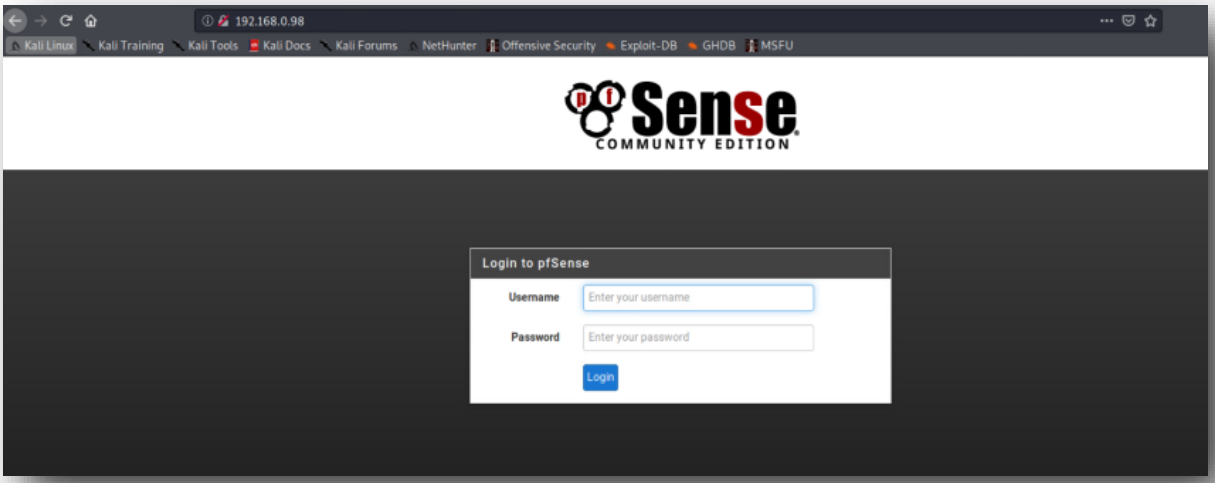


Figure 112:Accessing the firewall from the LAN side

Appendix D (UDP Scanning)

```
root@kali:~# nmap -sU -sV 192.168.0.97
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 13:48 EDT
Stats: 0:13:40 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 77.48% done; ETC: 14:06 (0:03:55 remaining)
Nmap scan report for 192.168.0.97
Host is up (0.0048s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    SNMPv1 server; net-snmp SNMPv3 server (public)
Service Info: Host: vyos

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1096.08 seconds
root@kali:~#
```

UDP Scanning 1

```
root@kali:~# nmap -sU -sV 192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 13:49 EDT
Stats: 0:13:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 77.26% done; ETC: 14:06 (0:03:57 remaining)
Nmap scan report for 192.168.0.230
Host is up (0.0012s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1101.70 seconds
root@kali:~#
```

UDP Scanning 2

```
root@kali:~# nmap -sU -sV 192.168.0.225 192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 13:48 EDT
Stats: 0:13:50 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 76.29% done; ETC: 14:06 (0:04:10 remaining)
Nmap scan report for 192.168.0.225
Host is up (0.00059s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server
```

UDP Scanning 3

```
Nmap scan report for 192.168.0.193
Host is up (0.00054s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server
MAC Address: 00:50:56:99:6C:E2 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 2 IP addresses (2 hosts up) scanned in 2223.06 seconds
root@kali:~#
```

UDP Scanning 4

```
root@kali:~# nmap -sU -sV 192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 13:49 EDT
Stats: 0:13:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 77.26% done; ETC: 14:06 (0:03:57 remaining)
Nmap scan report for 192.168.0.230
Host is up (0.0012s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
123/udp open  ntp     NTP v4 (unsynchronized)
161/udp open  snmp    net-snmp; net-snmp SNMPv3 server

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1101.70 seconds
root@kali:~#
```

UDP Scanning 5

```
root@kali:~# nmap -sU -sV 192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 13:49 EDT
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:08:21 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 47.49% done; ETC: 14:06 (0:09:01 remaining)
Stats: 0:13:33 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 76.87% done; ETC: 14:06 (0:04:01 remaining)
Stats: 0:19:30 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 14:08 (0:00:18 remaining)
Nmap scan report for 192.168.0.210
Host is up (0.00059s latency).
Not shown: 995 closed ports
PORT     STATE         SERVICE VERSION
111/udp  open          rpcbind 2-4 (RPC #100000)
631/udp  open|filtered ipp
1022/udp open          rpcbind 2-4 (RPC #100000)
2049/udp open          nfs_acl 2-3 (RPC #100227)
5353/udp open          mdns    DNS-based service discovery
MAC Address: 00:0C:29:AA:6E:93 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1196.55 seconds
root@kali:~#
```

UDP Scanning 6

```
root@kali:~# nmap -sU -sV 172.16.221.237
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 14:11 EDT
Stats: 0:13:08 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 74.50% done; ETC: 14:29 (0:04:26 remaining)
Stats: 0:17:09 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 97.20% done; ETC: 14:29 (0:00:29 remaining)
Nmap scan report for 172.16.221.237
Host is up (0.0012s latency).
Not shown: 999 closed ports
PORT     STATE SERVICE VERSION
5353/udp open  mdns    DNS-based service discovery

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1102.23 seconds
root@kali:~#
```

UDP Scanning 7

```
root@kali:~# nmap -sU -sV 192.168.0.34
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 14:11 EDT
Stats: 0:13:26 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.44% done; ETC: 14:28 (0:04:18 remaining)
Nmap scan report for 192.168.0.34
Host is up (0.0017s latency).
Not shown: 995 closed ports
PORT     STATE         SERVICE VERSION
111/udp  open          rpcbind 2-4 (RPC #100000)
631/udp  open|filtered ipp
998/udp  open          rpcbind 2-4 (RPC #100000)
2049/udp open          nfs_acl 2-3 (RPC #100227)
5353/udp open          mdns    DNS-based service discovery

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1200.21 seconds
```

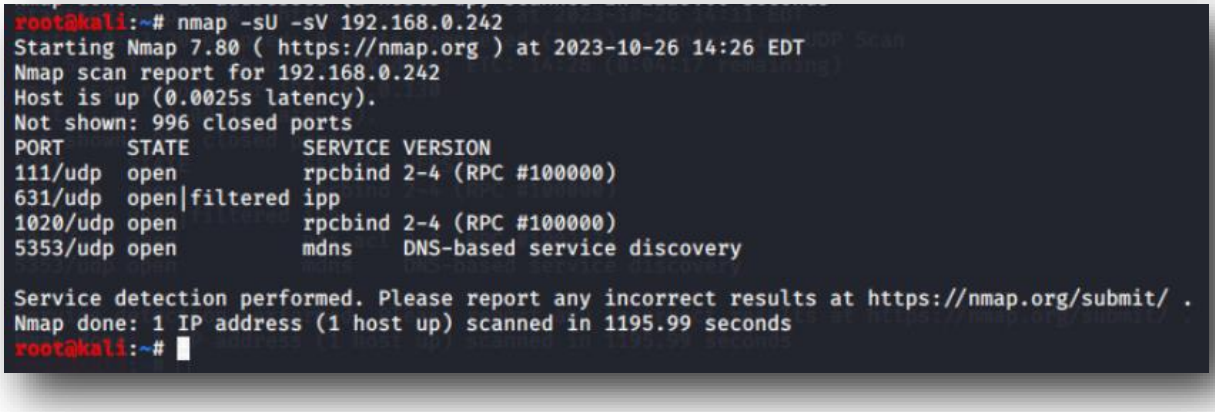
UDP Scanning 8

```
root@kali:~# nmap -sU -sV 192.168.0.130
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 14:11 EDT
Stats: 0:13:20 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 75.37% done; ETC: 14:28 (0:04:17 remaining)
Nmap scan report for 192.168.0.130
Host is up (0.0021s latency).
Not shown: 996 closed ports
PORT     STATE         SERVICE VERSION
111/udp  open          rpcbind 2-4 (RPC #100000)
631/udp  open|filtered ipp
2049/udp open          nfs_acl 2-3 (RPC #100227)
5353/udp open          mdns    DNS-based service discovery

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1195.99 seconds
```

UDP Scanning 9

```
root@kali:~# nmap -sU -sV 192.168.0.66
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-26 14:30 EDT
Nmap scan report for 192.168.0.66
Host is up (0.0041s latency).
Not shown: 994 closed ports
PORT      STATE         SERVICE VERSION
111/udp   open          rpcbind 2-4 (RPC #100000)
631/udp   open|filtered ipp
1022/udp  open          rpcbind 2-4 (RPC #100000)
2049/udp  open          nfs_acl 2-3 (RPC #100227)
5353/udp  open          mdns    DNS-based service discovery
36489/udp open|filtered unknown

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1196.50 seconds
root@kali:~#
```

UDP Scanning 10



UDP Scanning 11

Bibliography

Ashtari, H., N/A. *Intrusion Detection System vs. Intrusion Prevention System: Key Differences and Similarities*. [Online]
Available at: https://www.spiceworks.com/it-security/network-security/articles/ids-vs-ips/#:~:text=An%20intrusion%20detection%20system%20(IDS,and%20prevents%20any%20detected%20threats.
[Accessed 13 12 2023].

Chandel, R., 2019. *wordpress-reverse-shell*. [Online]
Available at: https://www.hackingarticles.in/wordpress-reverse-shell/
[Accessed 18 11 2023].

cvedetails.com, 2023. *Apache HTTP Server 2.2.22*. [Online]
Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-490988/Apache-Http-Server-2.2.22.html?page=1&order=1&trc=44&sha=414987d7cb7c92ccc0c9c5865d20e4a0a04ebfe3
[Accessed 13 12 2023].

cvedetails.com, 2023. *Lighttpd*. [Online]
Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-2713/Lighttpd.html
[Accessed 13 12 2023].

Gergen, D., 2014. *mitigating-bash-shellshock*. [Online]
Available at: https://www.crowdstrike.com/blog/mitigating-bash-shellshock/
[Accessed 10 12 2023].

Kumar, S., N/A. *web-application-security-common-vulnerabilities-and-how-to-mitigate-them*. [Online]
Available at: https://medium.com/@ersat.ice37/web-application-security-common-vulnerabilities-and-how-to-mitigate-them-e4feae3501dd#:~:text=According%20to%20NTT%20Application%20Security,for%20at%20least%20two%20years
[Accessed 22 11 2023].

MITRE, 2014. *CVE-2014-6271*. [Online]
Available at: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
[Accessed 05 12 2023].

NVD.NIST, 2023. *OpenSSH*. [Online]
Available at: https://nvd.nist.gov/vuln/search/results?form_type=Advanced&cves=on&cpe_version=cpe:/a:openbsd:openssh:6.6.1p1
[Accessed 13 12 2023].

Paloalto, N/A. *Types of Firewalls Defined and Explained.* [Online]
Available at: https://www.paloaltonetworks.co.uk/cyberpedia/types-of-firewalls
[Accessed 13 12 2023].

Pentestmonkey, n/a. *php-reverse-shell.* [Online]
Available at: https://pentestmonkey.net/tools/web-shells/php-reverse-shell
[Accessed 18 11 2023].

Vyos, 2023. *Support Vyos*. [Online]
Available at: https://support.vyos.io/en/support/solutions/articles/103000096330-vyos-default-user-and-password
[Accessed 15 11 2023].

VyOS, 2023. *VyOS*. [Online]
Available at: https://vyos.io/
[Accessed 15 11 2023].