Byte Capsule

STAY SAFE, STAY SECURE

Web Application Security- EHP

Date: 31.1.2025

Name: Sudipta Mondal

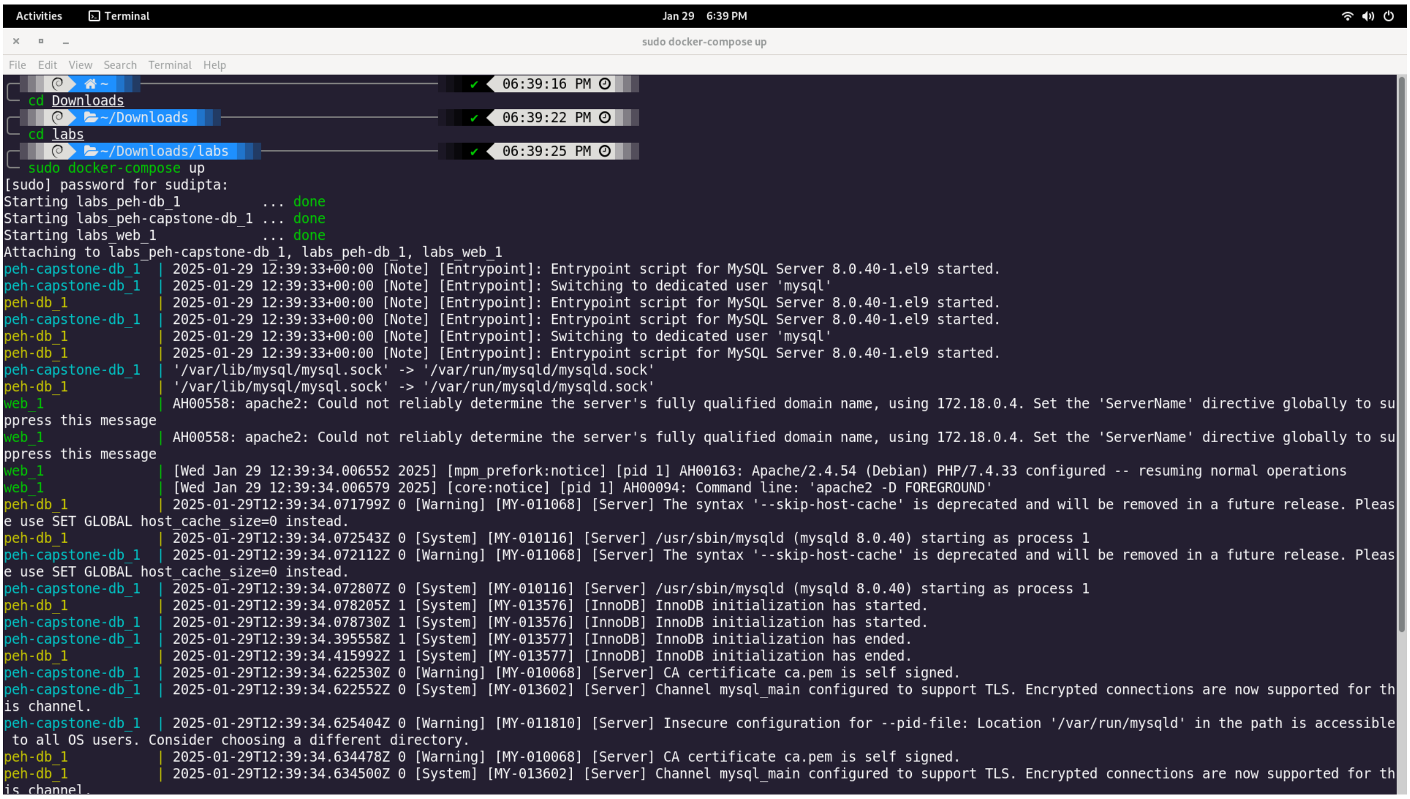
Email: sudiptamondol22@gmail.com

Contact Number: 01834644860

Address: Faridpur Sadar, Dhaka

Environment

Details of lab: The lab for the pentesting a website is given by the Bytecapsuleit authority .The lab is set up in my Parrot OS machine. Using docker ,I have done all the work. Here is the procedure of opening the lab-



```
~                                                        ✔ 06:39:16 PM
cd Downloads
~/Downloads                                              ✔ 06:39:22 PM
cd labs
~/Downloads/labs                                         ✔ 06:39:25 PM
sudo docker-compose up
[sudo] password for sudipta:
Starting labs_peh-db_1          ... done
Starting labs_peh-capstone-db_1 ... done
Starting labs_web_1             ... done
Attaching to labs_peh-capstone-db_1, labs_peh-db_1, labs_web_1
peh-capstone-db_1  | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.40-1.el9 started.
peh-capstone-db_1  | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
peh-db_1           | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.40-1.el9 started.
peh-capstone-db_1  | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.40-1.el9 started.
peh-db_1           | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
peh-db_1           | 2025-01-29 12:39:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.40-1.el9 started.
peh-capstone-db_1  | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
peh-db_1           | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
web_1              | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.4. Set the 'ServerName' directive globally to su
ppress this message
web_1              | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.4. Set the 'ServerName' directive globally to su
ppress this message
web_1              | [Wed Jan 29 12:39:34.006552 2025] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.54 (Debian) PHP/7.4.33 configured -- resuming normal operations
web_1              | [Wed Jan 29 12:39:34.006579 2025] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
peh-db_1           | 2025-01-29T12:39:34.071799Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Pleas
e use SET GLOBAL host_cache_size=0 instead.
peh-db_1           | 2025-01-29T12:39:34.072543Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.40) starting as process 1
peh-capstone-db_1  | 2025-01-29T12:39:34.072112Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Pleas
e use SET GLOBAL host_cache_size=0 instead.
peh-capstone-db_1  | 2025-01-29T12:39:34.072807Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.40) starting as process 1
peh-db_1           | 2025-01-29T12:39:34.078205Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
peh-capstone-db_1  | 2025-01-29T12:39:34.078730Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
peh-capstone-db_1  | 2025-01-29T12:39:34.395558Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
peh-db_1           | 2025-01-29T12:39:34.415992Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
peh-capstone-db_1  | 2025-01-29T12:39:34.622530Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
peh-capstone-db_1  | 2025-01-29T12:39:34.622552Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for th
is channel.
peh-capstone-db_1  | 2025-01-29T12:39:34.625404Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible
 to all OS users. Consider choosing a different directory.
peh-db_1           | 2025-01-29T12:39:34.634478Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
peh-db_1           | 2025-01-29T12:39:34.634500Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for th
is channel.
```

Fig1:Lab information

Target Website:

http://localhost/

http://localhost/capstone/index.php

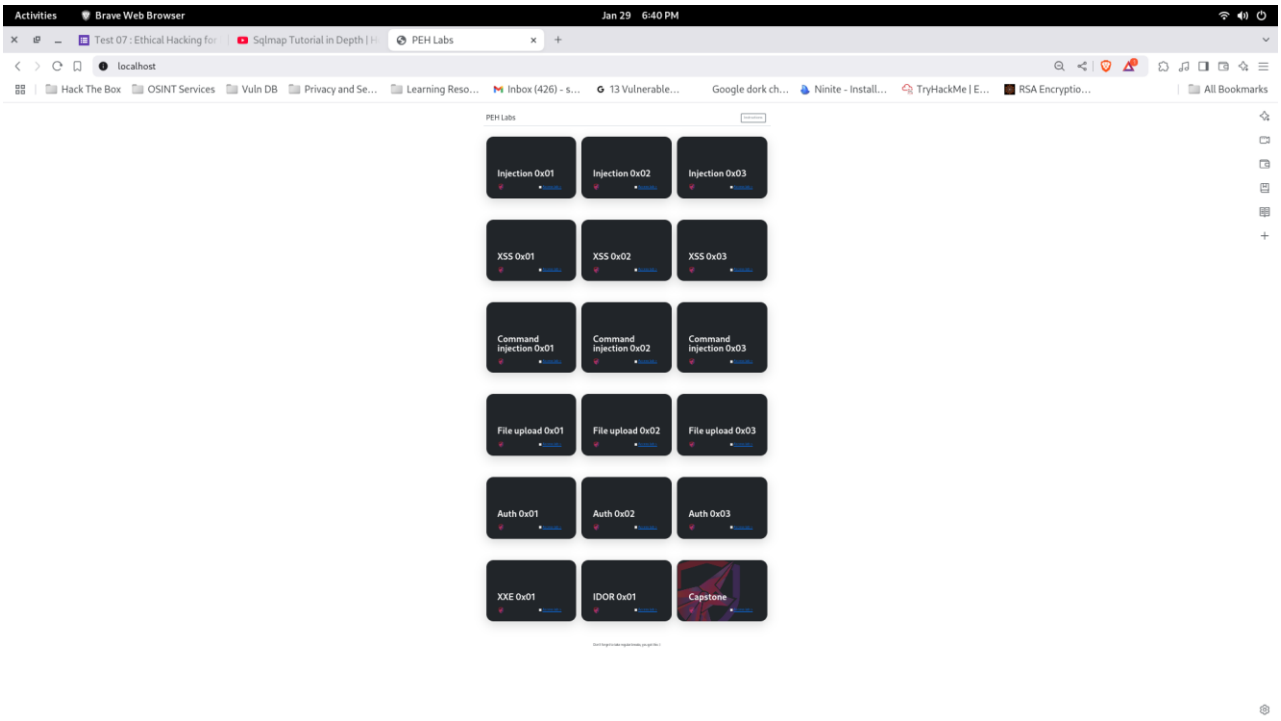


Fig 2: localhost/capstone(target website)

Login:

Login process: After creating a user account , we can log in to the website following this process-

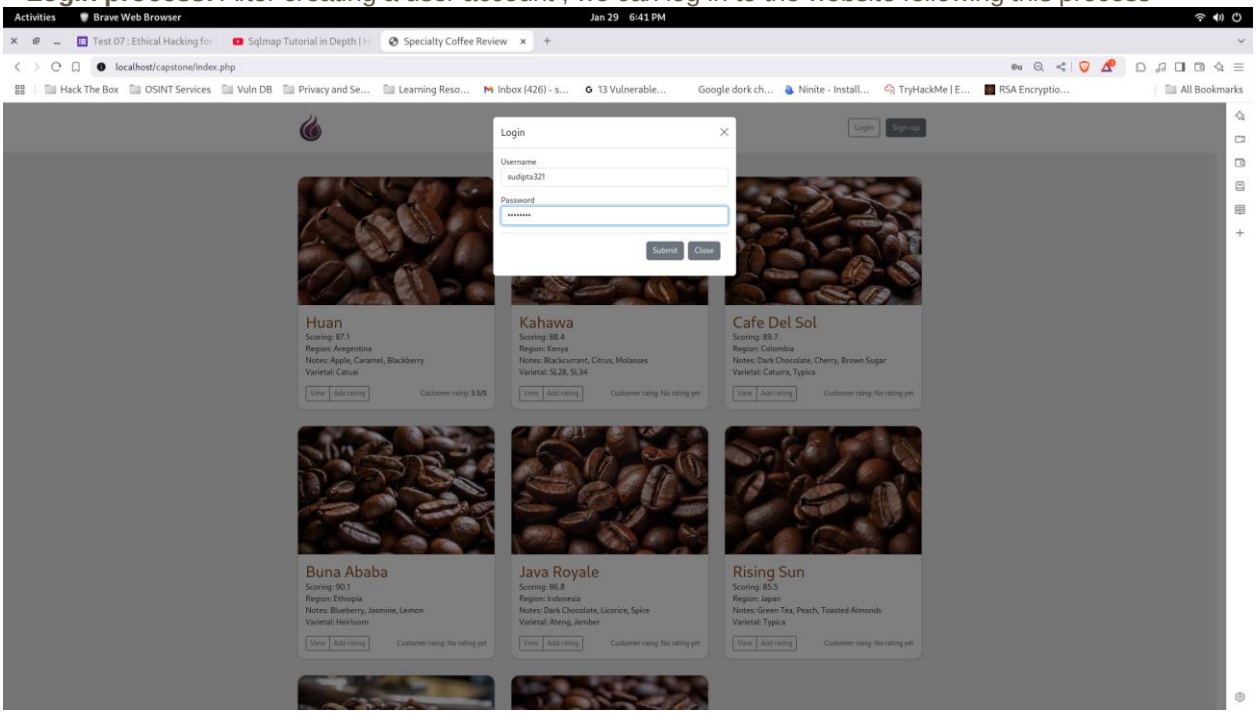


Fig 3: Creating account

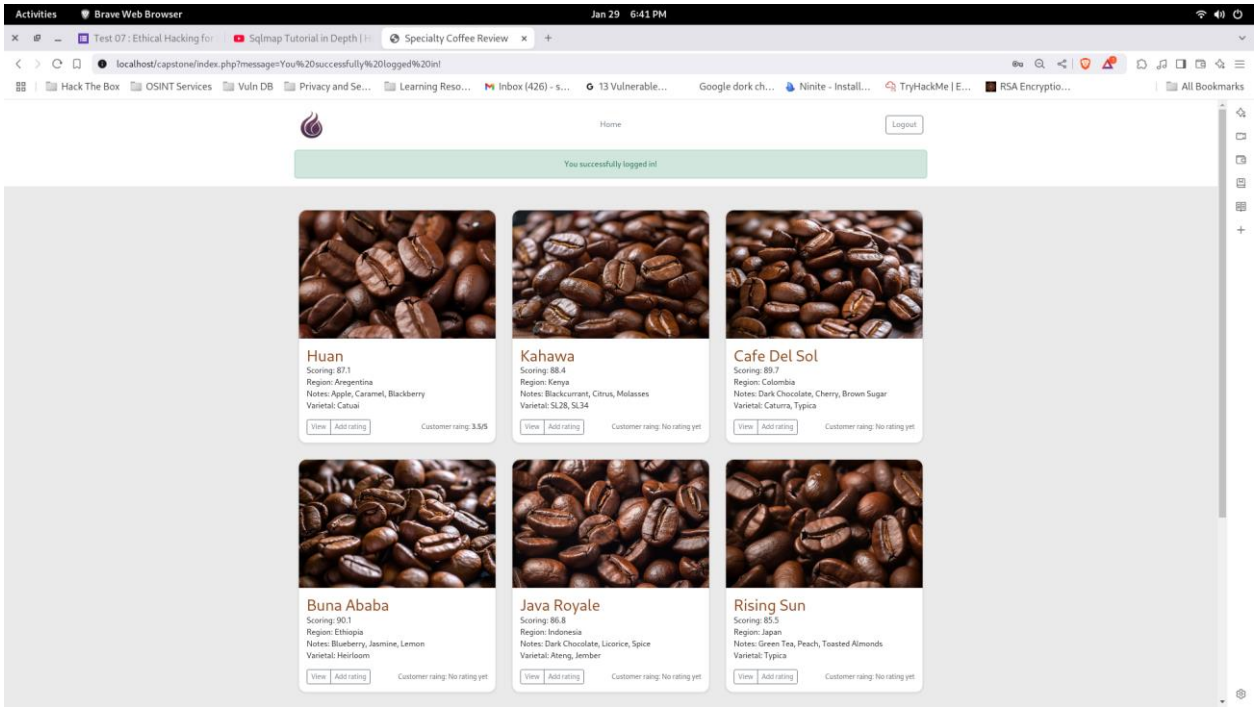


Fig 4: Registered and logged in successfully

Critical findings using SQL injection

Steps of finding bugs:

1.First I have given command in sqlmap to find the vulnerable part of the target website

2. Then from there I have taken out the names of Database, Tables, usernames and passwords

3.After finding hashed passwords I have cracked them with hashcat

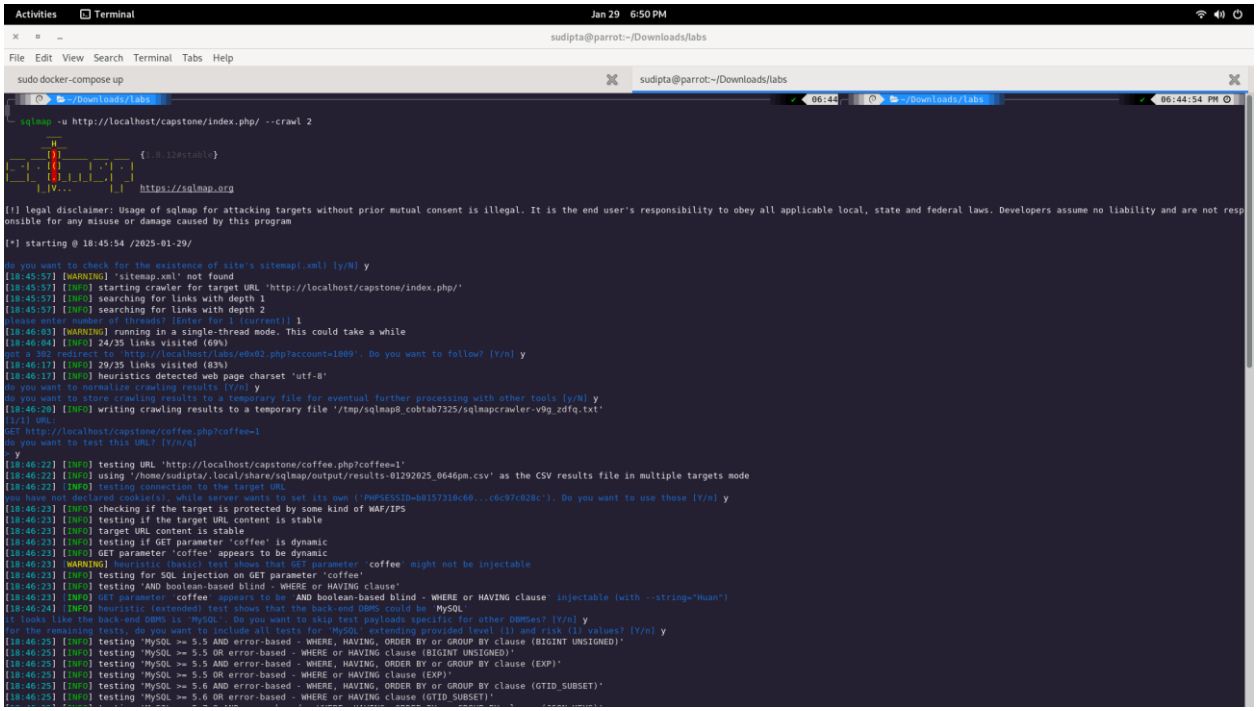


Fig-5: Sqlmap

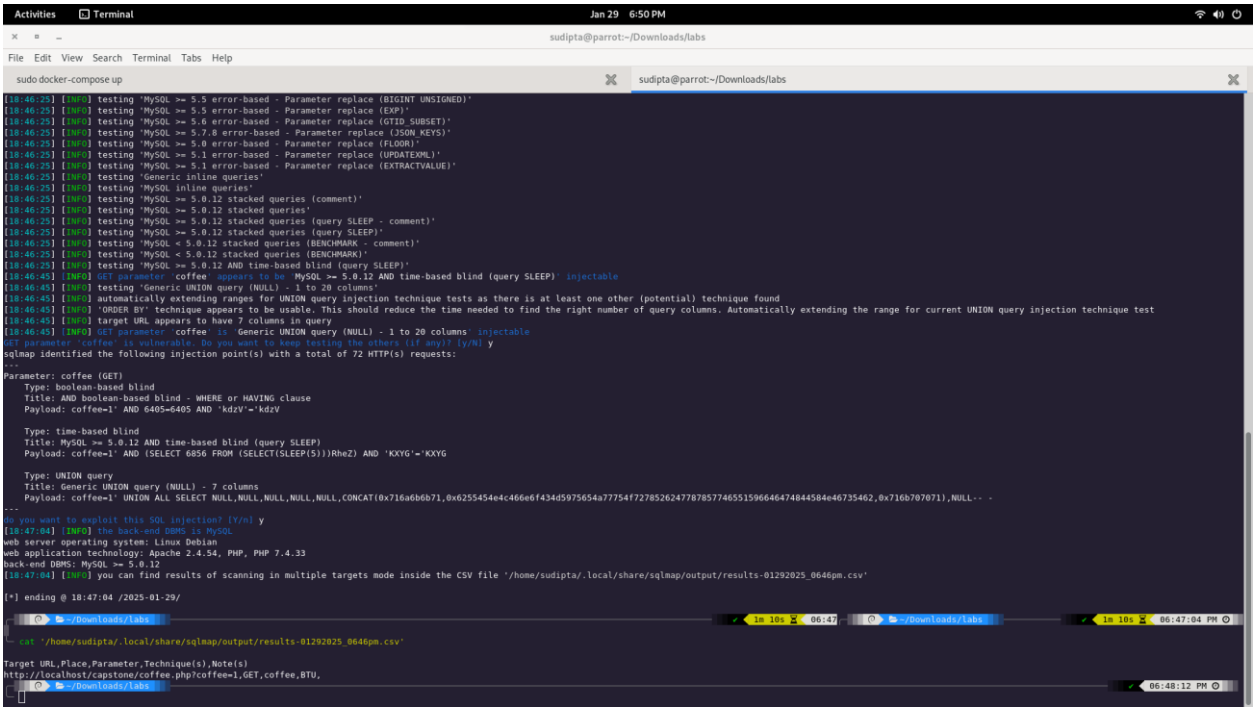


fig-6: Sqlmap(vulnerable part) using cat

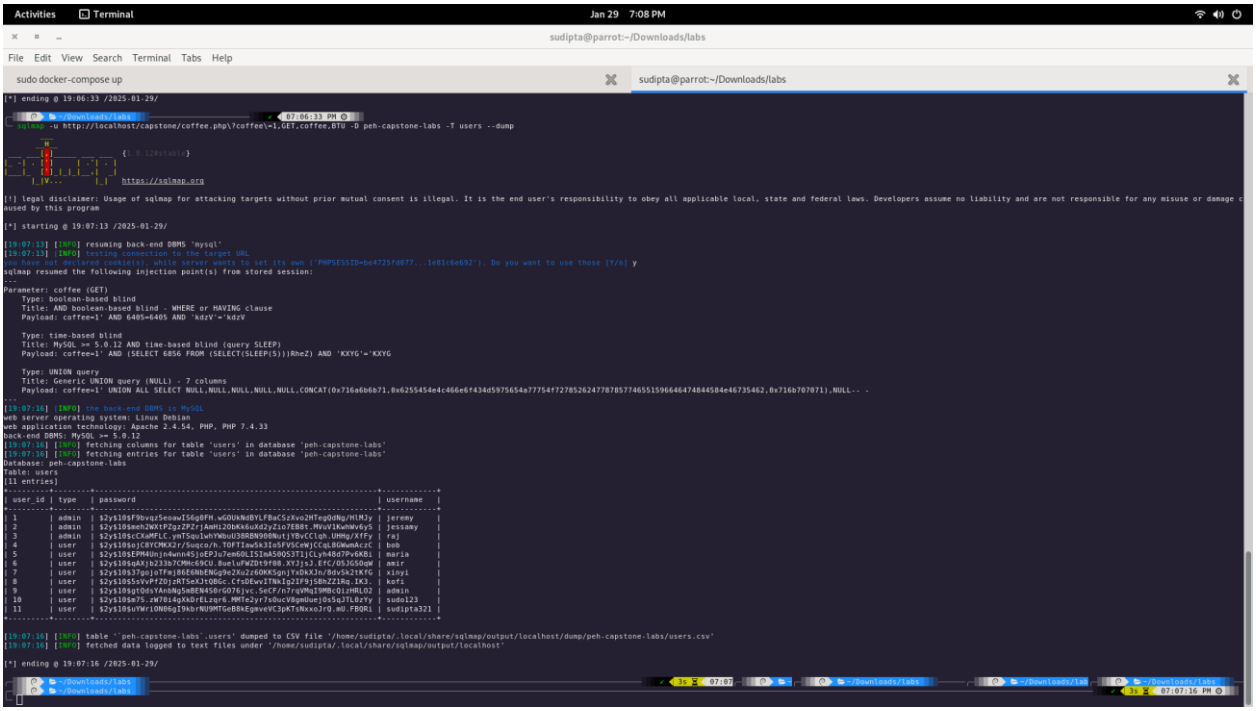


fig-7: usernames and hashed passwords using sqlmap

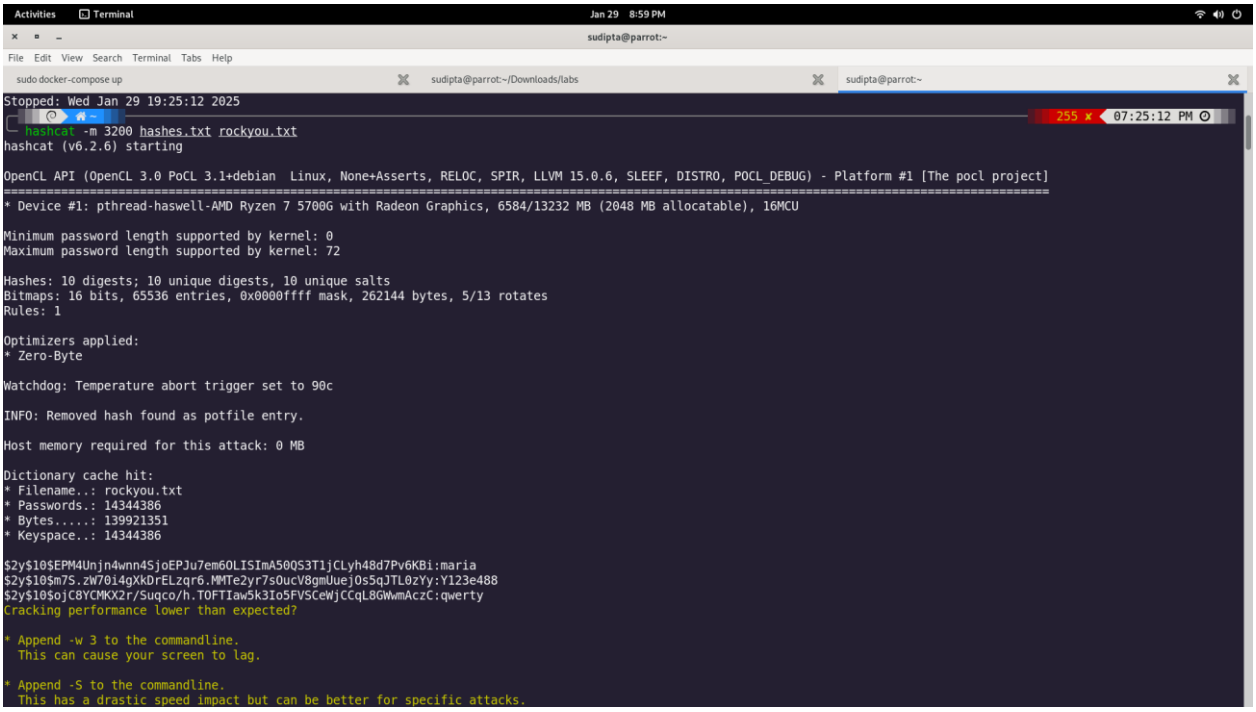


fig-8: Craking hased passwords using hashcat

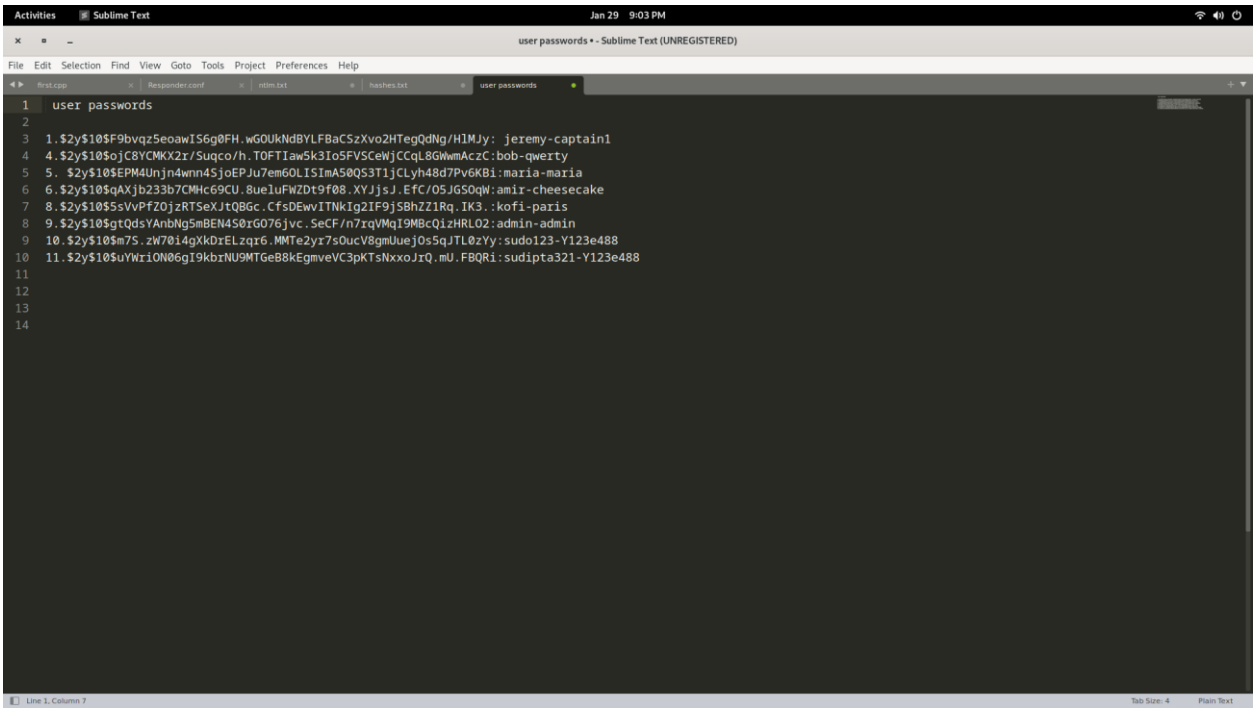


fig-9: Cracked admin and user credentials

Expected Results of a Successful SQL Injection

- **Authentication Bypass**
    - Login without a password
    - Access restricted areas
- **Data Extraction**
    - Retrieve usernames, passwords, emails
    - Use UNION-based injection
- **Error-Based SQL Injection**
    - Expose table names, column count, database errors
- **Blind SQL Injection**
    - Detect injection via True/False conditions
- **Database Dumping**
    - Extract entire database using SQLMap

Actual Result:

1.I have found user and admin credentials like username and passwords

2. I have found database name

3. I have found table names, column and also entire database

4. can now login as an admin

Reflected & Stored XSS injection using XSS Payload:

Steps to finding bugs: OWASP-ZAP is a automated tools . After opening the tools ,I perform active scan using the in-scope url and found some bugs .In short total 4 HIGH risk ,4 MEDIUM risk ,5 LOW risk bugs are found .The result of finding are given below-

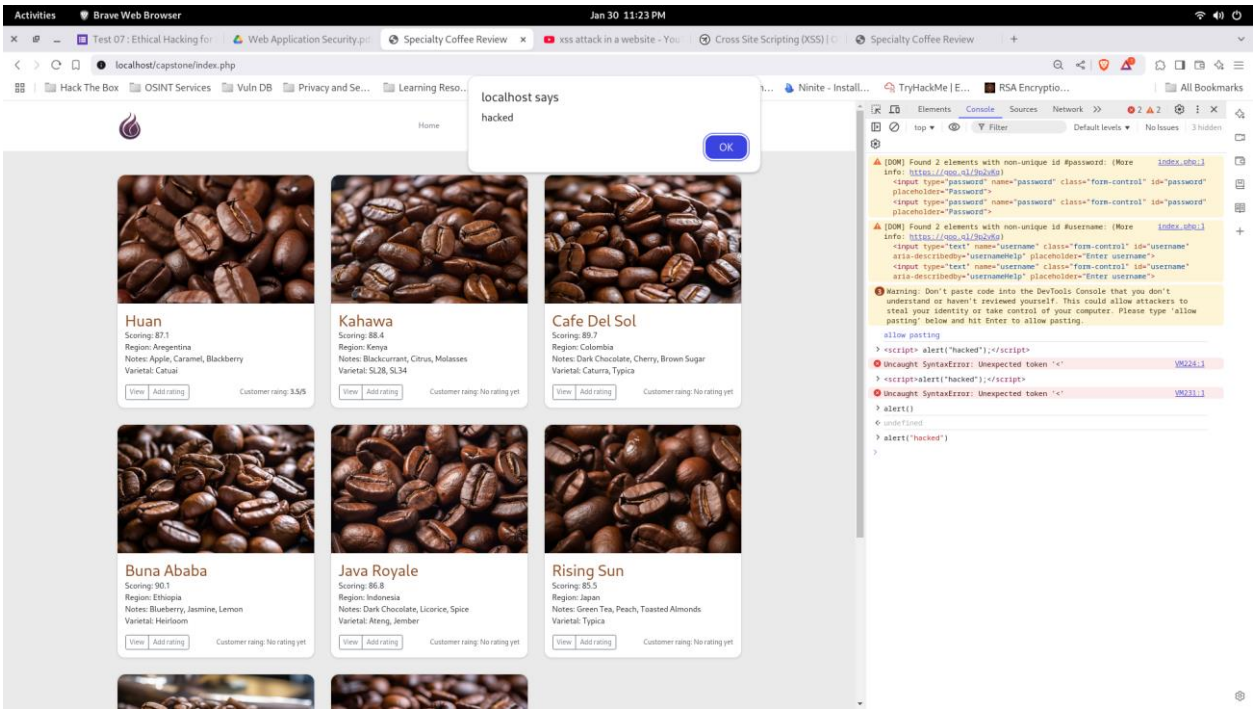


fig-10: XSS Vulnerability

Critical findings using Owasp-Zap:

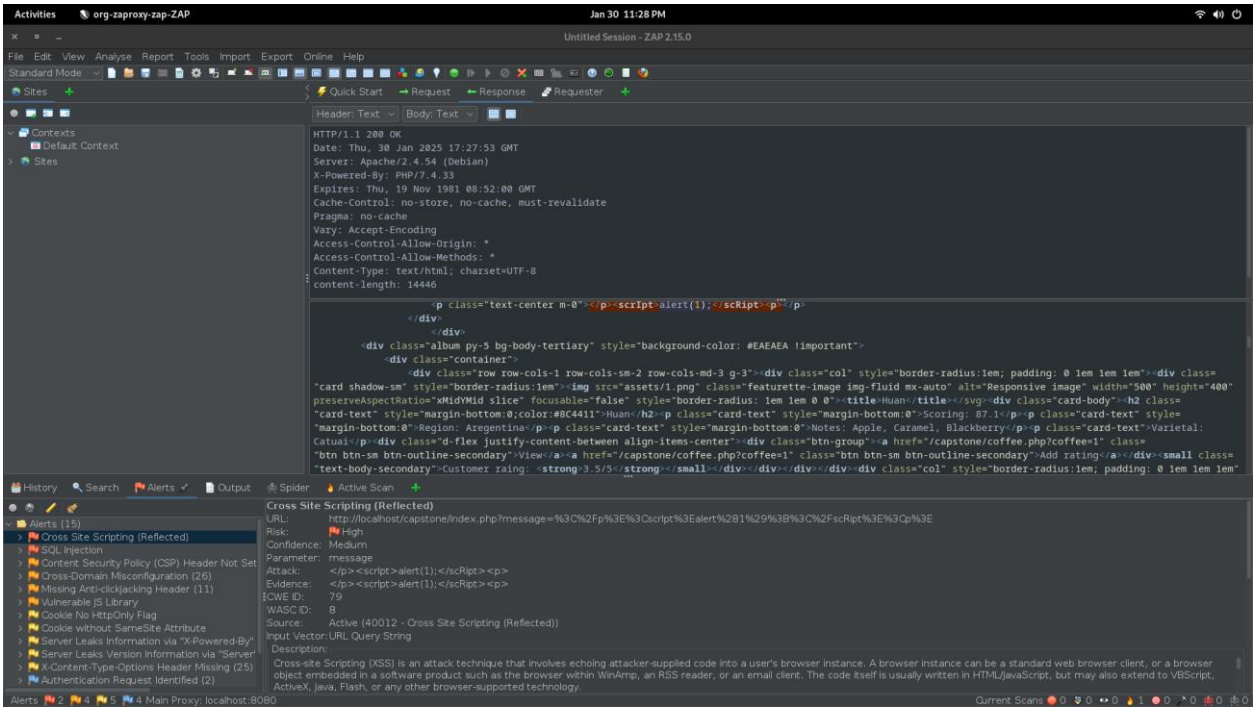


fig-11: Owasp-zap findings 1

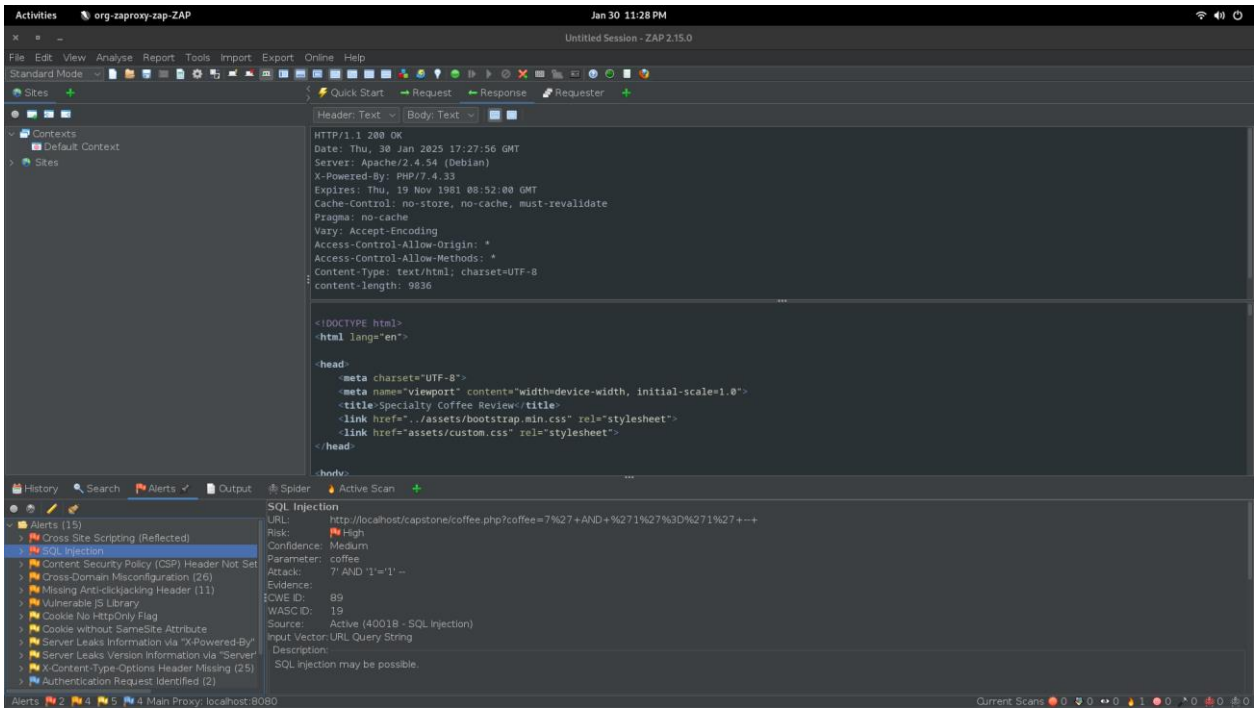


fig-12:Owasp-zap findings 2

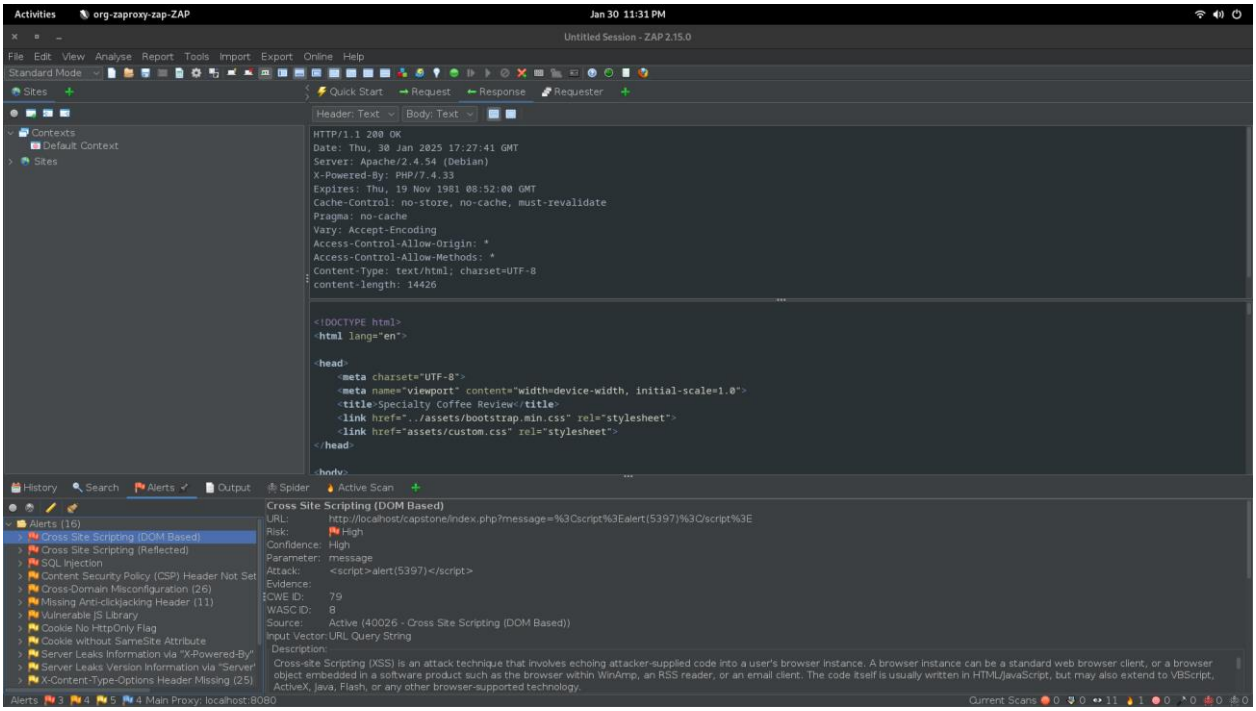


fig-13: Owasp-zap findings 3

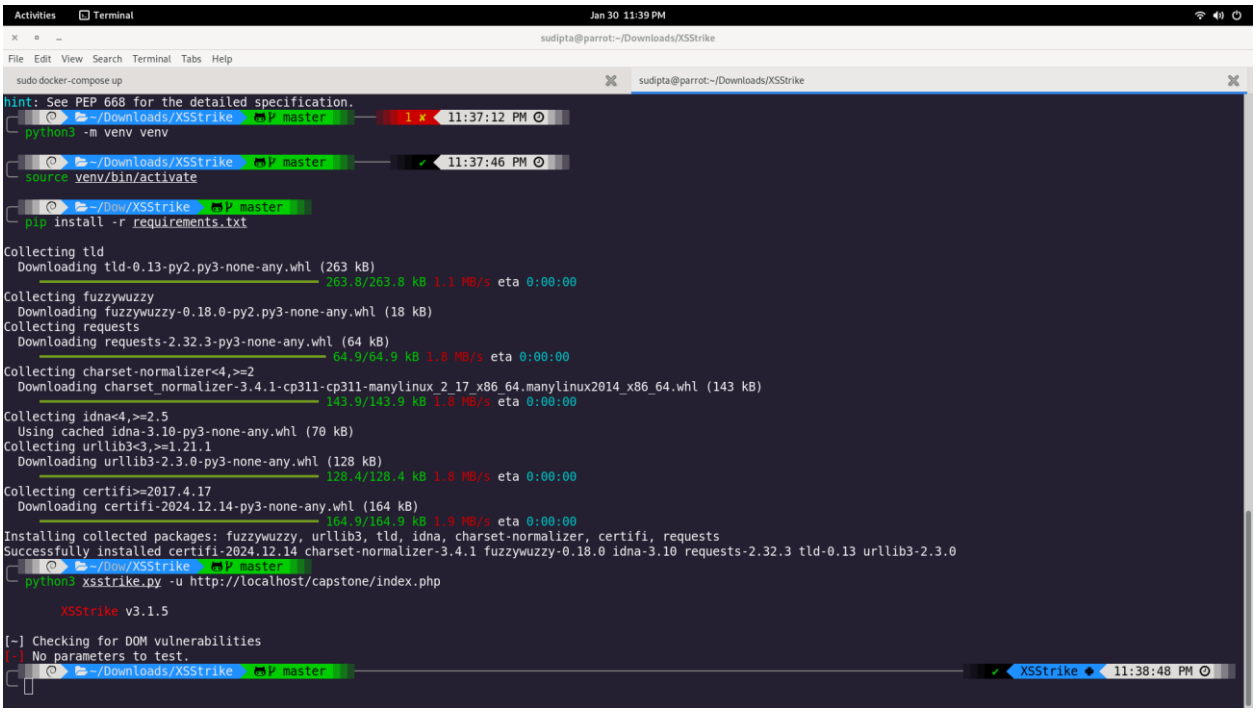


fig-14: Dom vulnerability checking using Xsstrike

Expected Results of XSS Testing Using OWASP ZAP

- **Reflected XSS**
  - Injected script executes immediately in the response.
  - Example: `<script>alert(1)</script>` pops an alert.
- **Stored XSS**
  - Payload is permanently stored (e.g., in comments, profiles).
  - Script executes when other users visit the affected page.
- **DOM-Based XSS**
  - JavaScript modifies the page dynamically to execute the payload.
- **Security Alerts in ZAP**
  - ZAP detects and reports potential XSS vulnerabilities.
  - Shows request/response details for manual verification.

Actual Result:

1.Found 4 high risk bugs

2. Found 4 medium risk bugs

3. Found 5 low risk bugs

4. Found reflected xss bugs

5. found DOM based bugs

Modify POST request and Upload Shell using Burpsuite

Steps of finding bug:

1. First I have found the upload features in localhost/capstone/admin/admin.php

2. then I tried to upload PNG file in the upload section

3. The website is giving error that's why I cannot go further procedure.

4. if the file uploaded successfully then I will go through fuzzing and put some php script on the birpsuite repeater and also get the vulnerable part like www-data, hostname etc etc

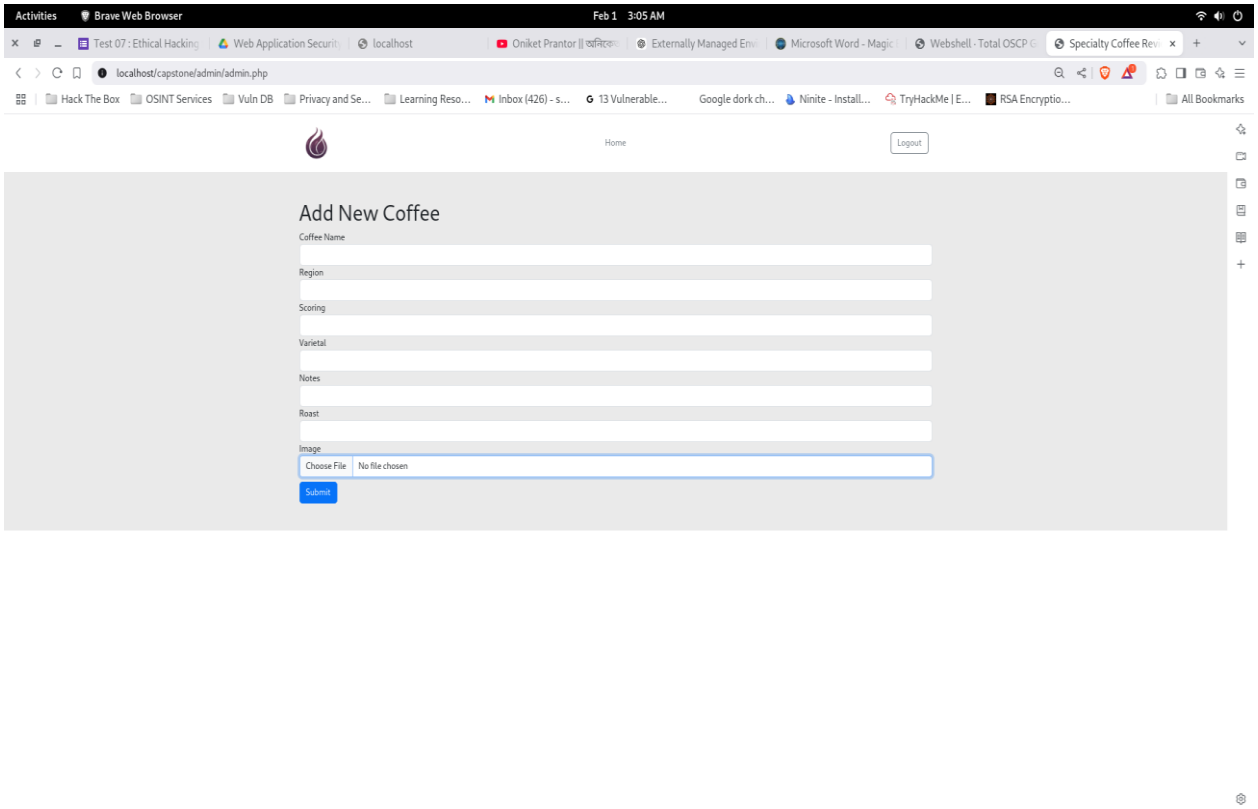


fig-15: Admin upload site

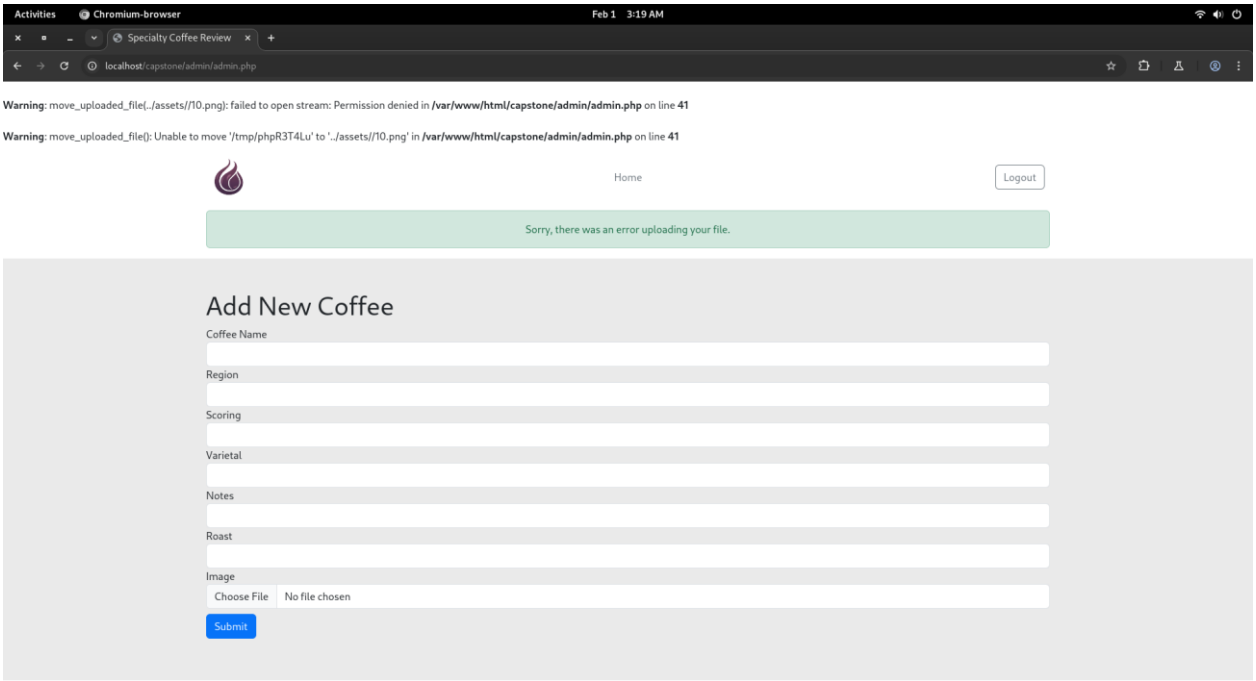


fig-16: Error giving in uploading png files

Expected Results of Uploading a Shell via Burp Suite

- **Bypass File Upload Restrictions** → Modify `POST` request, change `Content-Type`, remove filters.
- **Upload Web Shell** → Successfully store `shell.php`, `shell.jsp`, etc.
- **Execute Shell** → Access via `http:/localhost/capstone/uploads/shell.php`.
- **Remote Code Execution (RCE)** → Run system commands (`id`, `whoami`, `ls`).
- **Gain Reverse Shell** → Use Netcat (`nc -lvnp 4444`).
- **Privilege Escalation** → Exploit misconfigurations for root access

Actual Result:

1. For giving error in uploading png file I have failed to do anything further

Describing the vulnerability and its impact:

1.SQL Injection

Description:

- Exploiting a vulnerability in database queries by injecting malicious SQL code.
- Example: `' OR 1=1 --` (bypasses authentication).

Impact:

- Unauthorized access to databases.
- Data theft (usernames, passwords, emails).
- Database modification or deletion.

2.XSS (Cross-Site Scripting) Payloads

Description:

- Injecting malicious scripts into a web page, executed in a user's browser.
- Example Payload: `<script>alert('XSS')</script>`.

Impact:

- Stealing cookies/session tokens.
- Defacing websites.
- Redirecting users to malicious sites.

3.Shell Upload

Description:

- Uploading a malicious file (e.g., `shell.php`) to gain control of a server.
- Bypassing file type restrictions via Burp Suite.

Impact:

- Remote Code Execution (RCE).
- Full control over the server.
- Data theft, malware deployment, or complete site takeover.