**Name :- Dishant Sonani**

**University :- Parul University**

**Major :- Computer Science**

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

**Company :- Hack Secure**

**Domain :- Cyber Security**

**Mentor :- Mr. Nishant Prajapati**

**Assistant Mentor :- Mr. Aman Panday**

**Coordinator :- Mr. Shivam Kapoor**

**Email :- dishantsonani007@gmail.com**

* **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**

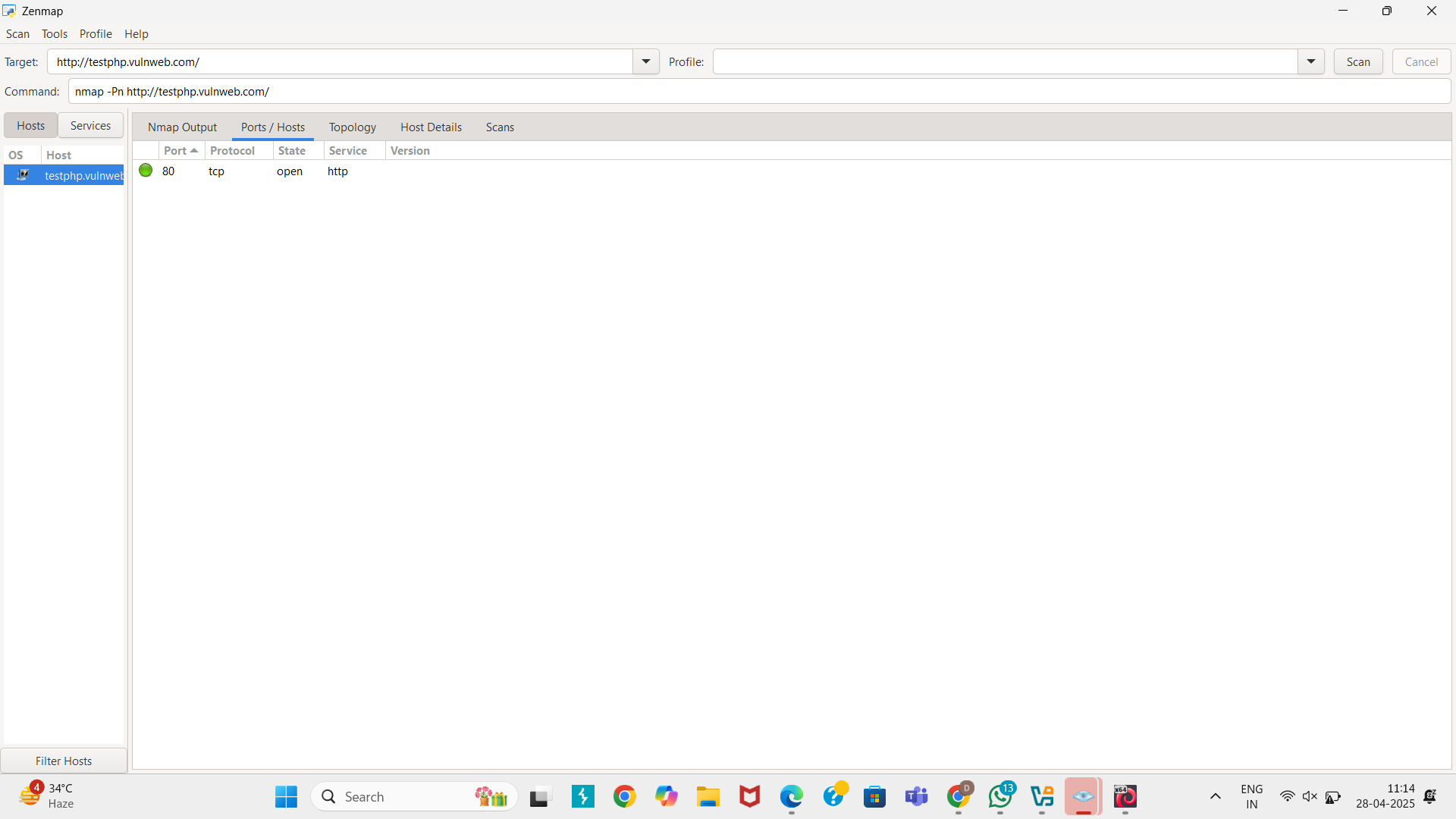
**During my internship, I was involved in the following key tasks:**

* **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.**

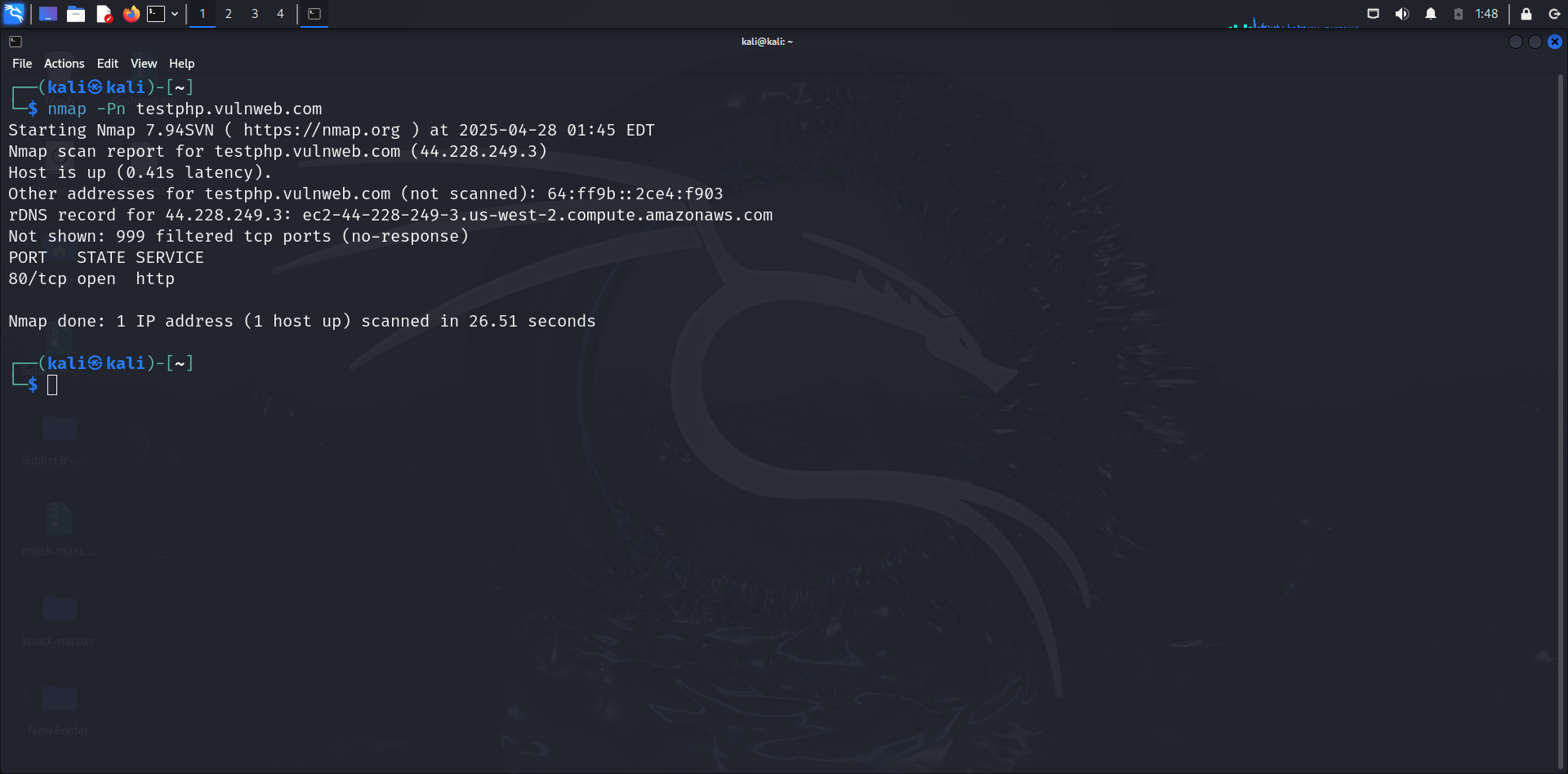
**A) TASK LEVEL INTERMIDIATE**

**1. Find all the ports that are open on the website** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)

* **Tool :- Nmap**
* **Command :- nmap -Pn testphp.vulnweb.com**

****

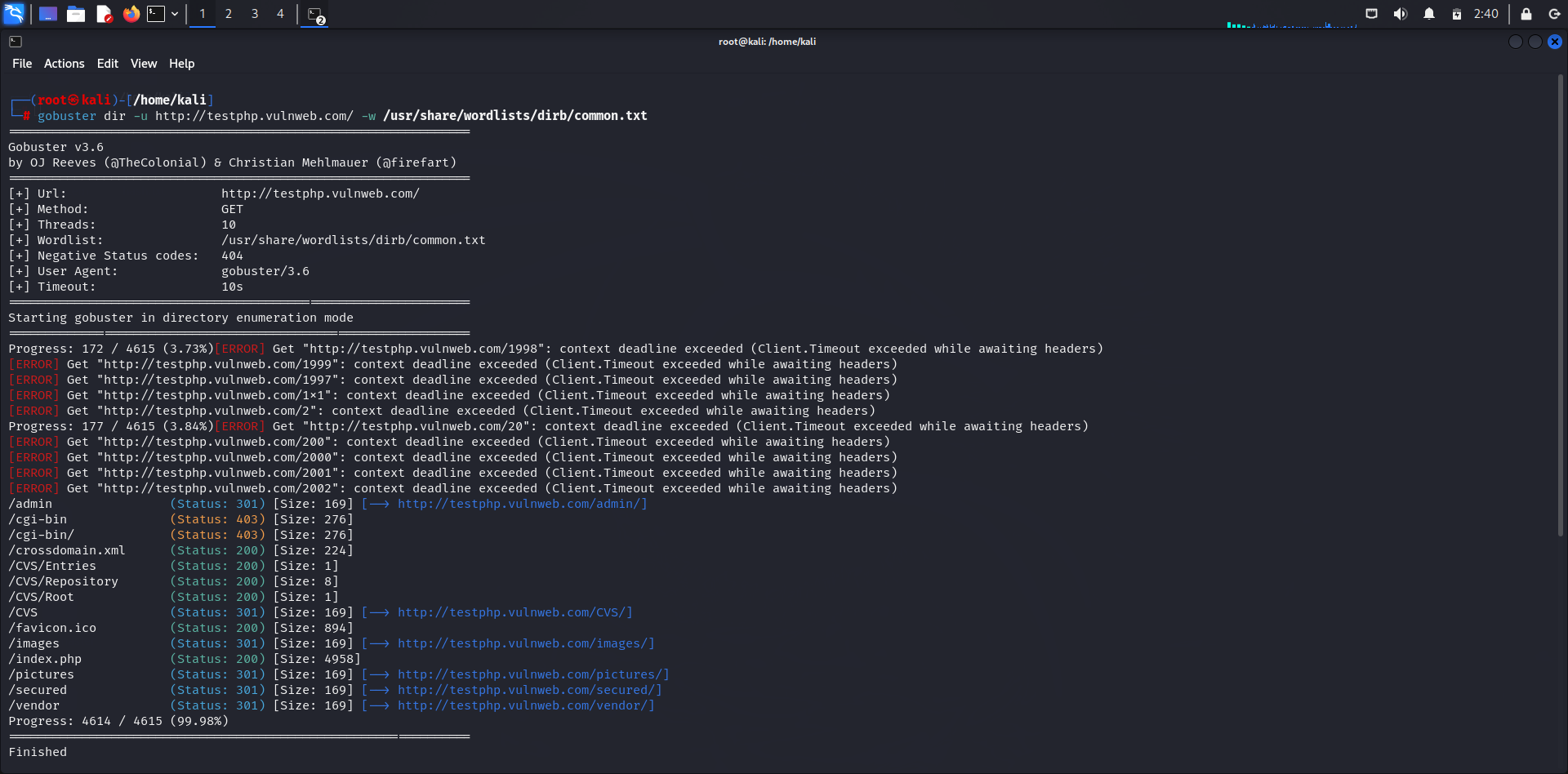
* **We can also use a Kali Linux where Nmap is installed**
* **-Pn :- Tells Nmap to skip the host discovery phase**

****

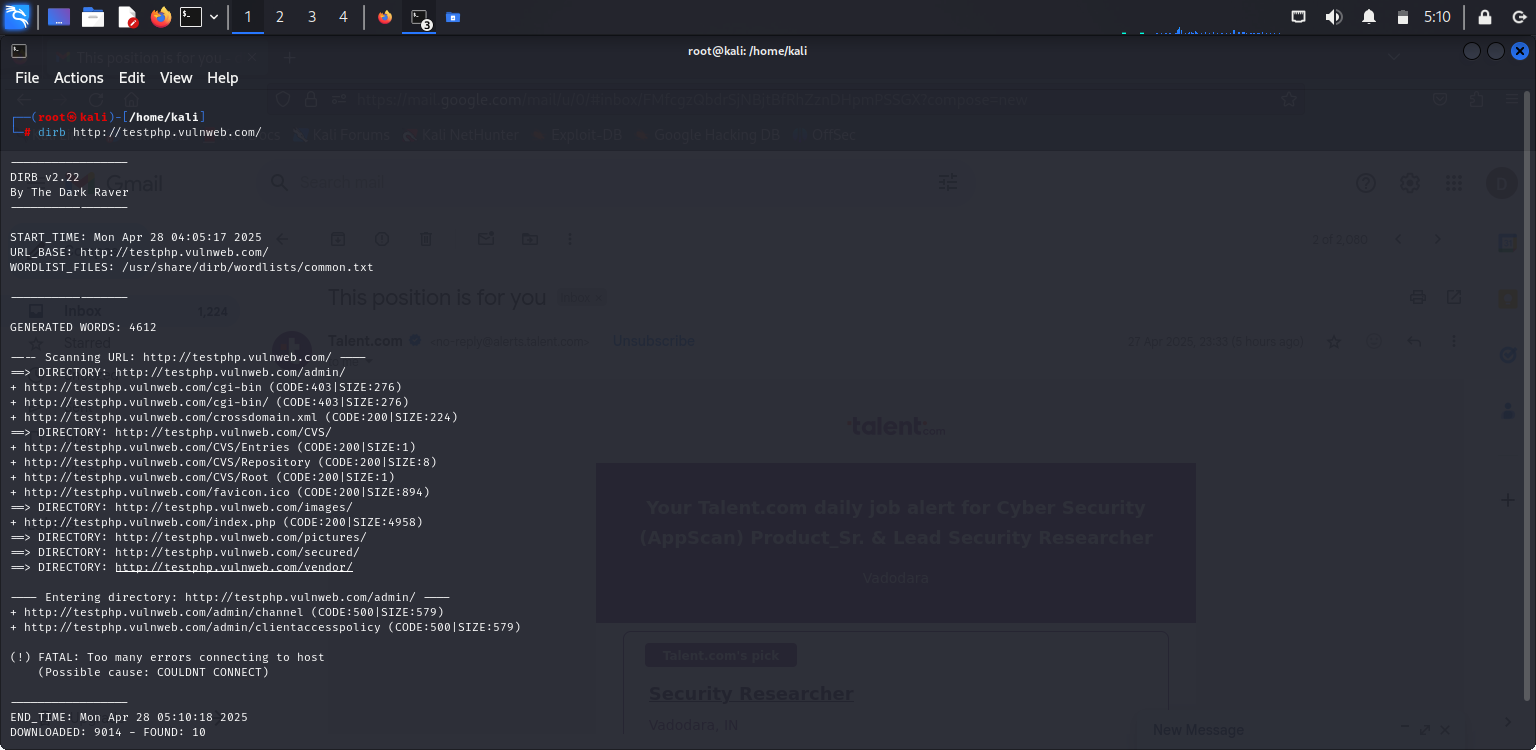
* **Shows that TCP port 80 (HTTP) is open, meaning a web server is running.**
* **Host Status :- Up**
* **Open Port :- 80 (HTTP)**
* **Total Scan Time :- 26.51 Seconds**

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

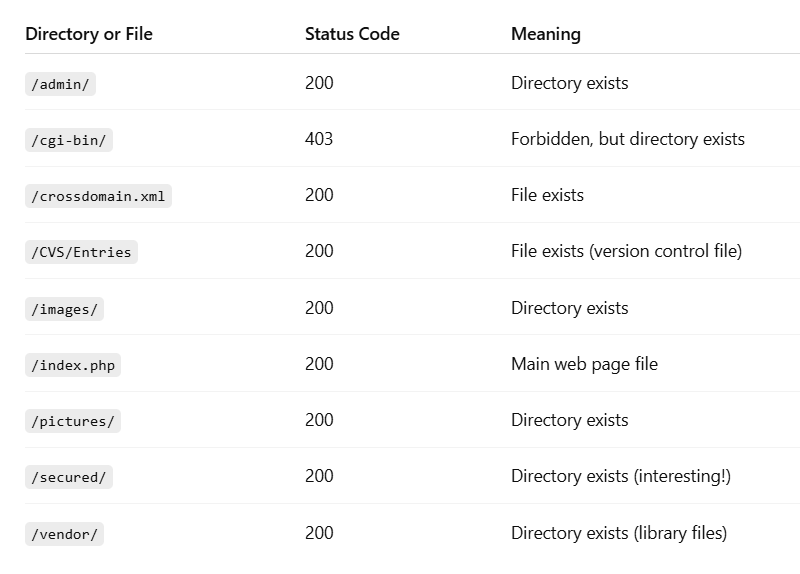
* **We use Gobuster to scan** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/) **with the common directory wordlist.**
* **The scanner attempted 4615 directory names.**
* **301 Redirect means directory exists but redirects.**
* **403 Forbidden means directory exists but you don't have permission to access directly — sometimes you can still exploit it.**
* **200 OK means file/directory is accessible.**

****

* **We use Dirb to scan** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/) **with the common directory wordlist.**
* **Command :- dirb** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **Wordlist used :- /usr/share/dirb/wordlists/common.txt**
* **Total words generated :- 4612**

****

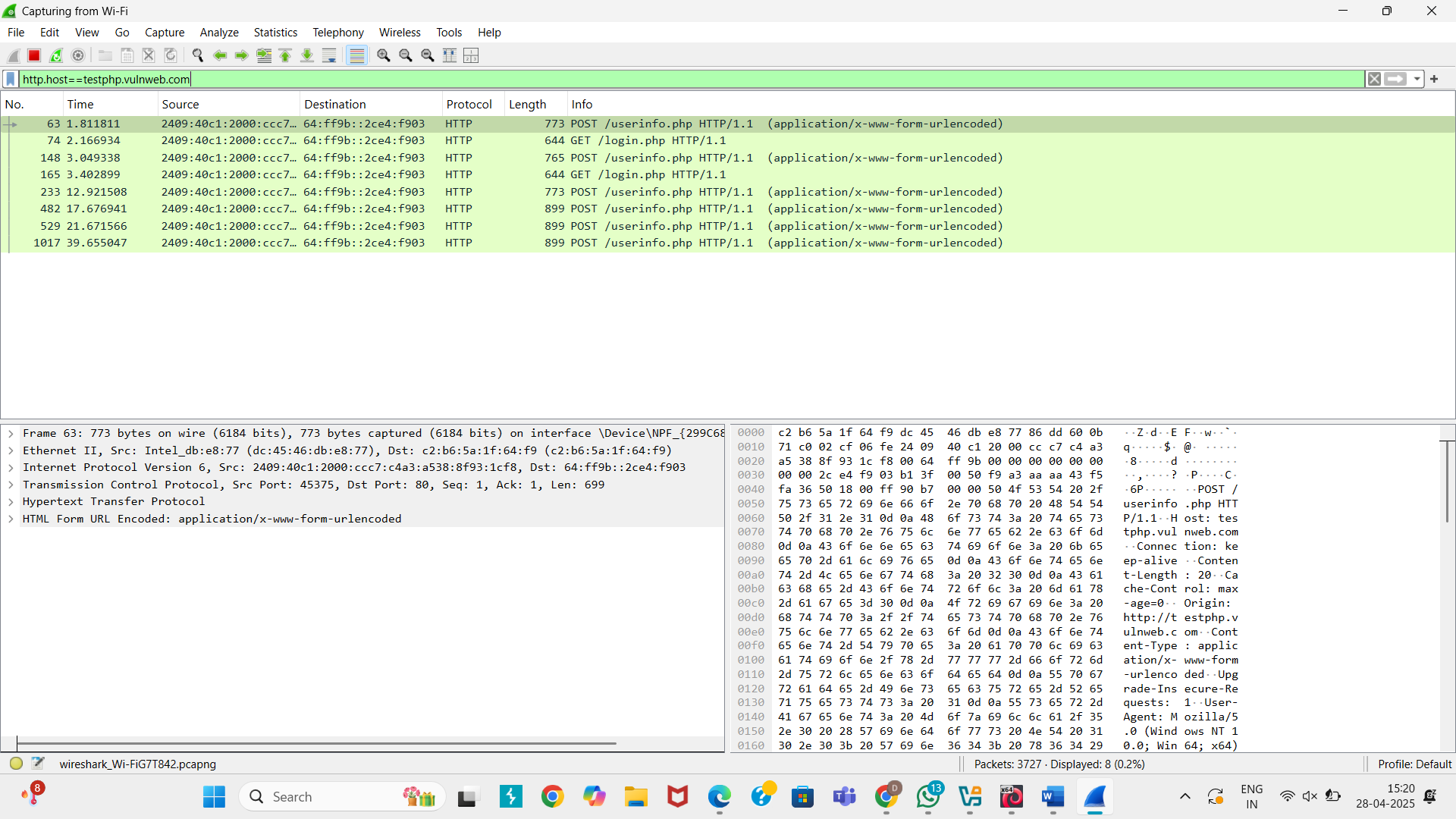
* **The /admin/ directory can be attacked with SQL Injection or brute force.**
* **The /secured/ directory may leak sensitive information.**

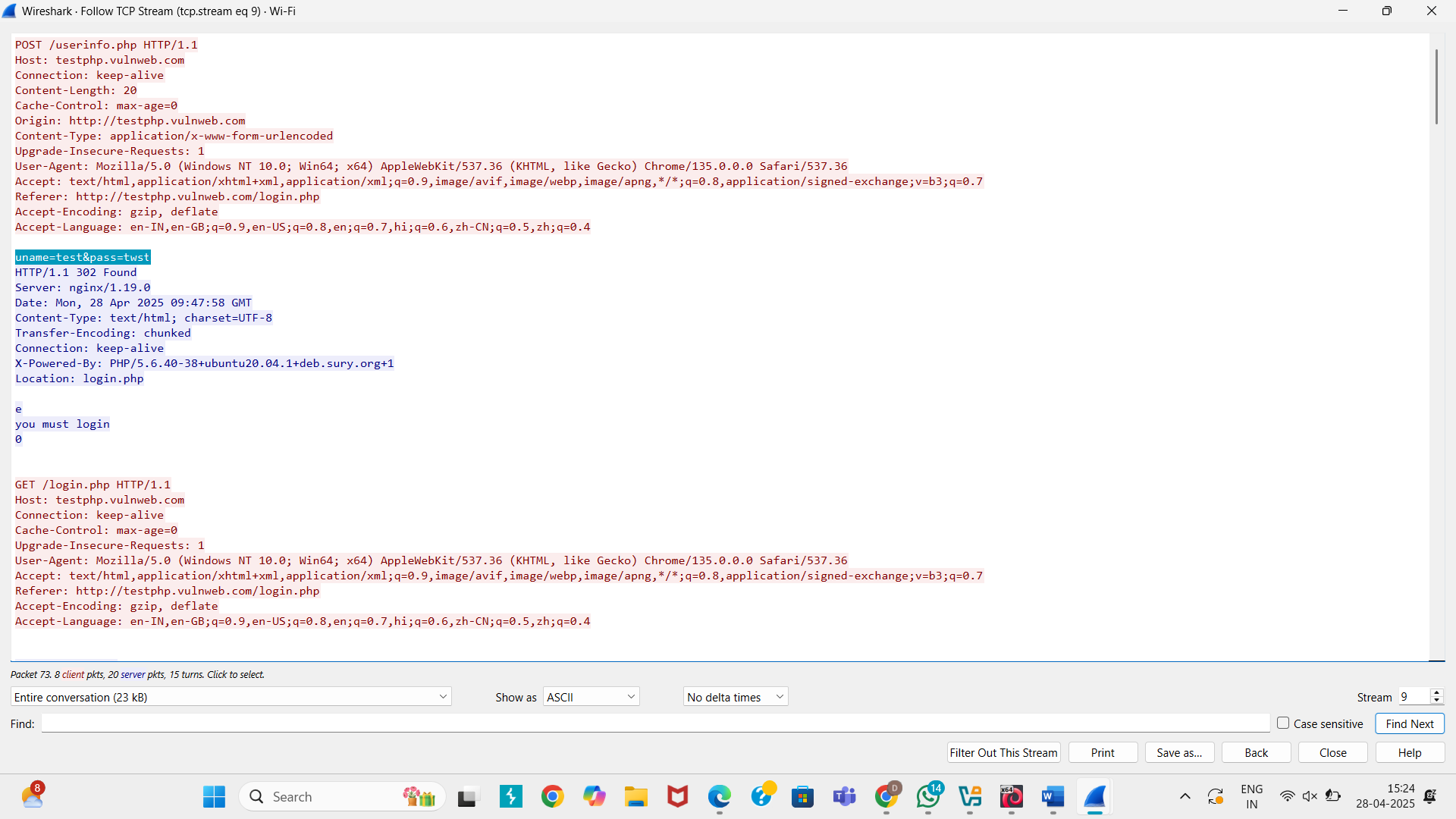
****

* **This is all the directories that are present in the website** [**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)

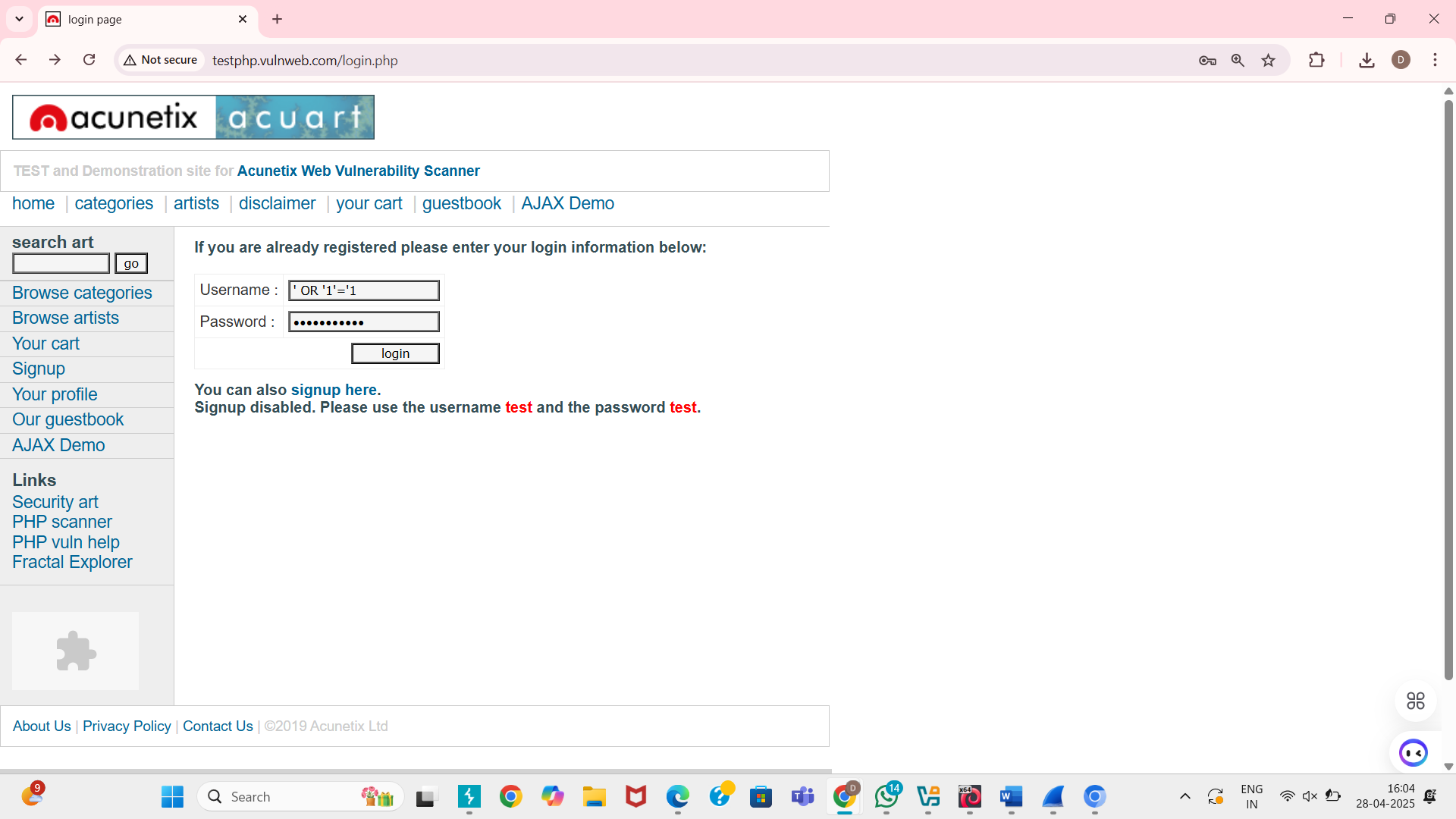
**3. Make a login in the website http://testphp.vulnweb.com/ and intercept the network traffic using Wireshark and find the credentials that were transferred through the network.**

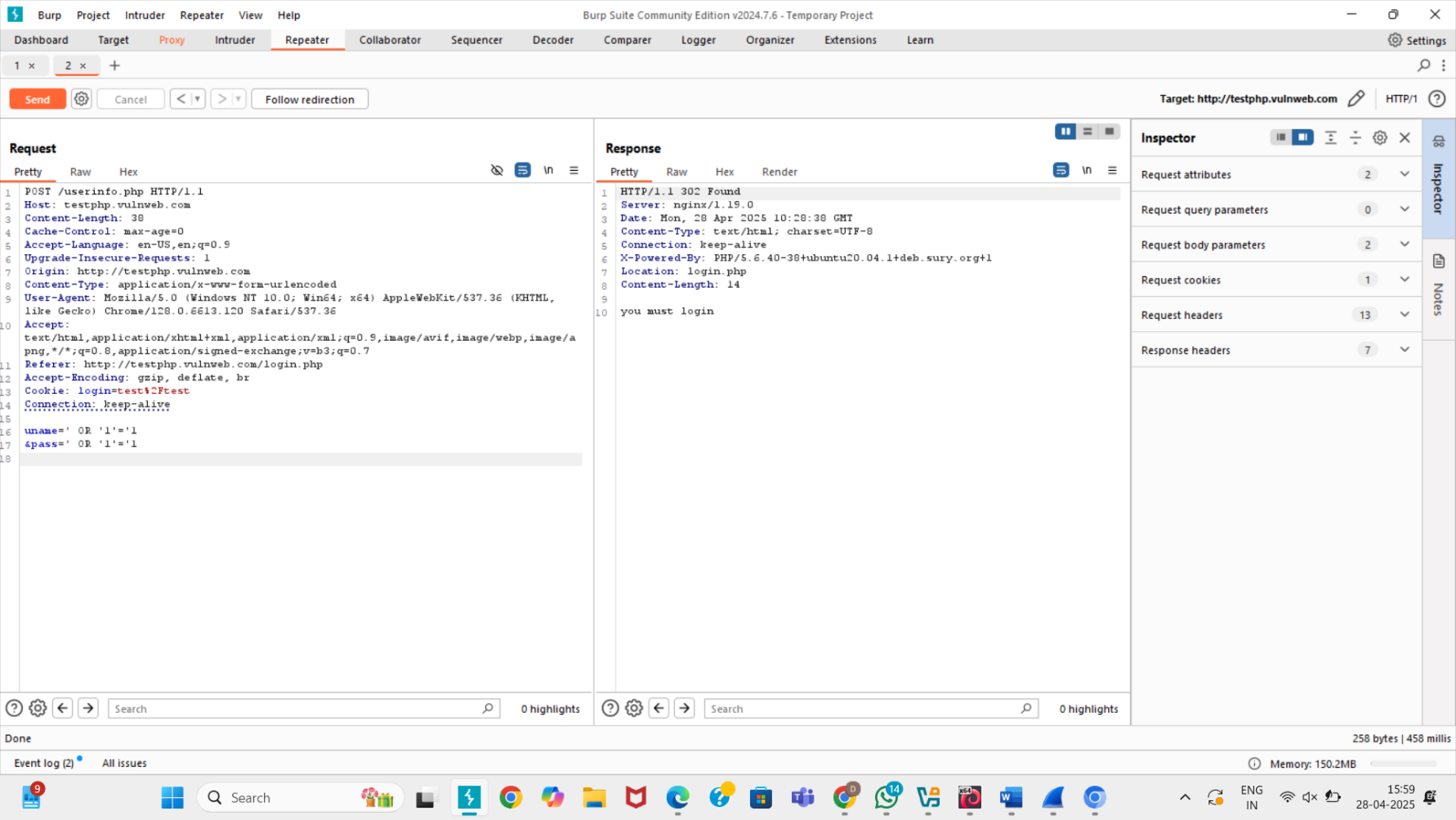
* **Wireshark packet capture session, This was mainly focusing on the HTTP requests to a web application at**[**http://testphp.vulnweb.com/**](http://testphp.vulnweb.com/)
* **Multiple HTTP requests and responses, particularly focusing on POST and GET requests.**

****

****

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

****

****

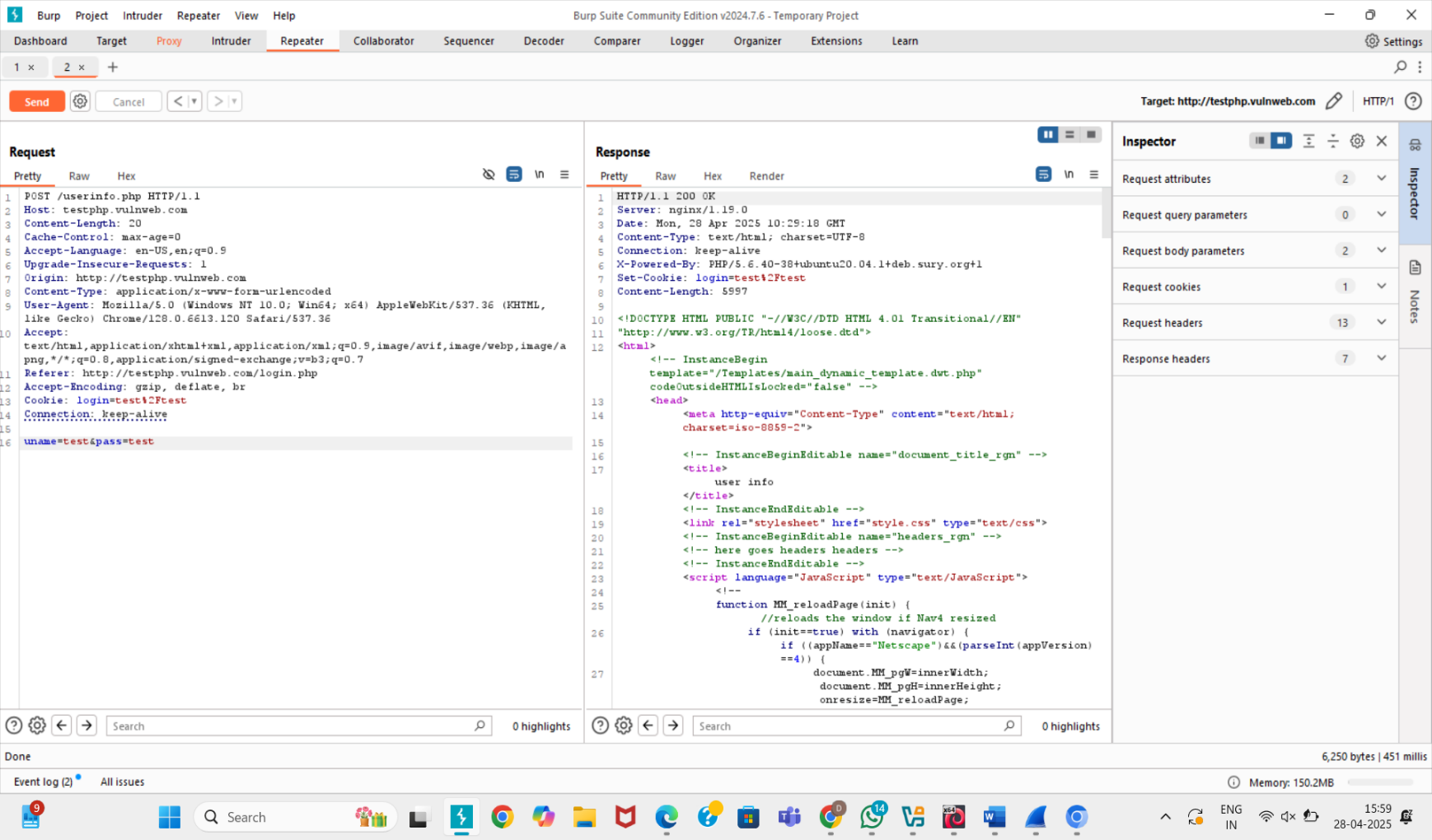
* **Vulnerable to SQL Injection is use to bypass login or get strange results.**
* **In the input field enter a common SQLi payload like :-**

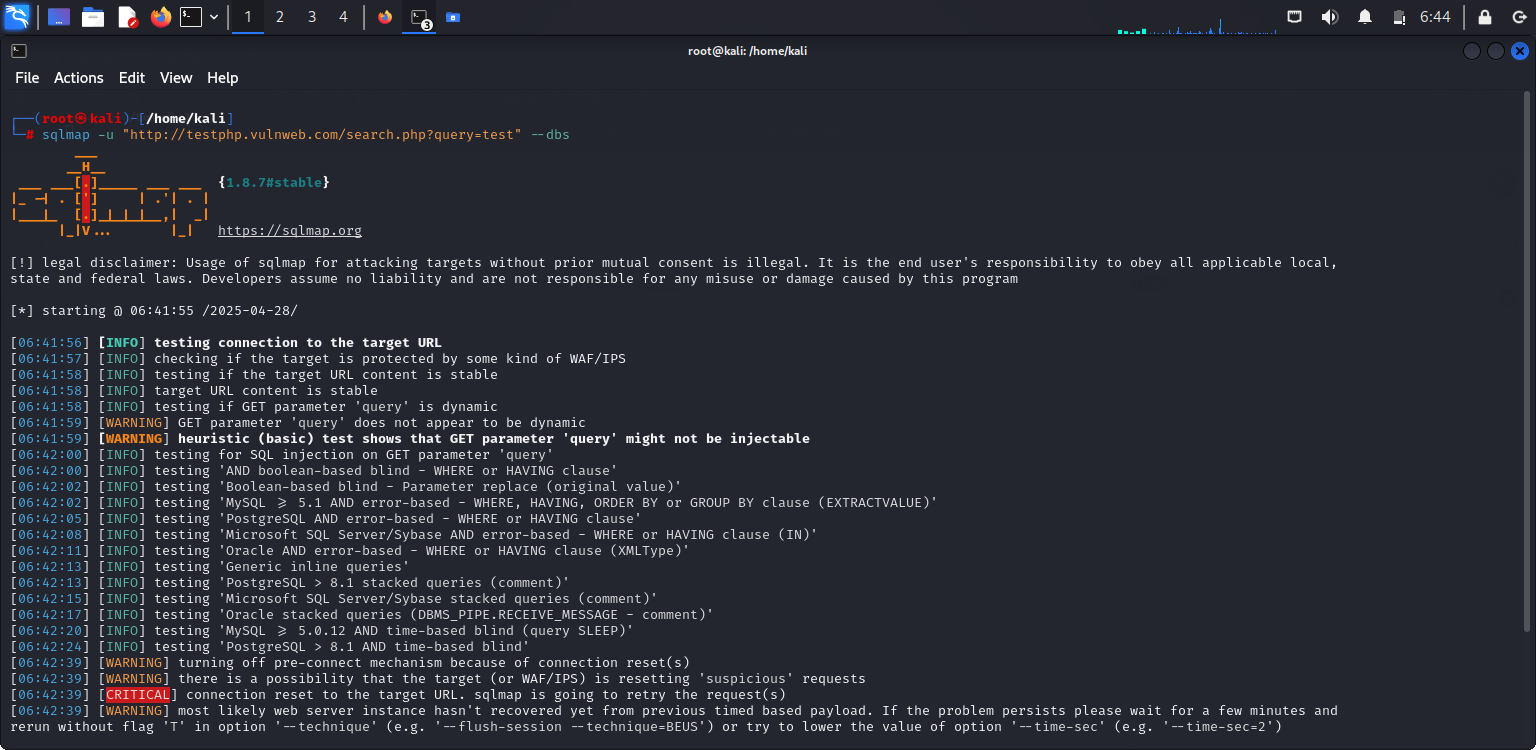
**' OR '1'='1**

* **Use to extract database information automatically.**
* **Command :-**

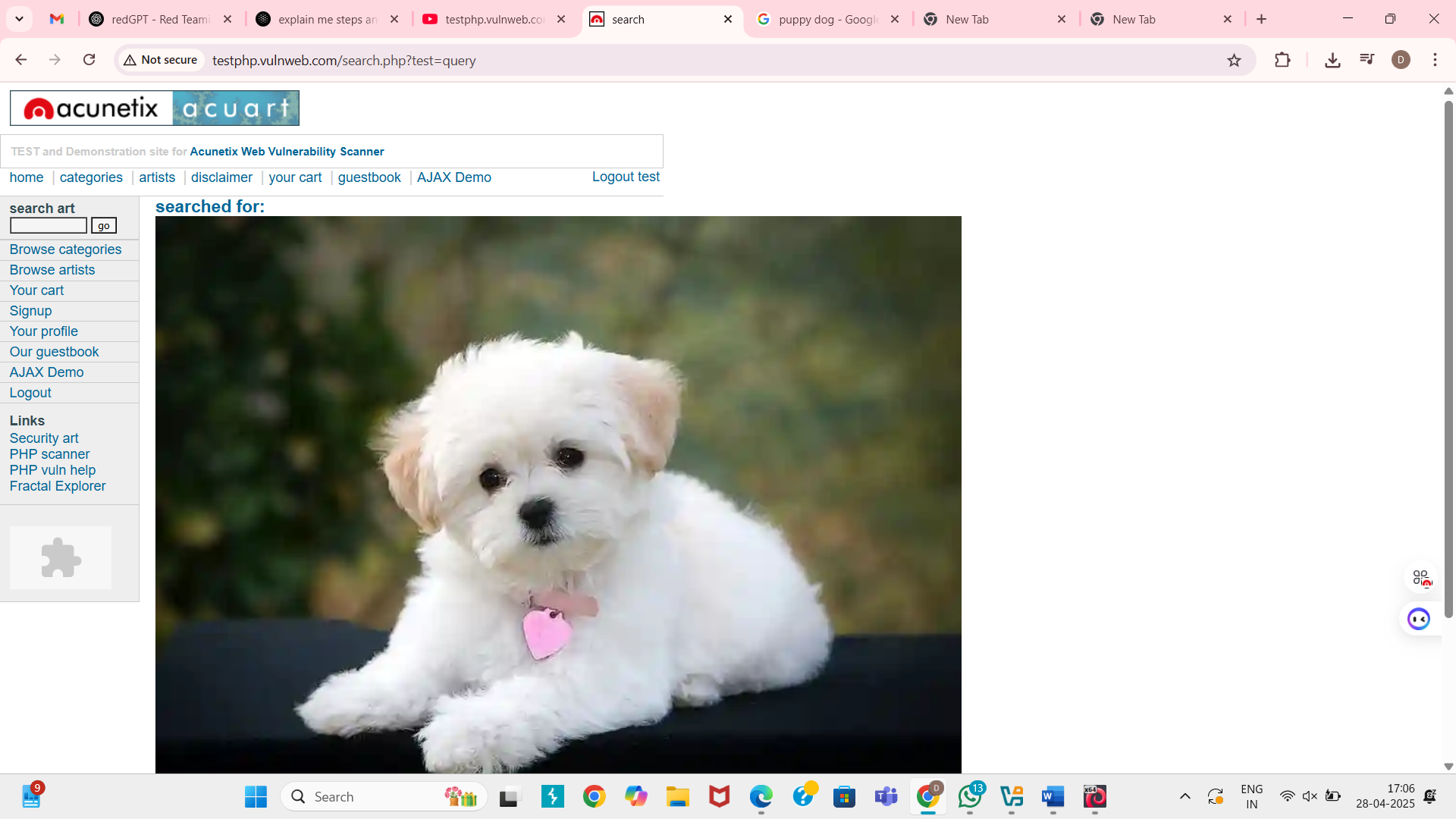
**sqlmap -u "http://testphp.vulnweb.com/search.php?query=test" –dbs**

* **Tells sqlmap to check for SQLi and list available databases.**

****

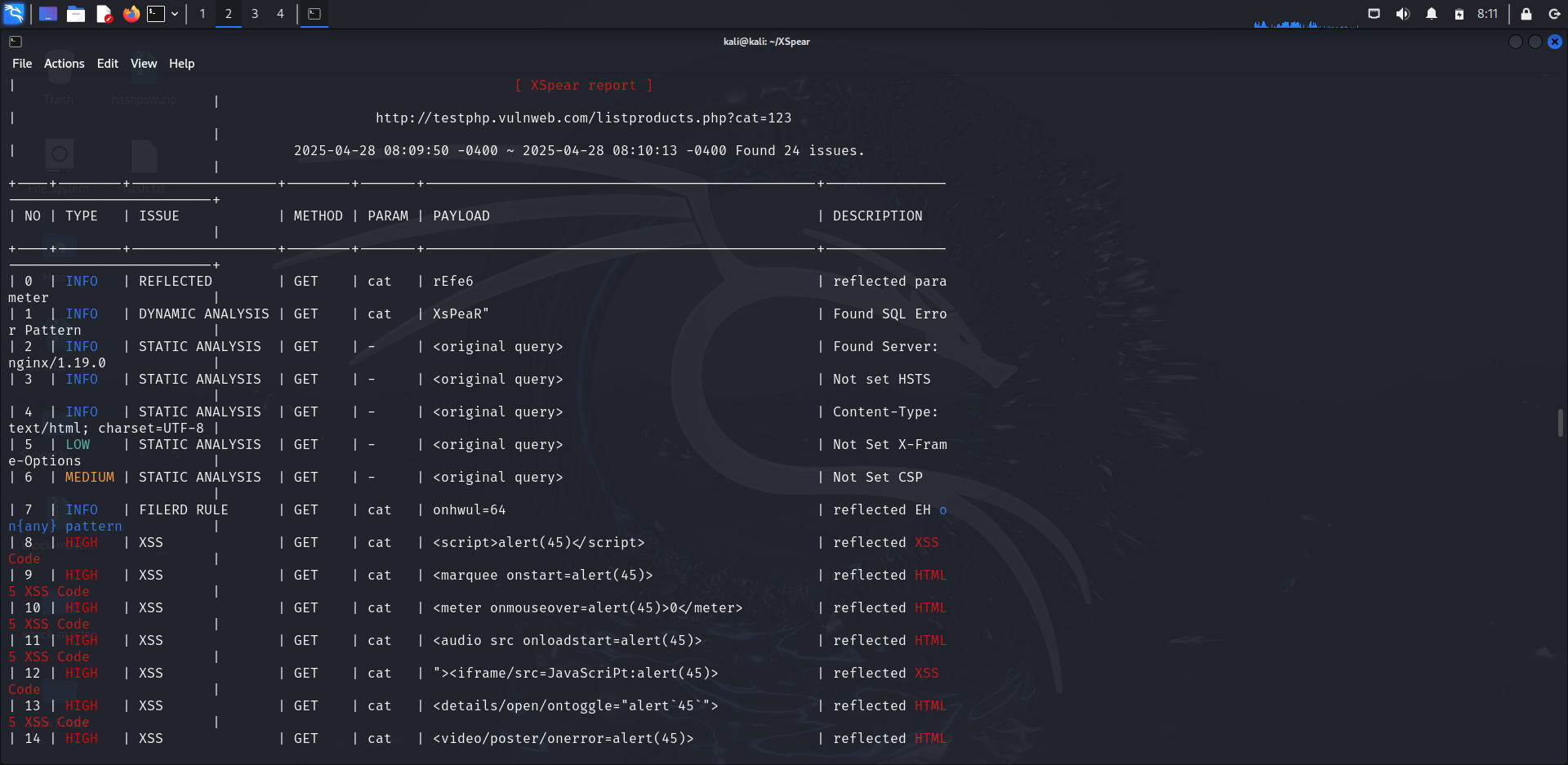
**x**

* **sudo sqlmap -u** [**http://testphp.vulnweb.com/artists.php\?artist\=1**](http://testphp.vulnweb.com/artists.php\?artist\=1)
* **sudo sqlmap -u** [**http://testphp.vulnweb.com/artists.php\?artist\=1**](http://testphp.vulnweb.com/artists.php\?artist\=1) **--dbs -D acuart --tables**
* **sudo sqlmap -u** [**http://testphp.vulnweb.com/artists.php\?artist\=1**](http://testphp.vulnweb.com/artists.php\?artist\=1) **--dbs -D acuart -T carts --columns**
* **sudo sqlmap -u** [**http://testphp.vulnweb.com/artists.php\?artist\=1**](http://testphp.vulnweb.com/artists.php\?artist\=1) **--dbs -D acuart -T carts -C item --dump-all**

****

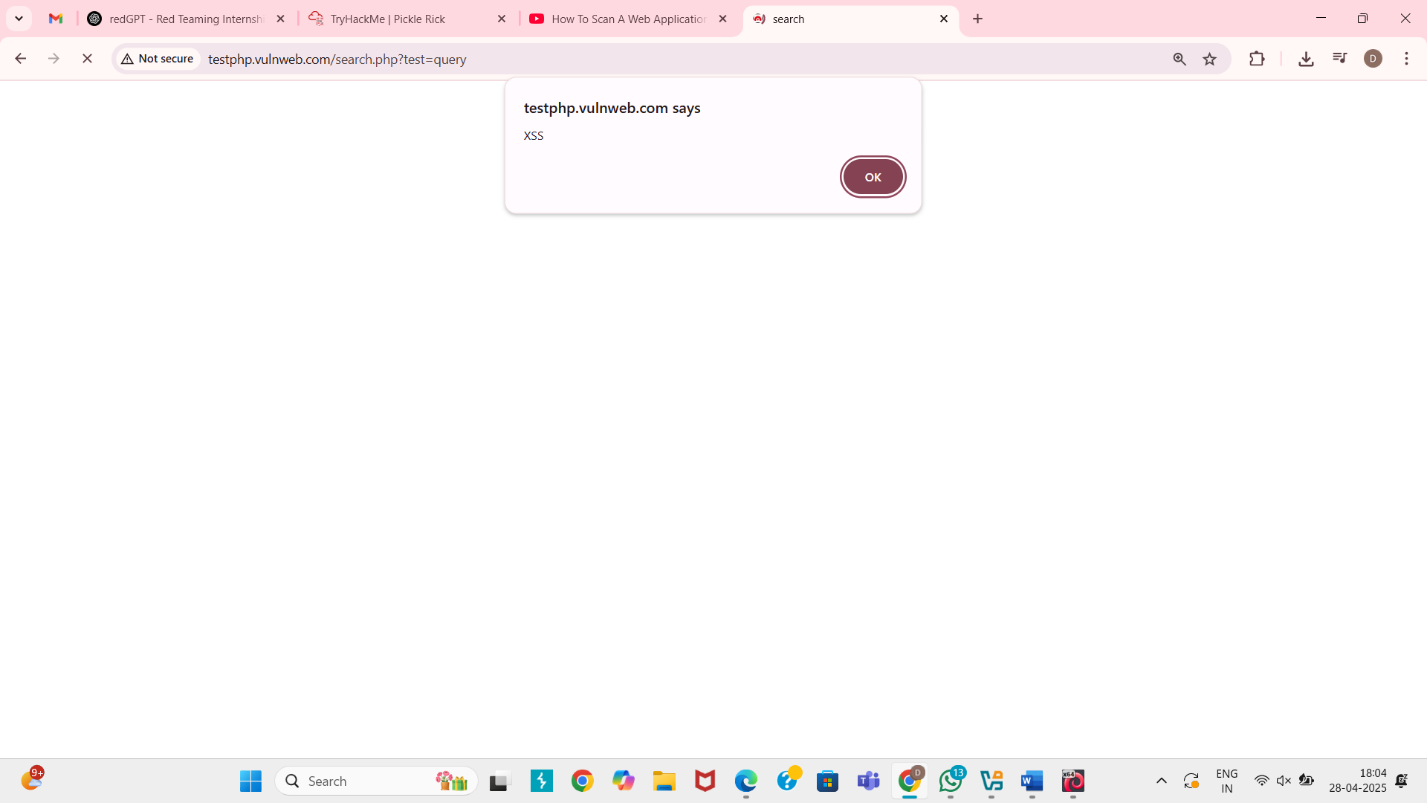
* **<imgsrc=**[**https://rukminim2.flixcart.com/image/850/1000/l071d3k0/poster/q/u/k/medium-cute-dogs-cute-puppies-pomeranian-fulffy-dog-golden-original-imagcf8wgyxqkeuf.jpeg?q=20&crop=false**](https://rukminim2.flixcart.com/image/850/1000/l071d3k0/poster/q/u/k/medium-cute-dogs-cute-puppies-pomeranian-fulffy-dog-golden-original-imagcf8wgyxqkeuf.jpeg?q=20&crop=false) **/>**

**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.**

****

****

* **Steps for the Scan A Web Application for XSS Vulnerability.**
* **1 git clone** [**https://github.com/hahwul/XSpear.git**](https://github.com/hahwul/XSpear.git)
* **cd XSpear/**
* **ls**
* **gem**
* **sudo gem install XSpear-1.4.1.gem**
* **XSpear -u** [**http://testphp.vulnweb.com/listproducts.php?cat=123**](http://testphp.vulnweb.com/listproducts.php?cat=123) **-v 1**

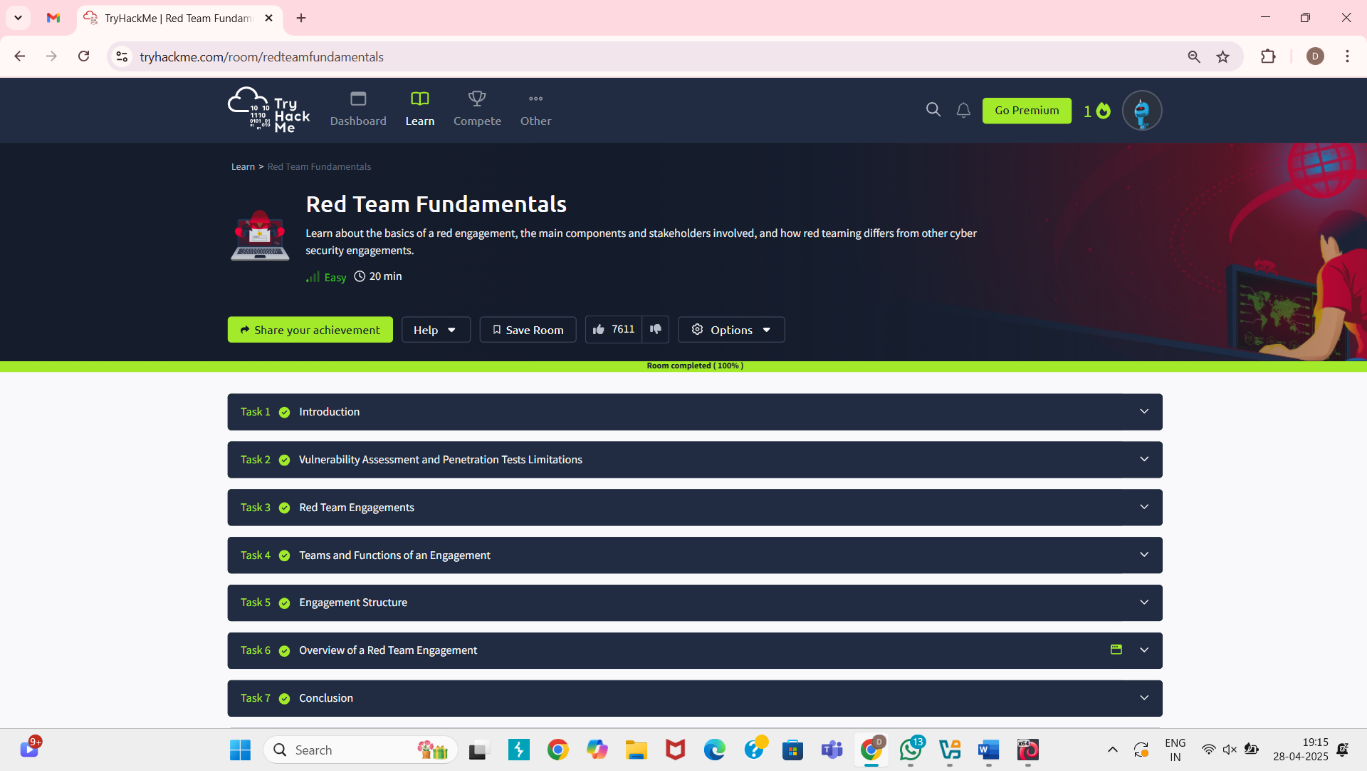
****

* **<script>alert('XSS')</script>**
* **If a small popup window appears with a message like "XSS" → The website is vulnerable to XSS.**

**B) TASK (CTF)**

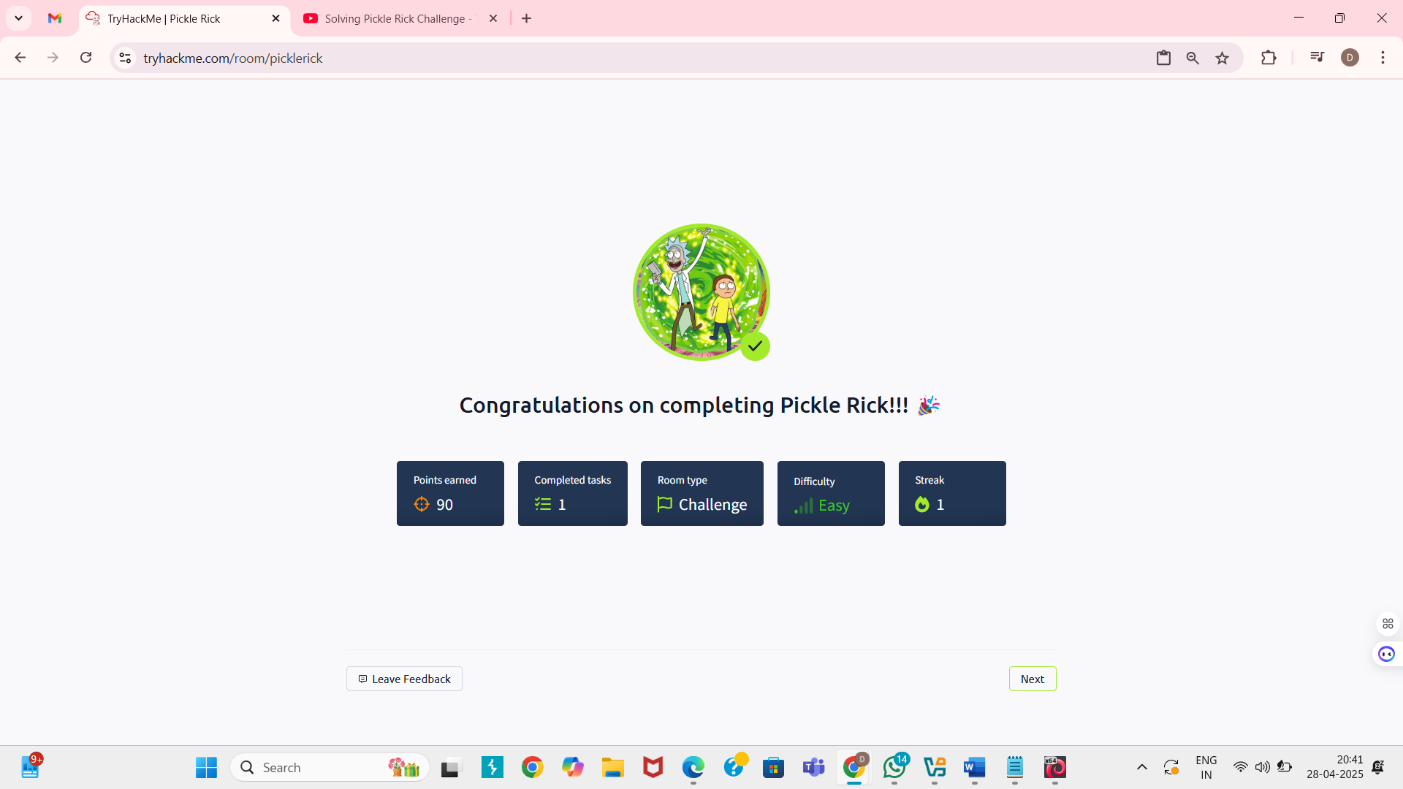
**1. Red Team Fundamentals**

* **Tools :- Mimikatz, PowerShell, Meterpreter.**

****

**2. PickleRick**

* **Tools :- PowerShell Empire, Covenant C2, Metasploit, BloodHound, Mimikatz.**

****

**C) ETHICAL HACKING PROJECT**

* **Title :- Password Strength Checker**

1. **Working Process and Key Functions :**

* **User Input :- The program asks the user to enter a password.**
* **Character Analysis :- It checks whether the password contains:**
* **Uppercase letters**
* **Lowercase letters**
* **Numbers**
* **Special characters**
* **Length Validation :- It ensures the password is at least 8 characters long.**
* **Result Output :- Based on these checks, the password is categorized as:**
* **Strong Password 🔥 (meets all conditions)**
* **Moderate Password ⚡ (meets some conditions)**
* **Weak Password ❌ (fails important checks)**

1. **Learning Outcomes :**

* **Technical Proficiency :- I gained practical coding skills in Python and learned how to implement basic cybersecurity concepts like password strength validation.**
* **Understanding of Cybersecurity Lifecycle :- I understood how proper password hygiene plays a critical role in securing user accounts and data.**
* **Problem-Solving :- Improved my logic skills by designing rules to classify passwords as Weak, Moderate, or Strong.**
* **Real-World Application :- Learned how basic tools like password checkers can protect users from common cyber threats.**
* **Project Documentation :- Gained experience in properly writing, documenting, and submitting a technical project.**

1. **Challenges and Solutions :**

* **Challenge 1 :- Program Closes Immediately After Running**

**Solution: Added input("Press Enter to exit...") at the end to keep the window open.**

* **Challenge 2: Confused About Password Checking Conditions**

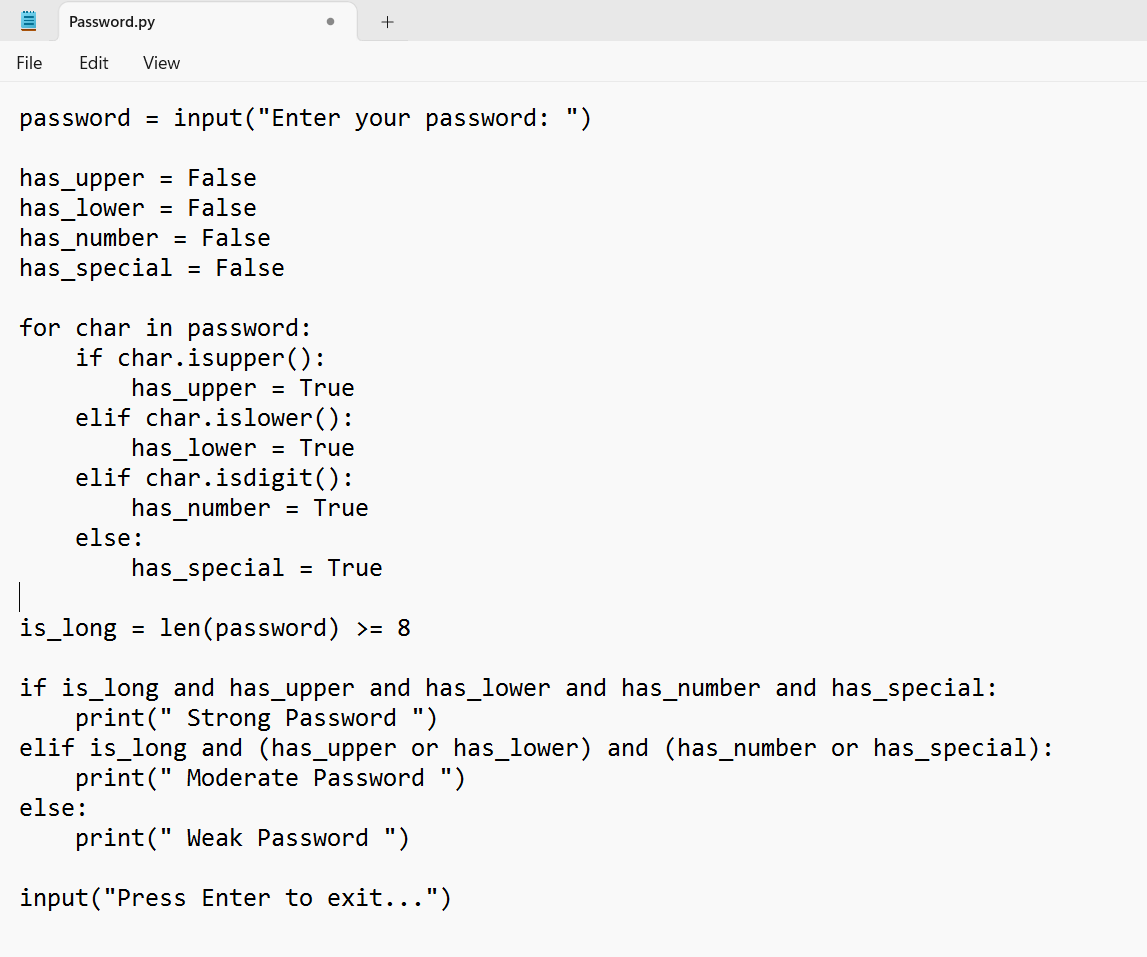
**Solution: Broke down the problem into small parts (check uppercase, lowercase, numbers, special characters separately).**

* **Challenge 3: GitHub Upload Confusion**

**Solution: Followed simple GitHub upload steps (create repository → upload file → copy link).**

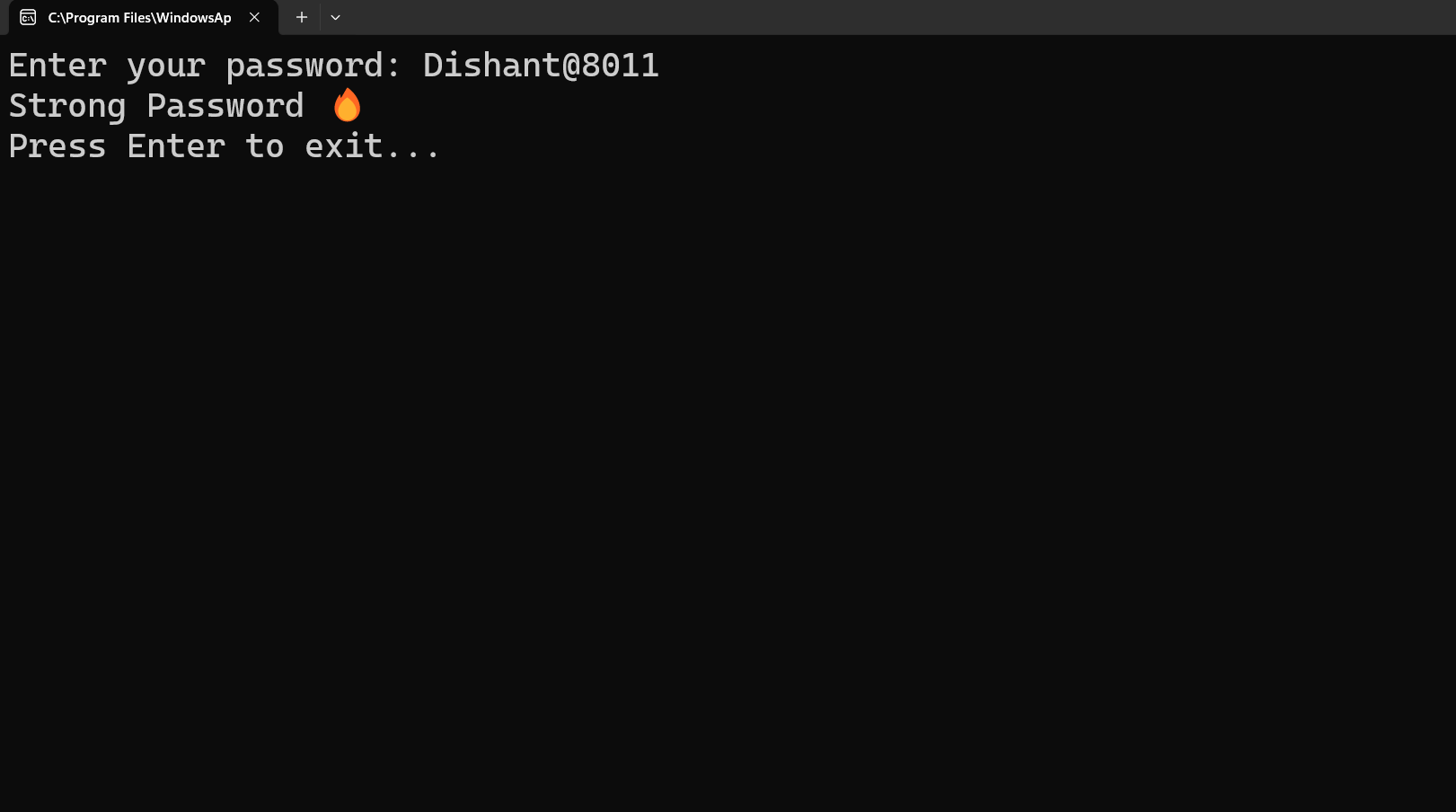
* **Challenge 4: Multiple Password Inputs**

**Solution: Created a function (check\_password\_strength) to easily reuse the checking code for multiple passwords.**

****

1. **GitHub Link :**

* **I uploaded the project to GitHub. Here is the repository link:**[**GitHub Repository - Password Strength Checker**](https://github.com/Dishantsonani/Password_strength_checker)

****

1. **Conclusion :**

* **My internship at Hack Secure helped me understand how small projects like this play a big role in real-world cybersecurity.**
* **Learning to identify and classify password weaknesses gave me insights into protecting user credentials.**
* **This project boosted my technical and problem-solving abilities, preparing me for advanced cybersecurity challenges ahead.**

1. **Acknowledgments :**

* **I sincerely thank Hack Secure, especially my mentor Mr. Nishant Prajapati and assistant mentor Mr. Aman Pandey, for their constant support and guidance.**
* **I also extend my gratitude to Hack Secure Hiring Team for offering this valuable internship opportunity, which greatly contributed to my personal and professional growth.**