Abertay University

Network Infrastructure Assessment

Declan Doyle
1600219

CMP314 Computer Networking 2

BSc Ethical Hacking
Year 3

2018/19

Abstract

Computer networks are a key part of any modern organisation, as they allow for a modern workflow. They can greatly increase productivity and improve an organisations greatly. However if a network is implemented incorrectly, many problems can occur.

Acme Inc. had a network manager who left the company in a bad position. There was no documentation on their current network setup and they were unsure if the network was secure. They requested an infrastructure assessment in order to provide them with a network map, as well as a security assessment to determine if their network was suitable.

It was found that the network was configured poorly, and that there were several security issues. Recommendations were made for mitigations on these security issues.

Table of Contents

1. Introduction

1.1. Background

Computer network infrastructure is the combination of hardware and software that allows an organisation or enterprise to have network connectivity and communication operations. (Techopedia, 2018) It can provide users access to larger areas of the organisation, and even access to the greater internet and the world wide web. Many organisations have a team dedicated to network management and development.

ACME Inc. had employed a network manager to maintain their network infrastructure, however the manager left and so ACME Inc, had little to no knowledge and understanding of their network infrastructure. They requested a complete network infrastructure assessment of their own network, including a security assessment, and a critical evaluation. They have provided a machine preloaded with Kali Linux, a linux distribution preloaded with a set of tools designed for penetration testing and performing security audits. (Offensive Security, 2013) They were concerned about the effects of unproven tools and so only the Kali Linux machine was used to test the network. The credentials for the Kali machine were *root* and *toor.*

1.2. Aims

- To conduct a network infrastructure assessment and successfully map out and detail the network infrastructure, providing critical feedback on its design.
- Discover any vulnerabilities in the network, and provide mitigations for said vulnerabilities.
- Provide critical feedback on the security of the network.

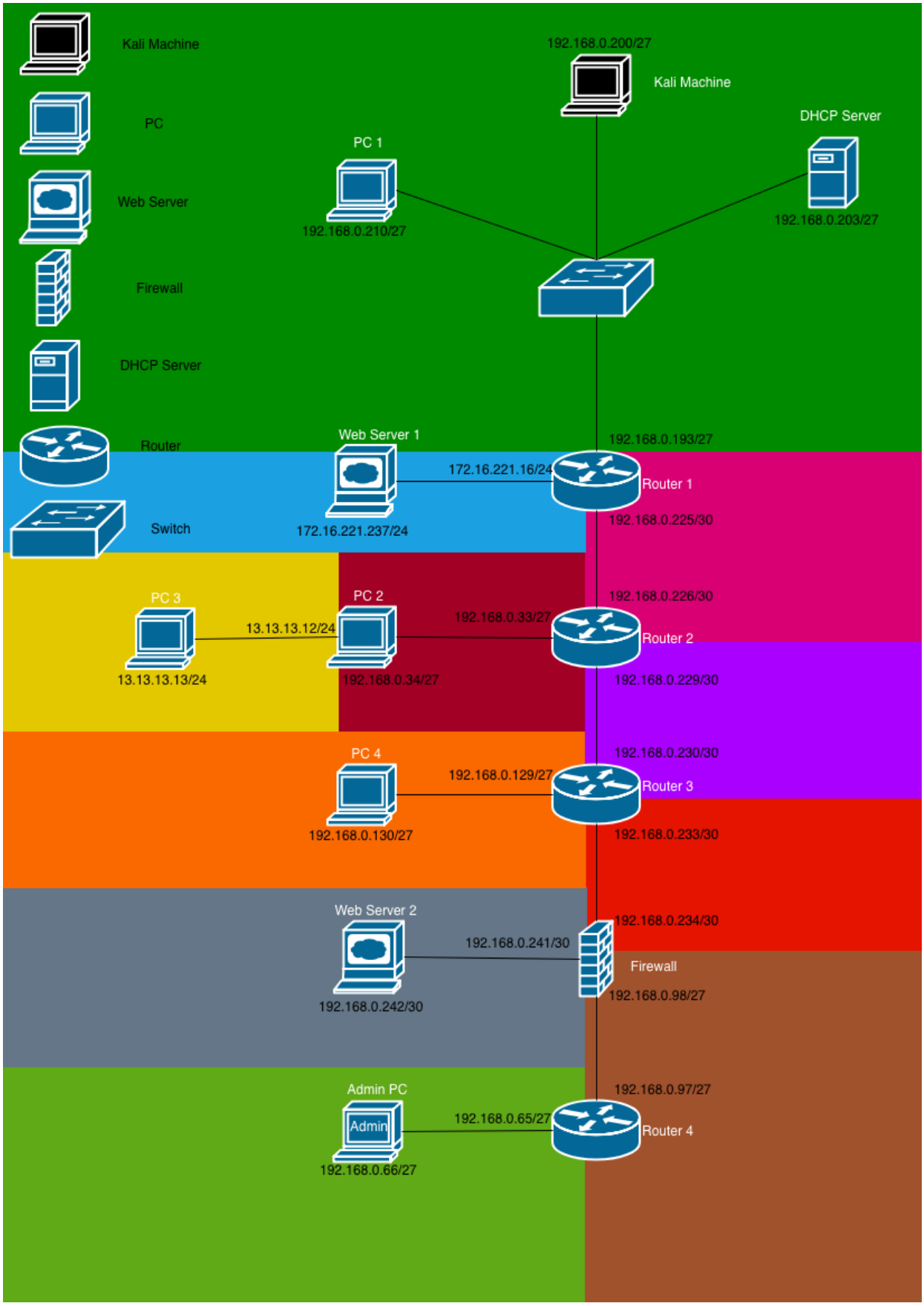
2. Key Terms

Term	Definition
IP Addresses	A numeric address assigned to a network interface so it can be located in a network
Ports	A logical construct that identifies a specific network process or service
HTTP	The hyper text transfer protocol. The protocol used for the world wide web and website
Telnet	A network protocol that allows a user on one computer to log into another computer that is part of the same network.
Network Interface	An interface that communicates between two pieces of equipment or protocols
Subnets	A logical devision of a network
Host	A computer or other device connected to a network
ARP	The address resolution protocol. This maps a devices MAC address to its IP address
MAC address	A unique address given to a physical network component or device
Web Server	Dedicated software and hardware used to provide content to the world wide web
Metasploit	A computer security project that provides information on vulnerabilities in computers and aids in penetration testing
MsfConsole	A popular interface to the Metasploit framework
Bash	A unix shell
Dictionary Attack	Using a large set of words to attempt to guess a password
Password Hash	The result of a password being entered into a complex algorithm to encrypt the original password
SSH	A secure version of telnet
Packet	A formatted unit of data used to carry information in a network
Proxy	A service that acts as an intermediate between two connections
NFS	Network File System. Allows for files to be accessed over a network
Apache	An open source web server
Wordpress	Open source content management system
UDP	User datagram protocol. Used for low latency connections between applications
DHCP	Dynamic Host Configuration Protocol. A management protocol used to dynamically assign IP addresses
SNMP	Simple Network Management Protocol. Used for collecting and managing information on devices in a network
SSH Key	A unique key used to confirm that only a specific computer is accessing a device using SSH

Term	Definition
CMS	Content Management System. Manages the creation and modification of digital content
HTTPS	Hyper text transfer protocol secure. HTTP but data transmission is encrypted
Penetration Testing	An authorised simulated attack on a device or network

3. Network Overview

3.1. Network Diagram



3.2. Subnet Table

Subnet Address	Subnet Mask	Host Range	IPs Used	Broadcast Address
192.168.0.32/27	255.255.255.224	192.168.0.33 - 192.168.0.62	192.168.0.33, 192.168.0.34	192.168.0.63
192.168.0.64/27	255.255.255.224	192.168.0.65 - 192.168.0.94	192.168.0.65, 192.168.0.94	192.168.0.95
192.168.0.96/27	255.255.255.224	192.168.0.97 - 192.168.0.126	192.168.0.97, 192.168.0.98	192.168.0.127
192.168.0.128/27	255.255.255.224	192.168.0.129 - 192.168.0.158	192.168.0.129, 192.168.0.130	192.168.0.159
192.168.0.192/27	255.255.255.224	192.168.0.193 - 192.168.0.222	192.168.0.193, 192.168.0.200, 192.168.0.203, 192.168.0.210	192.168.0.223
192.168.0.224/30	255.255.255.252	192.168.0.225 - 192.168.0.226	192.168.0.225, 192.168.0.226	192.168.0.227
192.168.0.228/30	255.255.255.252	192.168.0.229 - 192.168.0.230	192.168.0.229, 192.168.0.230	192.168.0.231
192.168.0.232/30	255.255.255.252	192.168.0.233 - 192.168.0.234	192.168.0.233, 192.168.0.234	192.168.0.235
192.168.0.240/30	255.255.255.252	192.168.0.241 - 192.168.0.242	192.168.0.241, 192.168.0.242	192.168.0.243
172.16.221.0/24	255.255.255.0	172.16.221.1 - 172.16.221.254	172.16.221.16, 172.16.221.237	172.16.221.255
13.13.13.0/24	255.255.255.0	13.13.13.1 - 13.13.13.254	13.13.13.12, 13.13.13.13	13.13.13.255

3.3. Port Table

Device	Ports
Kali Machine	111 - RPCBind
PC 1	22 - SSH 111 - RPCBind 2049 - NFS 631 - IPP 5353 - ZeroConf
PC 2	22 - SSH 111 - RPCBind 2049 - NFS 631 - IPP 5353 - ZeroConf
PC 3	22 - SSH
PC 4	22 - SSH 111 - RPCBind 2049 - NFS 631 - IPP 5353 - ZeroConf
Admin PC	22 - SSH 111 - RPCBind 2049 - NFS 631 - IPP 5353 - ZeroConf
Web Server 1	80 - HTTP 443 - HTTPS
Web Server 2	22 - SSH 80 - HTTP 111 - RPCBind 631 - IPP 5353 - ZeroConf
DHCP Server	67 - DHCPS
Router 1	22 - SSH 23 - Telnet 80 - HTTP 443 - HTTPS 123 - NTP 161 - SNMP
Router 2	23 - Telnet 80 - HTTP 443 - HTTPS 123 - NTP 161 - SNMP
Router 3	23 - Telnet 80 - HTTP 443 - HTTPS 123 - NTP 161 - SNMP

Device	Ports
Router 4	23 - Telnet 80 - HTTP 443 - HTTPS 123 - NTP 161 - SNMP
Firewall	53 - DNS 80 - HTTP 2601 - DISCP Client 2604 - OSPFD

4. Network Mapping

To map the network, the Kali Linux machine was logged into with the given credentials, and the command *ifconfig* was run. *Ifconfig* shows the network interfaces on the machine, which allowed for the discovery of the IP address of the Kali machine. This can be seen in figure 1.

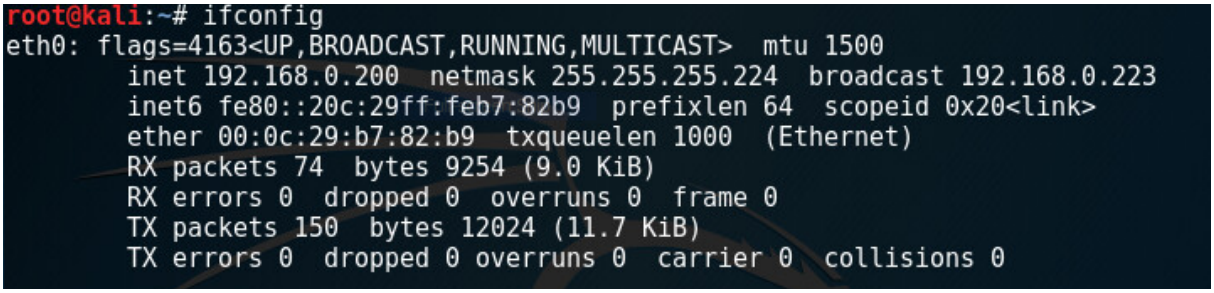


Figure 1

Once the IP address of the Kali machine was found, a tool called *Nmap* was used in order to discover hosts on the network. *Nmap* is a free and open-source security scanner, used to discover hosts and services on a network. (Nmap, 1997) The results of the *Nmap* scan can be seen in Appendix A.

The *Nmap* scan revealed 14 hosts, including the Kali Linux machine. The open ports of these hosts were reviewed, as they give an idea as to what each host is, and hosts with the same open ports were most likely the same kind of device. There were several hosts with port 80 open - HTTP, which means that they can be navigated to on a browser. Navigating to them showed a landing page for a *VyOS* router, with no login functionality, so nothing more could be done through the browser.

Upon inspection of the hosts that appeared to be *VyOS* routers, it was found that they have the telnet port, port 23, open. In order to telnet into the routers, credentials must be known. The default credentials for *VyOS* routers were researched, and it was found that both the username and password was ‘vyos’. (VyOS Wiki, 2018). Telnetting into the routers and using the default credentials was a success, and so full access was gained to the 3 routers discovered by the *Nmap* scan. An example of telnetting into one of the routers can be seen in figure 2.

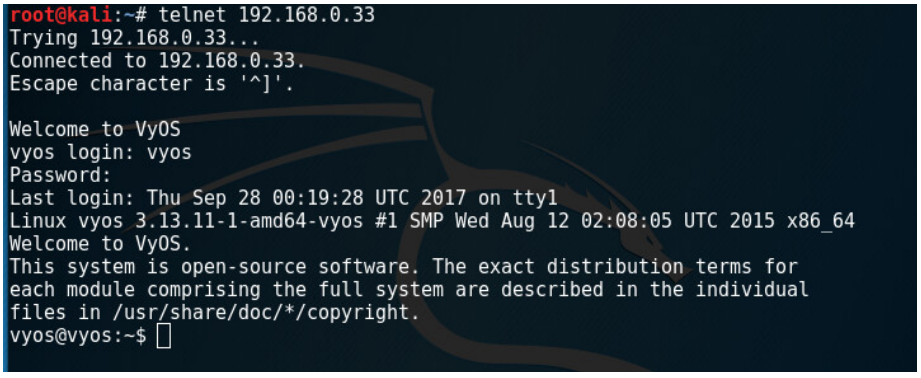


Figure 2

Once logged into the routers, the command *show interfaces* displayed the interfaces enabled on each router, showing the associated IP addresses. Figure 3 shows the interfaces on router 1, figure 4 shows the interfaces on router 2 and figure 5 shows the interfaces on router 3.

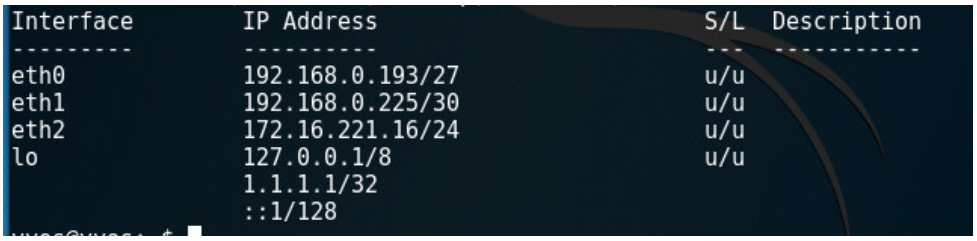


Figure 3

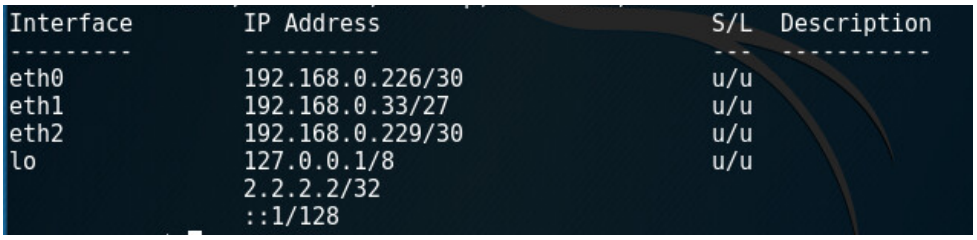


Figure 4

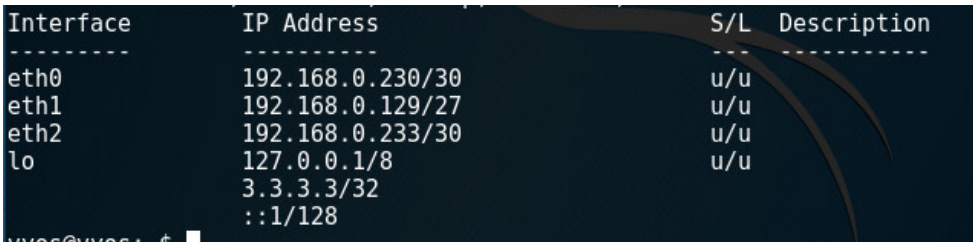


Figure 5

Reviewing the IP addresses on each of the router interfaces showed various subnets were prominent in the network. Subnet calculations were performed to map out these subnets, and determine where each device lied within the subnets. The calculations showed that the Kali machine was on the same subnet as router 1's interface — 192.168.0.193/27, meaning that this interface is the receiving end from the perspective of the Kali machine. This process was used to map out the routers in relation to the Kali Linux machine.

Show arp allowed the ARP tables of the 3 routers to be displayed, which shows the receiving interfaces of neighbouring devices allowing for the discovery of the location of more hosts. (WhatIs.com, 2005) The ARP tables for routers 1,2 and 3 can be seen in figures 6, 7 and 8 respectively.

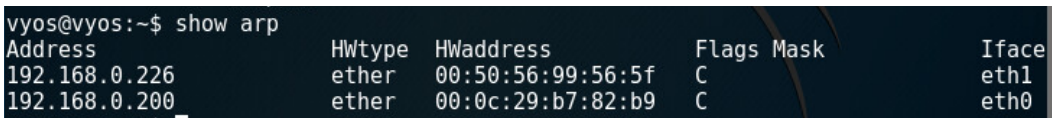
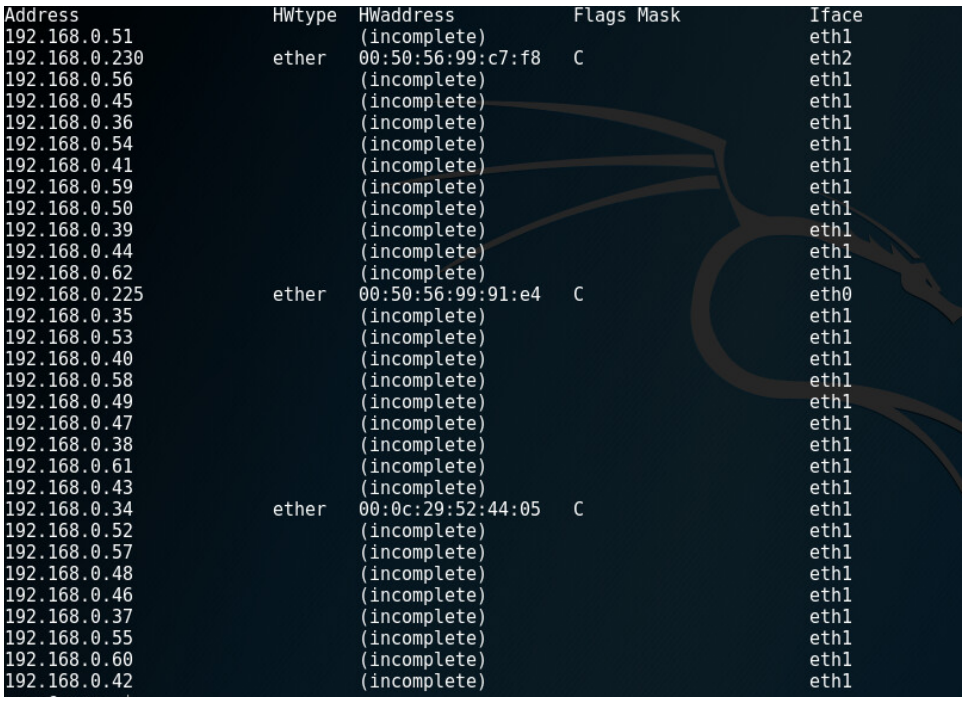
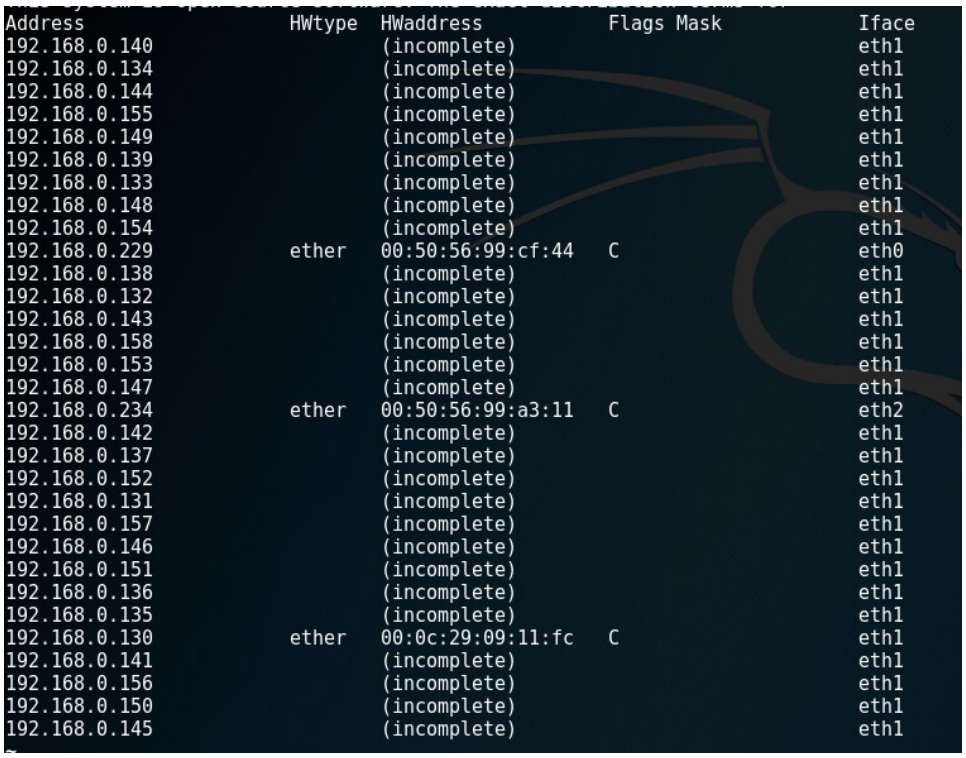


Figure 6



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

Figure 7



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

Figure 8

The ARP table for router 3 showed a host that did not appear on the *Nmap* scan in the initial stages of mapping. A *ping* command was performed against the new found host, 192.168.0.234, but there was no response. It was later discovered that this host was the firewall. The *show ip route* command was used on router 3 to reveal additional subnets, not seen in the initial mapping. These can be accessed via 192.168.0.234, but when pinging the address or the new subnets,

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

C>* 3.3.3.3/32 is directly connected, lo
C>* 127.0.0.0/8 is directly connected, lo
O>* 172.16.221.0/24 [110/30] via 192.168.0.229, eth0, 03:04:33
O>* 192.168.0.32/27 [110/20] via 192.168.0.229, eth0, 03:04:33
O>* 192.168.0.64/27 [110/30] via 192.168.0.234, eth2, 03:04:19
O>* 192.168.0.96/27 [110/20] via 192.168.0.234, eth2, 03:04:26
O   192.168.0.128/27 [110/10] is directly connected, eth1, 03:05:53
C>* 192.168.0.128/27 is directly connected, eth1
O>* 192.168.0.192/27 [110/30] via 192.168.0.229, eth0, 03:04:33
O>* 192.168.0.224/30 [110/20] via 192.168.0.229, eth0, 03:04:33
O   192.168.0.228/30 [110/10] is directly connected, eth0, 03:05:53
C>* 192.168.0.228/30 is directly connected, eth0
O   192.168.0.232/30 [110/10] is directly connected, eth2, 03:05:53
C>* 192.168.0.232/30 is directly connected, eth2
O>* 192.168.0.240/30 [110/20] via 192.168.0.234, eth2, 03:04:28
```

Figure 9

there is no response, suggesting that 192.168.0.234 is a firewall. Figure 9 shows the *show ip route* command on router 3.

With all of the routers before the firewall mapped out, the remaining devices were investigated. Reviewing the *Nmap* scan showed there were 3 hosts with identical ports open. 192.168.0.210, 192.168.0.34 and 192.168.0.130 all appeared to be standard office computers, as research into the open ports alluded to this. The position of 192.168.0.34 and 192.168.0.130 was known from the mapping performed already, however 192.168.0.210 was still to be plotted. The *trace route* command was used to show the devices a packet had to go through in order to reach its

```
root@kali:~# traceroute 192.168.0.210
traceroute to 192.168.0.210 (192.168.0.210), 30 hops max, 60 byte packets
 1  192.168.0.210 (192.168.0.210)  2.056 ms  2.030 ms  1.981 ms
root@kali:~#
```

Figure 10

```
root@kali:~# traceroute 192.168.0.203
traceroute to 192.168.0.203 (192.168.0.203), 30 hops max, 60 byte packets
 1  192.168.0.203 (192.168.0.203)  5.599 ms  5.385 ms  5.165 ms
root@kali:~#
```

Figure 11

destination. It was found that no routers were used in order for the Kali machine to reach the device. This was also true for the unknown device 192.168.0.203. The *trace route* results can be seen in figures 10 and 11.
Router 1 was telnetted into and a *ping* was issued to the two unmapped devices. They were both reachable from the router, so the *show arp* command was run again to see the updated ARP

```
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.226            ether   00:50:56:99:56:5f   C                     eth1
192.168.0.210            ether   00:0c:29:0d:67:c6   C                     eth0
192.168.0.200            ether   00:0c:29:b7:82:b9   C                     eth0
192.168.0.203            ether   00:0c:29:da:42:4c   C                     eth0
```

Figure 12

table. The table showed that 192.168.0.210, 192.168.0.203 and 192.168.0.200 were all connected

via the same interface. This means that there is a switch between router 1 and these devices. The *show arp* results can be seen in figure 12.

There was also the host 192.168.0.242 that had HTTP enabled, so it was navigated to in a browser, which revealed a landing page similar to that of a web server. After the discovery of the web server, a tool called *Nikto* was used to find any vulnerabilities in the web server. *Nikto* is a web server scanner that discovers misconfigurations, dangerous files, and outdated software in web servers. (CIRT.net, 2012) The scan revealed that the web server was vulnerable to a bug known as *Shellshock,* which allows for remote code execution. (Symantec, 2014) The output of the *Nikto* scan can be seen in Appendix B.

MSFconsole was used to exploit the *shellshock* vulnerability. *MSFconsole* is an interface to the *Metasploit framework*. (Offensive Security, 2009) *Metasploit* was configured using the results from the *Nikto* scan, and an attack was launched. Configuring *Metasploit* can be seen in figure 13. The attack resulted in an interactive bash shell being provided, meaning that commands could be executed on the web server.

```
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) >
```

Figure 13

Following the shell being achieved, the *shadow* file was then located. This is located on the linux file system in the folder /etc/ and the file is called *shadow.* This file contains encrypted passwords, such as the root password. This file was downloaded to from the web server to the Kali machine, where the passwords could be cracked using a dictionary attack. The tool *John the Ripper* was used to crack the password hashes, using a common password list. The password cracking was successful and the root password was found to be ‘apple’. The password cracking can be seen in figure 14.

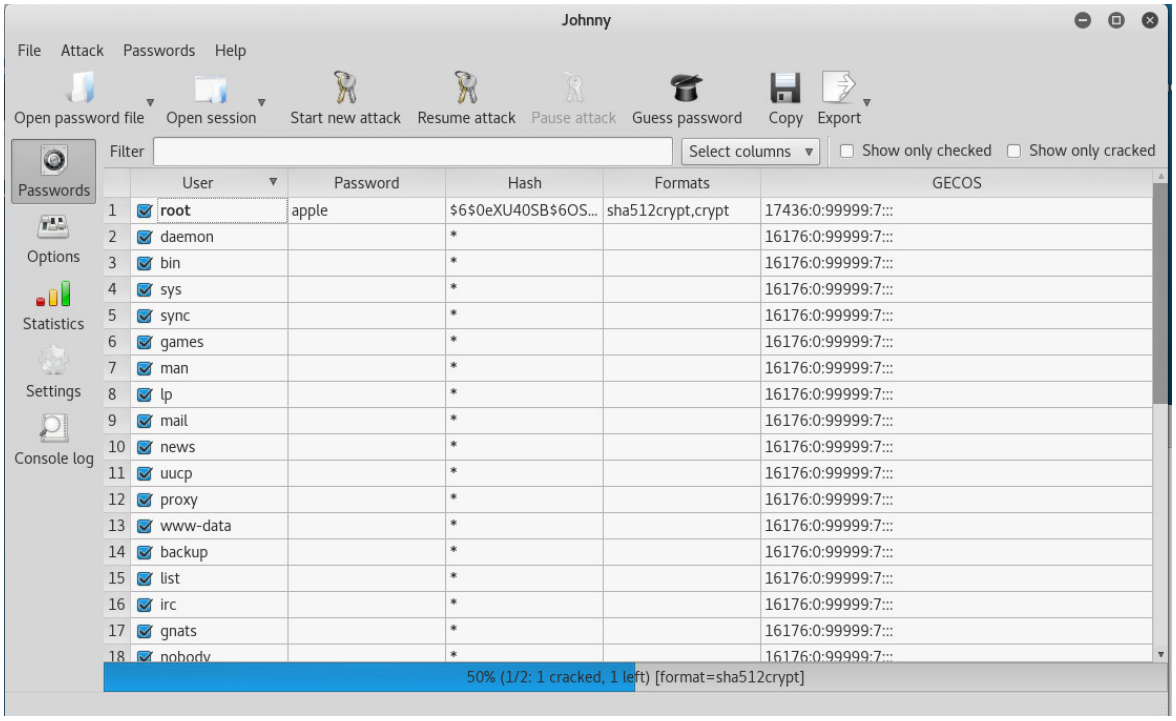


Figure 14

Cracking the root password meant that instead of using the *shellshock* vulnerability to gain access to the web server, it can be SSHed into instead. This means that control over the web server can be achieved without having to set up the *Metasploit* framework every time. The web server was SSHed into and a *ping scan* was ran to see if the web server could reach beyond the firewall. It was found that the web server had access to a new machine, 192.168.0.66.

An *Nmap* scan was used on the subnet 172.16.221.0 to discover a newly found device, 172.16.221.237. This device had HTTP enabled and so was navigated to in the browser, like the previous web server. It was found that this device was also a web server, and so a *Nikto* scan was performed on the server. This web server was not vulnerable to *shellshock* however it was running an outdated version of Apache and was running a Wordpress site. The *Nikto* scan can be seen in Appendix C.

To find out what device the host 192.168.0.203 was, a UDP *Nmap* scan was ran. The results showed that the device was a DHCP server, meaning that it was responsible for assigning IP addresses in the network. (Infoblox, 2018) The UDP scan also showed that many devices had SNMP ports open. The UDP scan results can be seen in Appendix D.

As the device 192.1689.0.210 had NFS enabled, it was mounted as a drive on the Kali machine, on order to explore the file system, where the *shadow* file could be found. As previously, the password hash in this file was cracked and the XADMIN password was found to be ‘plums’. This password was then used to log to 192.168.0.34, where it was found that it had a second interface leading to a new found subnet, 13.13.13.0. An SSH tunnel was created in order to run an *Nmap* scan against the new subnet which led to a new device being found, 13.13.13.13. This is another PC that has a peer to peer connection with 192.168.0.34.

To map beyond the firewall, a *SOCKS5* proxy was created. This is a form of ssh tunnelling, but is designed for web traffic. (Digital Ocean, 2016) Routing the web traffic through 192,168,0,242 meant that access to the firewall’s web portal was accessed. It was discovered that the firewall used was PFsense, and so the default credentials for this were researched, and were found to be ‘admin’ and ‘pfsense’. (pfSense Documentation, 2018) This showed the interface of the firewall leading to the newfound subnet. Setting up the *SOCKS5* proxy in the browser can be seen in figure 15 and figure 16 show it being configured in the terminal.

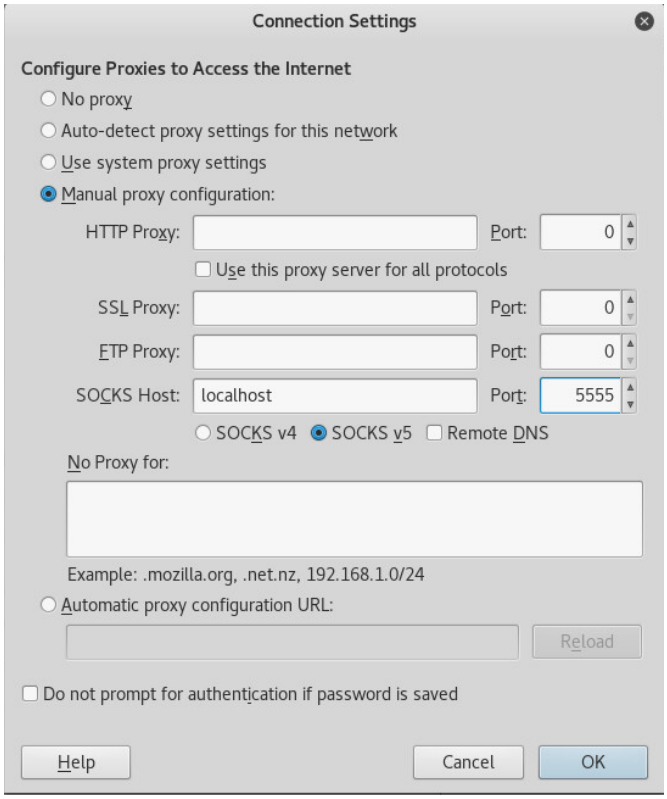


Figure 15

```
root@kali:~# ssh -D 5555 -f -C -q -N root@192.168.0.242
root@192.168.0.242's password:
root@kali:~#
```

Figure 16

Access to the web client of the firewall allowed for the confirmation of the subnet still to be mapped. This could be seen on the interface panel portion of the firewall, which is shown in figure 17.

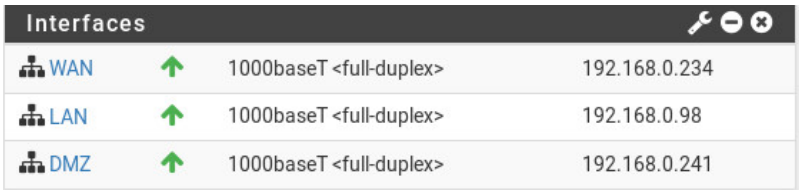


Figure 17

In order to scan the host 192.168.0.66 to discover what it is, an SSH tunnel was set up to allow *Nmap* to scan the machine. An SSH tunnel allows the network traffic to 'tunnel' from the Kali machine through 192.168.0.242 to 192.168.0.66. The SSH tunnel was set up and can be seen in Appendix E.

Once the SSH tunnel was set up, an *Nmap* scan was run on the host 192.168.0.66, and it was found to to be a regular computer. It also had the NFS port open (port 2049) which meant that the machine could be mounted to the Kali machine as a drive, so that the file system could be explored. The results of the *Nmap* scan can be seen in figure 18.

```
Nmap scan report for 192.168.0.66
Host is up (0.0089s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
2049/tcp open  nfs
```

Figure 18

When exploring the file system, the Kali machines SSH key was added to the machine in order to allow SSH access. Once this was achieved, the machine was SSHed into and a trace route was performed to the Kali machine, which showed that there was another router between the machine and the firewall. The router was telnetted into, and the *show arp* command was performed in order to complete the mapping of the network. The results of the *show arp* command can be seen in figure 19.

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

Figure 19

5. Security Evaluation

5.1. Routers

5.1.1. Use of Telnet

The Telnet port (port 23) is open on all routers and allows telnet connections. This is very insecure as telnet has no form of transmission protection and just transmits data in clear text. (The Packet University, 2012) An attacker could listen on the network and intercept telnet connections, and read the data that is being transferred. In order to prevent this, the routers should enable SSH and disable telnet. SSH uses encryption and so the traffic cannot be easily read by interception.

5.1.2. Using Default Credentials

The routers use their default credentials for logging in, meaning that an attacker could go and research the default credentials for the router type, and then login. It is recommended that as soon as any new device is set up, the default credentials are changed to something more secure.

5.2. Web Server

5.2.1. ShellShock

The web server is running an outdated version of Apache that is vulnerable to a vulnerability called Shellshock. This is a bash bug that occurs when an attacker forces an application to send a malicious environment variable to bash. A scan on the web server showed that the status cgi script was able to be used to launch an attack resulting in a remote bash shell. As the vulnerability occurs on an outdated version of Apache, to mitigate the vulnerability, the Apache server should be updated. Updating Apache can be seen in figure 20.

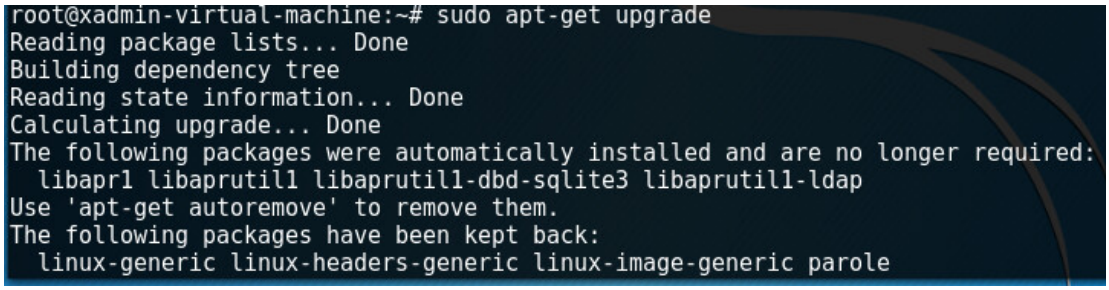


Figure 20

5.2.2. Word Press

One of the web servers hosts a Wordpress site that is out of date. Wordpress has been found to be extremely vulnerable and so it is recommended that it is removed and replaced with a better CMS. If Wordpress must be kept, then it should be kept up to date with regular checks for updates.

5.3. Office Computers

5.3.1. NFS

The office computers all have NFS enabled, which means that all computers can be mounted as drives, and so their file systems can be explored. Some computers even have read and write access enabled, so the files can be edited. This should be changed so that no computer has write access, and that key files are blocked from viewing.

5.3.2. Password Policy

The passwords for the ‘xadmin’ account on all of the office computers are the same, and so if access can be gained from one machine, all are vulnerable. The company should employ a password policy so that there is no password reuse, and that passwords are more secure.

5.3.3. SSH Brute Force

There is no protection against SSH brute force attacks, which means that an attacker can figure out the password to SSH into any machine, as all they need is time. SSH brute forcing can be disabled by adding a service such as *Fail2ban* in order to automatically ban IP addresses that continually try to SSH with incorrect passwords. (Security Trails, 2018)

5.4. Firewall

5.4.1. Misconfiguration

The firewall has been incorrectly configured to allow the web server access to the administrator PC, which allowed for the discovery of the entire network. This vulnerability should be considered the most critical as it allowed for other vulnerabilities to be exploited. The rule that allows the web server to communicate with the administrator PC should be removed. The incorrect rule can be seen in figure 21.

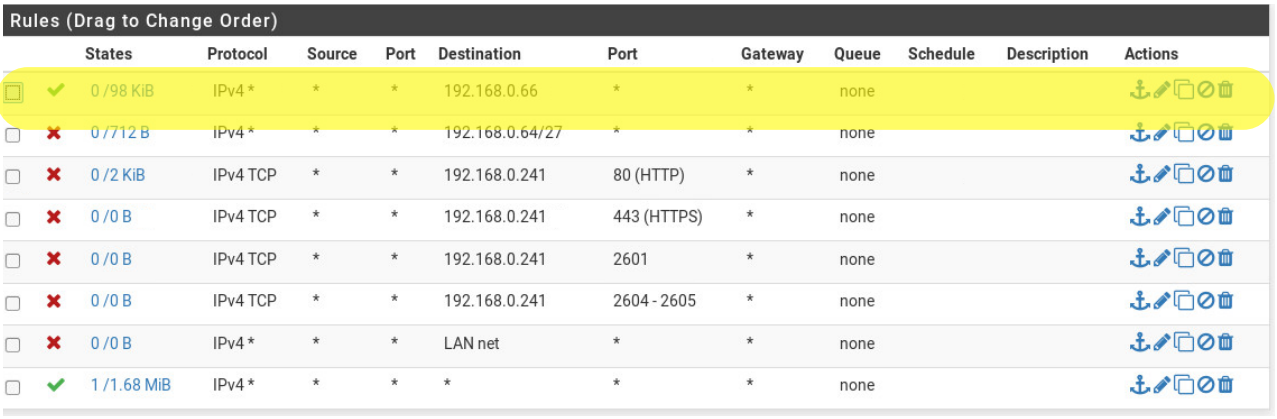


Figure 21

5.4.2. Default Credentials

The firewall uses the default credentials for PFsense, and so it can be logged into by anyone who has the ability to research the default credentials. These credentials should be changed and should follow a strong password policy.

5.4.3. Insecure Transmission

PFsense uses HTTP and so an attacker could intercept traffic and potentially steal credentials for the firewall. The firewall should be upgraded to uses HTTPS so that the traffic is encrypted. Enabling HTTPS can be seen in figure 22.

Figure 22

5.4.4. Default Session Timeout

The firewall is using the default setting for session timeout, which means that an attacker could potentially have 4 hours to attack if they can hijack the session. This should be changed so that the session will timeout much quicker.

5.5. General Issues

5.5.1. Password Policy

Throughout the network, it is clear that poor passwords are being used, which means that they have can be cracked or guessed easily. It is recommended that ACME Inc. immediately implement a password policy that is appropriate. This will ensure that the passwords will be strong and secure. For more information on strong password policies, see the NCSC's password guidelines. (NCSC, 2016)

5.6. Conclusion

The overall security of the network is extremely poor and major changes need to made in order for it to be considered acceptable. Any attacker with a basic knowledge of penetration testing could find weaknesses in the network and exploit them. ACME Inc should consider hiring a network security specialist to fix, improve and maintain the security of their network.

6. Critical Evaluation

The network design shows a decent attempt at correct configuration and sensible design. The network uses variable length subnet masks in order to divide the network into sub categories, meaning that IP addresses are not wasted. Subnets have been configured to allow for expansion in areas that may need it, with also ensuring that IP addresses are not wasted. The firewall was almost correctly configured to not allow the main network access to the web server and administrator section. With some small changes, this could be a good design. Having the peer to peer connection between two PCs is an interesting choice, as traffic has to flow through a machine, meaning that it may be less efficient at other tasks. It may be worth considering moving the 13.13.13.0 subnet to attach to a router. As the 13.13.13.0 subnet is a class A IP address, it can have up to 65,534 hosts, and so there is much room for expansion.

The network has a main flaw in that if one router goes down, the whole network will be broken, as there is no redundancy taken into consideration. It is recommended that the network be adjusted so that there can be no single point of failure.

7. Conclusions

Overall ACME Inc. were correct in their concerns about their network and were right to request an infrastructure evaluation. In future they should ensure that their network manager keeps an accurate log and appropriate documentation, as to not have a repeat situation. The network shows signs of competency, however, the individual who configured it showed serious lack of knowledge of networks security, and showed poor judgement in setting up sections of the network.

8. References

Techopedia. 2018. What is Network Infrastructure. [ONLINE] Available at: https://www.techopedia.com/definition/16955/network-infrastructure. [Accessed 12 December 2018].

Offensive Security. 2013. What is Kali Linux?. [ONLINE] Available at: https://docs.kali.org/introduction/what-is-kali-linux. [Accessed 12 December 2018].

Nmap. 1997. The Network Mapper. [ONLINE] Available at: https://nmap.org. [Accessed 12 December 2018].

WhatIs.com. 2005. What is Address Resolution Protocol?. [ONLINE] Available at: https://searchnetworking.techtarget.com/definition/Address-Resolution-Protocol-ARP. [Accessed 12 December 2018].

CIRT.net. 2012. Nikto. [ONLINE] Available at: https://cirt.net/Nikto2. [Accessed 12 December 2018].

Offensive Security. 2009. MSFconsole. [ONLINE] Available at: https://www.offensive-security.com/metasploit-unleashed/msfconsole/. [Accessed 12 December 2018].

Digital Ocean. 2016. How to Route Web Traffic Securely Without a VPN Using a SOCKS Tunnel. [ONLINE] Available at: https://www.digitalocean.com/community/tutorials/how-to-route-web-traffic-securely-without-a-vpn-using-a-socks-tunnel. [Accessed 12 December 2018].

Infoblox. 2018. What is a DHCP Server?. [ONLINE] Available at: https://www.infoblox.com/glossary/dhcp-server/. [Accessed 12 December 2018].

pfSense Documentation. 2018. Installing pfSense. [ONLINE] Available at: https://www.netgate.com/docs/pfsense/install/installing-pfsense.html. [Accessed 12 December 2018].

The Packet University. 2012. What's Wrong with Telnet?. [ONLINE] Available at: https://packetu.com/2012/04/17/whats-wrong-with-telnet/. [Accessed 12 December 2018].

Security Trails. 2018. Mitigating SSH based attacks. [ONLINE] Available at: https://securitytrails.com/blog/mitigating-ssh-based-attacks-top-15-best-security-practices. [Accessed 12 December 2018].

NCSC. 2016. Password Guidance. [ONLINE] Available at: https://www.ncsc.gov.uk/guidance/password-guidance-simplifying-your-approach. [Accessed 12 December 2018].

Symantec. 2014. ShellShock. [ONLINE] Available at: https://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability. [Accessed 12 December 2018].

Appendices

A. Nmap Scan

[Due to the nature of the test environment, the actual text output from the Nmap scan could not be retrieved, and instead, only screenshots of the output could be retrieved. These screenshots were put through optical character recognition software, which has allowed for the text to be inserted, however there may be some inaccuracies.]

```
Nmap scan report for 192.168.0.33
Host is up (0.0018s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.34
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
111/tcp    open       rpcbind
2049/tcp   open       nfs

Nmap scan report for 192.168.0.129
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.130
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
111/tcp    open       rpcbind
2049/tcp   open       nfs

Nmap scan report for 192.168.0.225
Host is up (0.00068s latency).
Not shown: 996 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.226
Host is up (0.0016s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
```

```
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.229
Host is up (0.0016s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.230
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.233
Host is up (0.0029s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https

Nmap scan report for 192.168.0.242
Host is up (0.0031s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
80/tcp     open       http
111/tcp    open       rpcbind

Nmap scan report for 192.168.0.193
Host is up (0.00039s latency).
Not shown: 996 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
23/tcp     open       telnet
80/tcp     open       http
443/tcp    open       https
MAC Address: 00:50:56:99:6C:E2 (VMware)

Nmap scan report for 192.168.0.203
Host is up (0.00056s latency).
All 1000 scanned ports on 192.168.0.203 are closed
MAC Address: 00:0C:29:DA:42:4C (VMware)

Nmap scan report for 192.168.0.210
Host is up (0.00018s latency).
Not shown: 997 closed ports
PORT       STATE      SERVICE
22/tcp     open       ssh
111/tcp    open       rpcbind
```

```
2049/tcp   open       nfs
MAC Address: 00:oC:29:oD:67:C6 (VMware)

Nmap scan report for 192.168.0.200
Host is up (0.0000010s latency).
Not shown: 999 closed ports
PORT       STATE      SERVICE
111/tcp    open       rpcbind

Nmap done - 256 IP addresses (14 hosts up) scanned in 46.80 seconds
```

B. Nikto Scan for Web Server

[Due to the nature of the test environment, the actual text output from the Nikto scan could not be retrieved, and instead, only screenshots of the output could be retrieved. These screenshots were put through optical character recognition software, which has allowed for the text to be inserted, however there may be some inaccuracies.]

```
- Nikto v2.1.6
---------------------------------------------------------
+ Target IP: 192.168.0.242
+ Target Hostname: 192.168.0.242
+ Target Port: 80
---------------------------------------------------------
+ Server: Apache/2.4.10 (Unix)
+ Server leaks inodes via ETags, header found with file /, fields: ox650
ox558adddob8740
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to
the MIME type
+ Apache/2.4.10 appears to be outdated (current is at least Apache/
2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: POST, OPTIONS, GET, HEAD, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is
vulnerable to XST
+ Uncommon header 'nikto-added-cve-2014-6278' found, with contents: true
+ OSVDB-112004: /cgi-bin/status: Site appears vulnerable to the
'shellshock' vulnerability (http://cve.mitre.org/cgi-bin/cvename.cgi?
name=CVE-2014-6271).
+ OSVDB-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
---------------------------------------------------------
+ 1 host(s) tested
```

C. Nikto Scan for Wordpress Server

[Due to the nature of the test environment, the actual text output from the Nikto scan could not be retrieved, and instead, only screenshots of the output could be retrieved. These screenshots were put through optical character recognition software, which has allowed for the text to be inserted, however there may be some inaccuracies.]

```
- Nikto v2.1.6
---------------------------------------------
+ Target IP: 172.16.221.237
+ Target Hostname: 172.16.221.237
+ Target Port: 86
---------------------------------------------
+ Server: Apache/2.2.22 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, inode: 45778,
size: 177, mtime: Tue Apr 29 66:43:57 2614
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
user agent to protect against some forms
of XSS
+ The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in
a different fashion to the MIME type
+ Apache/2.2.22 appears to be outdated (current is at least Apache/
2.4.12). Apache 2.6.65 (final release) and 2.2.29
are also current.
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows
attackers to easily brute force file names. See
http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following
alternatives for 'index' were found: index.html
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
+ OSVDB-3233: /icons/README: Apache default file found.
+ Retrieved x-powered-by header: PHP/5.3.16-1ubuntu3.26
+ wordpress/: A Wordpress installation was found.
+ 8346 requests: 6 error(s) and 11 item(s) reported on remote host
---------------------------------------------

+ 1 host(s) tested
```

D. UDP Nmap Scan

[Due to the nature of the test environment, the actual text output from the Nmap scan could not be retrieved, and instead, only screenshots of the output could be retrieved. These screenshots were put through optical character recognition software, which has allowed for the text to be inserted, however there may be some inaccuracies.]

```
Nmap scan report for 192.168.0.33
Host is up (0.0025s latency).
Not shown: 948 closed ports, 50 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 (0.0044s 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 192.168.0.129
Host is up (0.0031s latency).
Not shown: 883 closed ports, 115 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 (0.0055s 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 (0.00185 latency).
Not shown: 998 closed ports
PORT  STATE      SERVICE
123/udp    open  ntp
161/udp    open  snmp

Nmap scan report for 192.168.0.226
Host is up (0.0028s latency).
Not shown: 890 closed ports, 108 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 (0.0029s latency).
Not shown: 900 closed ports, 98 open]filtered ports
```

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

Nmap 5can report for 192.168.0.230
Host is up (0.0040s latency).
Not shown: 943 closed ports, 55 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 (0.0043s latency).
Not shown: 936 closed ports, 62 open]filtered ports
PORT  STATE      SERVICE
123/udp    open  ntp
161/udp    open  snmp

Nmap scan report for 192.168.0.242
Host is up (0.0070s 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 (0.0015s latency).
Not shown: 998 closed ports
PORT  STATE      SERVICE
123/udp    open  ntp
161/udp    open  snmp
MAC Address: 00:50:56:99:6C:E2 (VMware)

Nmap scan report for 192.168.0.203
Host is up (0.0016s latency).
Not shown: 999 closed ports
PORT  STATE           SERVICE
67/udp     open|filtered dhcps
MAC Address: 00:oC:29:DA:42:4C (VMware)

Nmap scan report for 192.168.0.210
Host is up (0.0017s 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:oC:29:oD:67:C6 (VMware)

Nmap scan report for 192.168.0.200
Host is up (0.000042s latency).
Not shown: 999 closed ports
PORT  STATE      SERVICE
```

```
111/udp    open  rpcbind
```

E. SSH Tunnel Setup

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

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

Last login: Thu Sep 28 03:58:12 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# nano /etc/ssh/sshd_config
root@xadmin-virtual-machine:~# exit
logout
Connection to 192.168.0.242 closed.
root@kali:~# ssh -w 0:0 root@192.168.0.242
root@192.168.0.242's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

Last login: Thu Sep 28 04:04:55 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# ip addr add 1.1.1.2/30 dev tun0
root@xadmin-virtual-machine:~# ip link se tun0 up
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

```
root@kali:~# ip addr add 1.1.1.1/30 dev tun0
root@kali:~# ip link set tun0 up
root@kali:~# route add -net 192.168.0.64/27 tun0
root@kali:~#
```

F. Subnet Calculations

Subnet Address: 192.168.0.0
Subnet Mask: 255.255.255.224
Convert last 8 bits to binary

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

Number of borrowed bits = 3 so
CIDR Notation: 8 + 8 + 8 + 3 = /27
Number of Networks: $2^3 = 8$
Number of Hosts: $2^5 = 32$ so magic number is 32
Usable Hosts: 32 - 2 = 30

Subnet Address: 192.168.0.224
Subnet Mask: 255.255.255.252
Convert last 8 bits to binary

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

Number of borrowed bits = 6 so
CIDR Notation: 8 + 8 + 8 + 7 = /30
Number of Networks: $2^6 = 64$
Number of Hosts: $2^2 = 4$ so magic number is 4
Usable Hosts: 32 - 2 = 30

Subnet Address: 172.16.221.0
Subnet Mask: 255.255.255.0
Convert last 8 bits to binary

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

Number of borrowed bits = 0 so
CIDR Notation = 8 + 8 + 8 + 0 = /24
Number of Networks: $2^0 = 1$
Number of Hosts: $2^8 = 256$ so magic number is 256
Usable Hosts: 256 - 2 = 254

Subnet Address: 13.13.13.0
Subnet Mask: 255.255.255.0
Convert last 8 bits to binary

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

Number of borrowed bits = 0 so
CIDR Notation = 8 + 8 + 8 + 0 = /24
Number of Networks: $2^0 = 1$
Number of Hosts: $2^8 = 256$ so magic number is 256
Usable Hosts: 256 - 2 = 254