Penetration Test - Standard

Application: Capstone Web App

Client: Capstone

Table of Contents

Contacts

Contact	Organization	Email
Khulah Faisal Sardar	Knowledge Streams	sardarkhulahfaisal@gmail.com
Syed Abeer Ahmed	Knowledge Streams	syedabeer@gmail.com

Executive Summary

Capstone has engaged Knowledge Streams to perform a Penetration Test – Standard assessment on the Capstone web application. The purpose of this assessment was to assess the overall security posture of the application from a black-box perspective. This includes determining the application's ability to resist common attack patterns and identifying vulnerable areas in internal or external interfaces that may be exploited by a malicious user. During the assessment, Knowledge Streams identified 5 findings characterized as follows:

- 1 **Critical** Severity Findings
- 6 **High** Severity Finding
- 1 **Medium** Severity Findings
- 0 **Low** Severity Findings
- 0 **Minimal** Severity Findings

Observations

The findings from the penetration testing exercise reveal critical and high-severity vulnerabilities within the system. The critical vulnerabilities encompass issues such as client-side manipulation of product ratings, potentially leading to altered product information. These critical flaws demand immediate attention and the implementation of input validation measures on the server side. Moreover, high-severity concerns include unrestricted file uploads enabling PHP execution and SQL injection risks that grant unauthorized access to databases, potentially exposing sensitive user information. Remediation for these vulnerabilities involves stringent content and file extension validations, input sanitization, and the use of prepared statements to prevent SQL injection attacks. Additionally, identified weaknesses in password hashing mechanisms emphasize the need for stronger algorithms and regular security monitoring. The medium-severity vulnerabilities, notably the cross-site scripting concerns, accentuate the importance of input validation and robust security mechanisms against malicious script injections. These observations underscore the necessity for comprehensive security measures, ranging from input validation to directory access controls, to fortify the system against potential unauthorized access, data breaches, and manipulative attacks.

Recommendations

It is paramount that the 1 critical findings be resolved as soon as possible to prevent the ease of an attacker gaining elevated privileges with little effort.

Tentative investigation was performed on the scale of the vulnerability of the XSS and XXE attack vectors. They have a moderate potential for being open to more malicious attacks than the ones performed in this test, so they must also be remediated as soon as possible.

1 Introduction

Capstone has engaged Knowledge Streams to perform a Penetration Test - Standard on Capstone Web App beginning on 30/10/2023 and ending on 08/11/2023.

Scope

The scope of this assessment was limited to components and interfaces specific to the Capstone Web App. The following endpoint was considered in scope:

- http://172.19.16.243/capstone/index.php

2 Methodology

Risk Assessment Methodology

Risk Assessment Table

The severity assigned to each vulnerability is calculated using Common Vulnerability Scoring System (CVSS v3.1) standard. CVSS scoring methodology is based on 3 groups: Base, Temporal, and Environmental. The Base group determines the risk score specific to the vulnerability. The Temporal group determines the temporary vulnerability score subject to change over time. The Environmental score is based on user/application environment. In this penetration test, the CVSS score of vulnerabilities was evaluated only using the Base metrics.

More information on the CVSS v3.1 standard for risk assessment can be found at the link below:

https://www.first.org/cvss/specification-document

Risk Level	Definition
Critical	**CVSS v3.1 Score: 9.0 to 10.0** This finding will compromise Confidentiality and/or Integrity and/or Availability. These findings represent an important risk to the application's security; therefore it is a top priority and must be remediated in an immediate manner (or risk accepted).
High	**CVSS v3.1 Score: 7.0 to 8.9** This Finding on its own will compromise Confidentiality and/or Integrity and/or Availability of a significant data element. These findings represent an elevated and important risk to the application's security; hence this must be considered a top priority to remediate and must be remediated (or risk accepted).
Medium	**CVSS v3.1 Score: 4.0 to 6.9** This Finding will compromise Confidentiality and/or Integrity and/or Availability of a significant data element but requires one or more "pre-conditions" to exist. These findings represent a significant but less important risk to the application's security in that should the pre-conditions be introduced to the environment, so too would the significant risk. These findings should be addressed quickly and must be remediated (or risk accepted).
Low	**CVSS v3.1 Score: 0.1 to 3.9** A less important risk to the application's security – this should be addressed within a reasonable time unless there is business justification not to.
Minimal	**CVSS v3.1 Score: 0.0** Potential for some risk to the application's security – this is used to identify discussion items in order to determine whether remediation is necessary.

3 Findings

Summary of Findings

Finding	CWE ID	CVSS Score	Severity	Status
Client-Side Enforcement of Server-Side Security	602	9.1	Critical	Open
Use of Weak Hash	328	8.7	High	Open
SQL Injection	89	8.5	High	Open
Reverse Shell via Unrestricted Upload of File with Dangerous Type	434	8.0	High	Open
Unrestricted Upload of File with Dangerous Type	434	8.0	High	Open
Exposure of Information Through Directory Listing	548	7.5	High	Open
Cross-site Scripting (Stored)	79	7.4	High	Open
Cross-site Scripting (Reflected)	79	5.4	Medium	Open

3.1 Critical Severity Findings

Client-Side Enforcement of Server-Side Security

CVSS:3.1/AV:cN/AC:L/PR:L/UI:N/S:C/C:L/I:H/A:L CVSS Score: 9.1

CWE-602: Client-Side Enforcement of Server-Side Security

Description

This vulnerability allows to change the rating of any product other than the given rating options.

Instances

http://0.tcp.ap.ngrok.io:12261/capstone/coffee.php?coffee=2

Username: abeer

Password: 123

Role: user

Steps To Reproduce

1. Login with the user/admin account.
2. Open the detail page of any coffee and add rating from the given option then add that post request to the burp suite.
3. Change the rating to any other option that is not given in the given rating list and it will be get updated.

Evidence

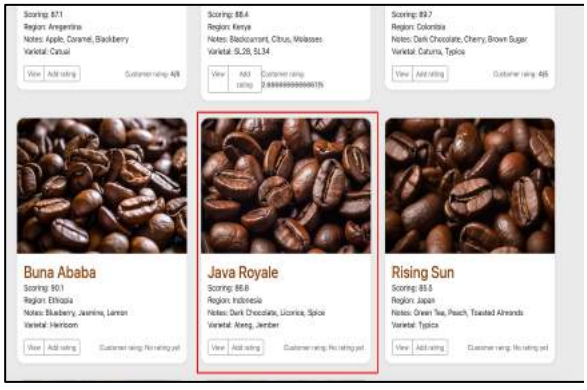

Figure 1: Home Page

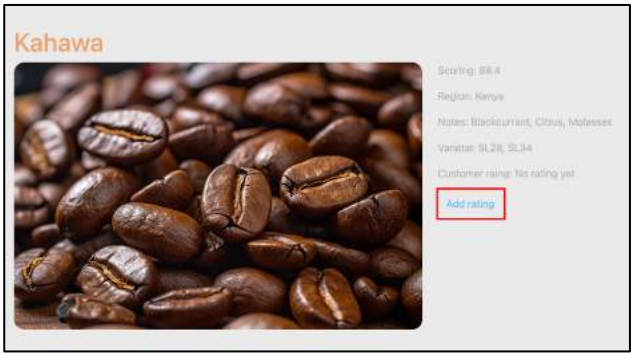

Figure 2: Coffee Details

Figure 3: Rating Popup

Figure 4: Burp Suite

Impact

This vulnerability can affect the business of the capstone since crucial information about the product could be changed.

Remediation

Impose input sanitization/validation on the server side as well.

3.2 High Severity Findings

Use of weak Hashes

CVSS: 3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N **CVSS Score: 8.7**

CWE-328: Use of weak Hashes

Description

This vulnerability allows to decrypt the hash password.

Instances

http://172.19.16.243/capstone/index.php

Steps To Reproduce

1. Using SQLMap to get the details of the databases.
2. Then we get the tables name and further we use the **user table**.
3. After that we run the command to get user's table data.
4. Now first of all we checked the hash type and by using online tool we know this is **bcrypt** hash type.
5. Then by using hash-Cat and file xato-net-10-million-passwords=10000.txt we bruteforce the password.
6. And finally, we get admin's password i.e captain1

Evidence

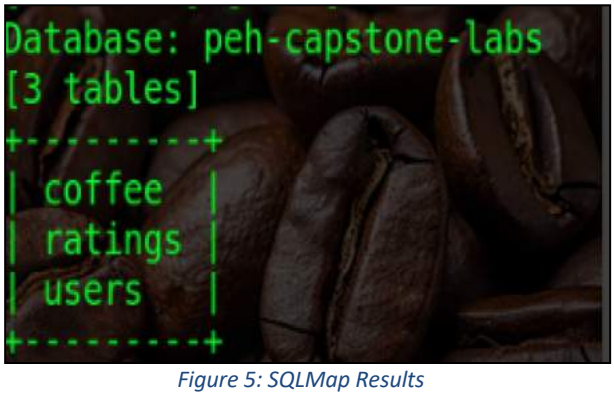

Figure 5: SQLMap Results

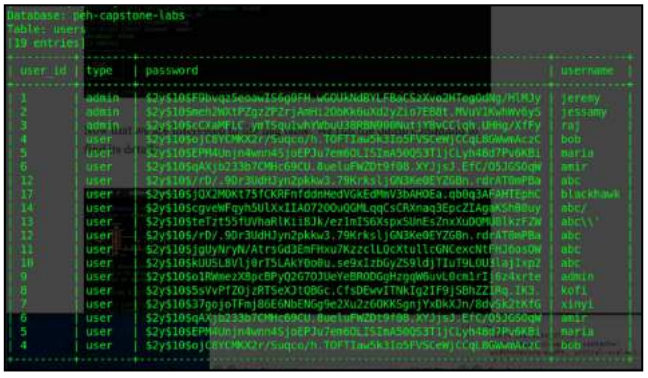

Figure 6: SQLMap Results

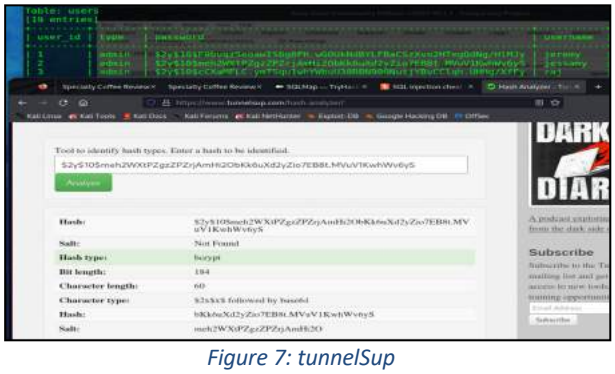

Figure 7: tunnelSup

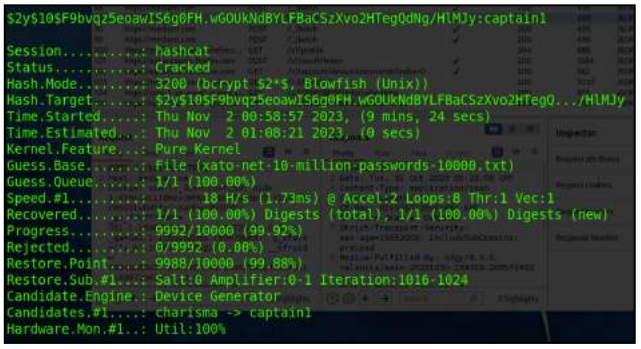

Figure 8: Hash-Cat Results

Impact

Weak hashes can result in password breaches, identity theft, and data exposure.

Remediation

Remediation for weak hashes involves upgrading to strong hashing algorithms, salting, and using key derivation functions. Regularly monitor and educate teams to strengthen password security

Unrestricted Upload of File with Dangerous Type

CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H CVSS Score: 8.0

CWE-434: Unrestricted Upload of File with Dangerous Type

Description

This vulnerability allows to upload a php file allowing us to execute shell commands on the web server using parameter in the URL.

Instances

http://172.19.16.243/capstone/index.php
http://0.tcp.ap.ngrok.io:12261/capstone/admin/admin.php

Username: jeremy
Password: captain1
Role: Admin

Steps To Reproduce

1. Login with the admin account using the SQL injection password.
2. Navigate to the admin.php webpage found through the Exposure of Information Through Directory Listing.
3. Upload a normal image and capture the post request in the Burp Suite.
4. Change the extension in the filename to .php and add one-liner php script to the end of the content of image inside the burp suite.
5. Upload the file to the website.
6. Open the image of the newly added coffee form the home page in a new window.
7. Replace .png to php in the URL request.
8. Execute the shell command in the cmd parameter in the URL request.

Evidence

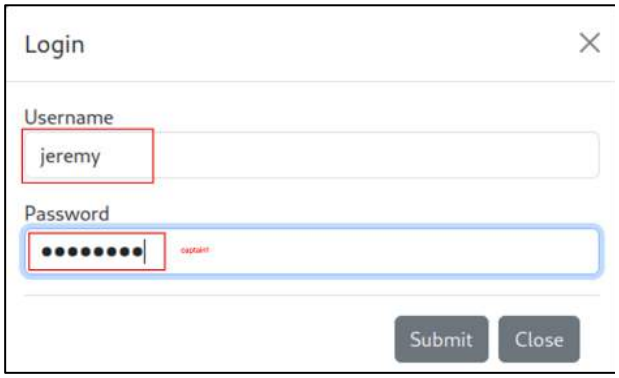

Figure 9: Login Popup

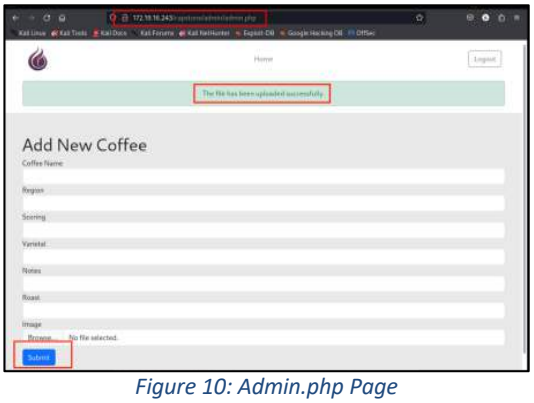

Figure 10: Admin.php Page

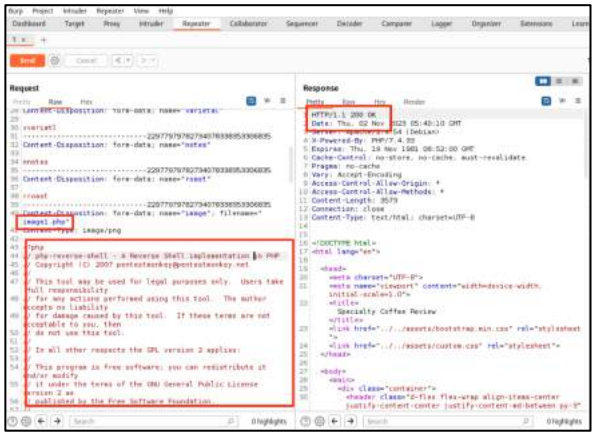

Figure 11: Burp Suite

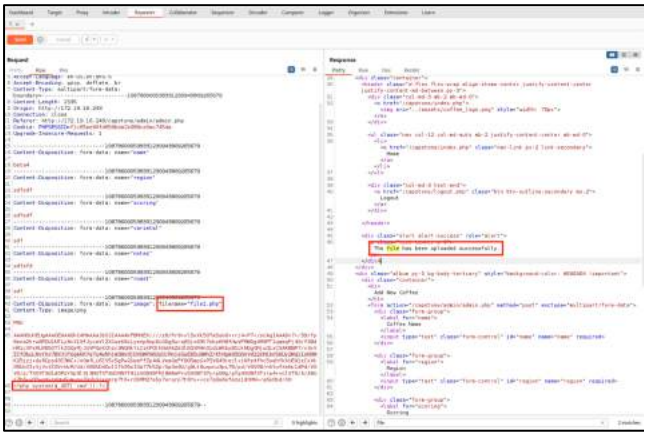

Figure 12: Burp Suite

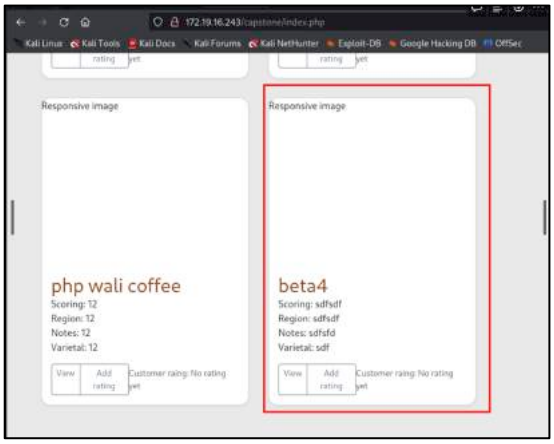

Figure 13: Home Page

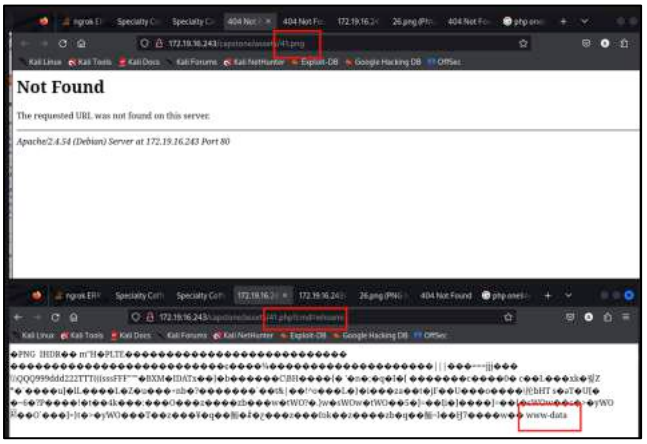

Figure 14: Browser

Impact

Allowing shell command execution through URL parameters in a web application can lead to unauthorized access, data breaches, server compromise, and reputational damage.

Remediation

Check file extensions as well along with content of the file, use both black list and white list blocking methods.

Reverse Shell via Unrestricted Upload of File with Dangerous Type

CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H **CVSS Score: 8.0**

CWE-434: Unrestricted Upload of File with Dangerous Type

Description

This vulnerability allows to upload a php file instead of png allowing us to get reverse shell via file upload.

Instances

http://172.19.16.243/capstone/index.php
http://0.tcp.ap.ngrok.io:12261/capstone/admin/admin.php

Username: jeremy
Password: captain1
Role: Admin

Steps To Reproduce

1. Login with the admin account using the SQL injection password.
2. Navigate to the admin.php webpage found through the Exposure of Information Through Directory Listing.
3. Upload a normal image and capture the post request in the Burp Suite.
4. Change the extension in the filename to .php and add the php reverse-shell script to the end of the content of image inside the burp suite.
5. Upload the file to the website.
6. Start the listener first in your machine.
7. Open the image of the newly added coffee form the home page in a new window.
8. Replace .png to php in the URL request.

Evidence

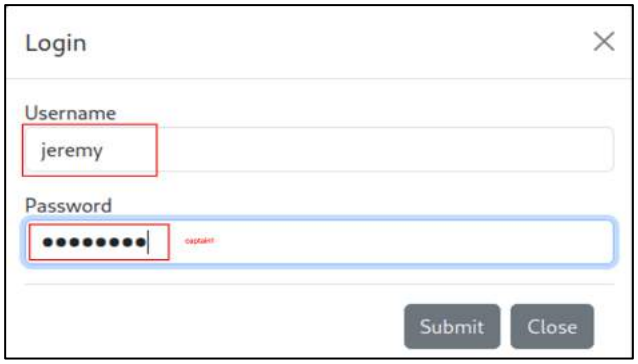

Figure 15: login Popup

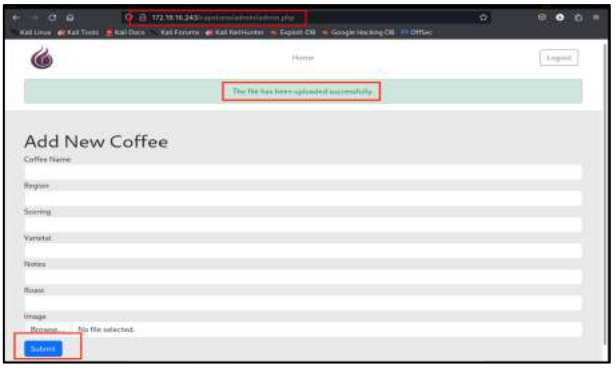

Figure 16: Admin.php

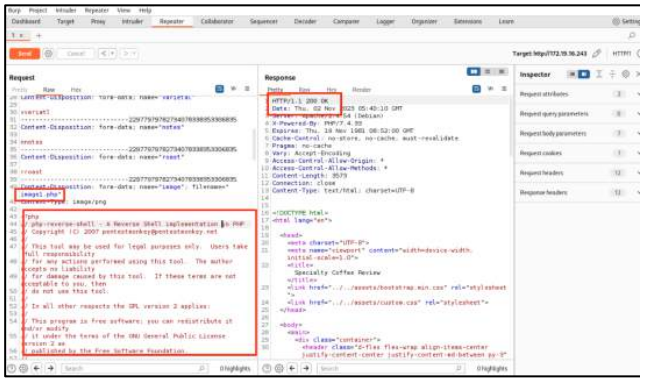

Figure 17: Burp Suite

Figure 18: Burp Suite

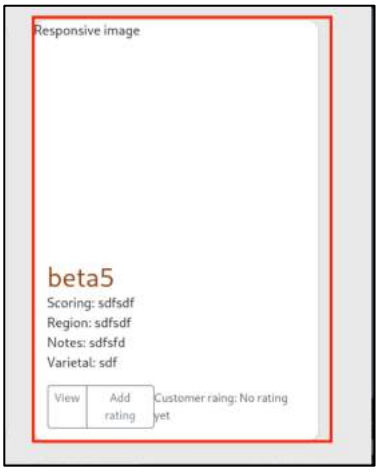

Figure 19: Coffee Tiles

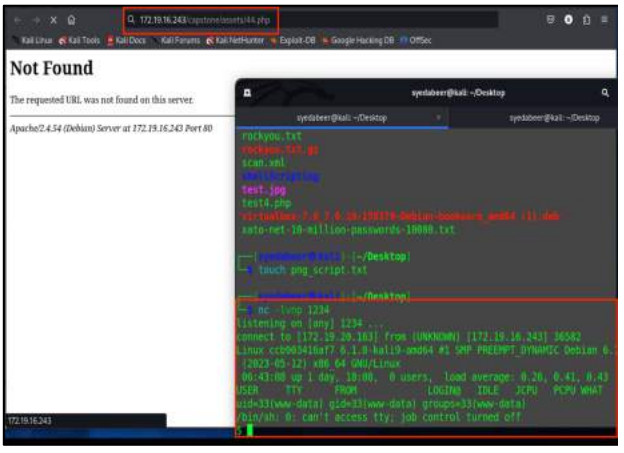

Figure 20: Reverse Shell + Web Browser

Impact

Reverse shell can lead to unauthorized access, data breaches, server compromise, and reputational damage.

Remediation

Check file extensions as well along with content of the file, use both black list and white list blocking methods.

SQL Injection

CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:H CVSS Score: 8.5

CWE-83: SQL Injection

Description

This vulnerability allows to us to access all the databases of the capstone web app , check users credentials and other valuable information as well.

Instances

http://0.tcp.ap.ngrok.io:12261/capstone/coffee.php?coffee=2
Username: abeer
Password: 123
Role: user

Steps To Reproduce

1. Login with the your user account.
2. Navigate to the page of a coffee detail and use it as your required URL for SQL injection.
3. Use SQLMap for injection and you will receive all the details of all the databases.

Evidence

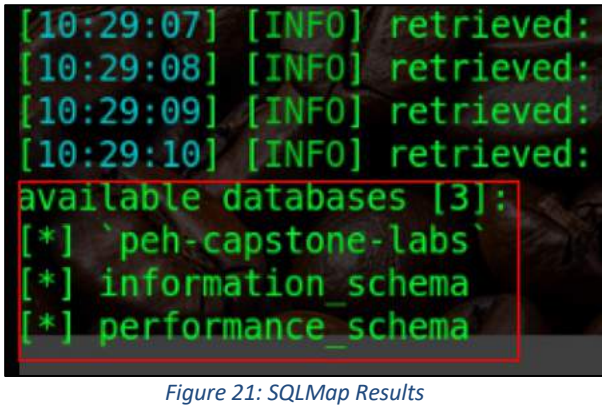

Figure 21: SQLMap Results

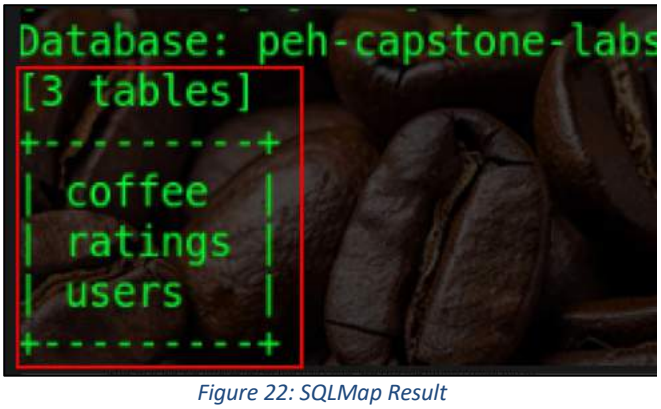

Figure 22: SQLMap Result

Figure 23: SQLMap Results

Impact

The attacker is able to get access to all the databases including the confidential information as well such the username and passwords of all users which could leads to further damages.

Remediation

SQL Injection attacks can be blocked by using input validation and parametrized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes.

Exposure of information through Directory listing

CVSS: 3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N **CVSS Score: 7.5**

CWE-548: Exposure Of Information Through Directory Listing

Description

Here through directory busting we found sensitive files and directories.

Instances

http://172.19.16.243/capstone/index.php

Steps To Reproduce

1. Open terminal and run dirbuster command.
2. Give common.txt file as wordlist, give url of the home page and click on start directory busting.
3. From here we get the directory to access admin page.

Evidence

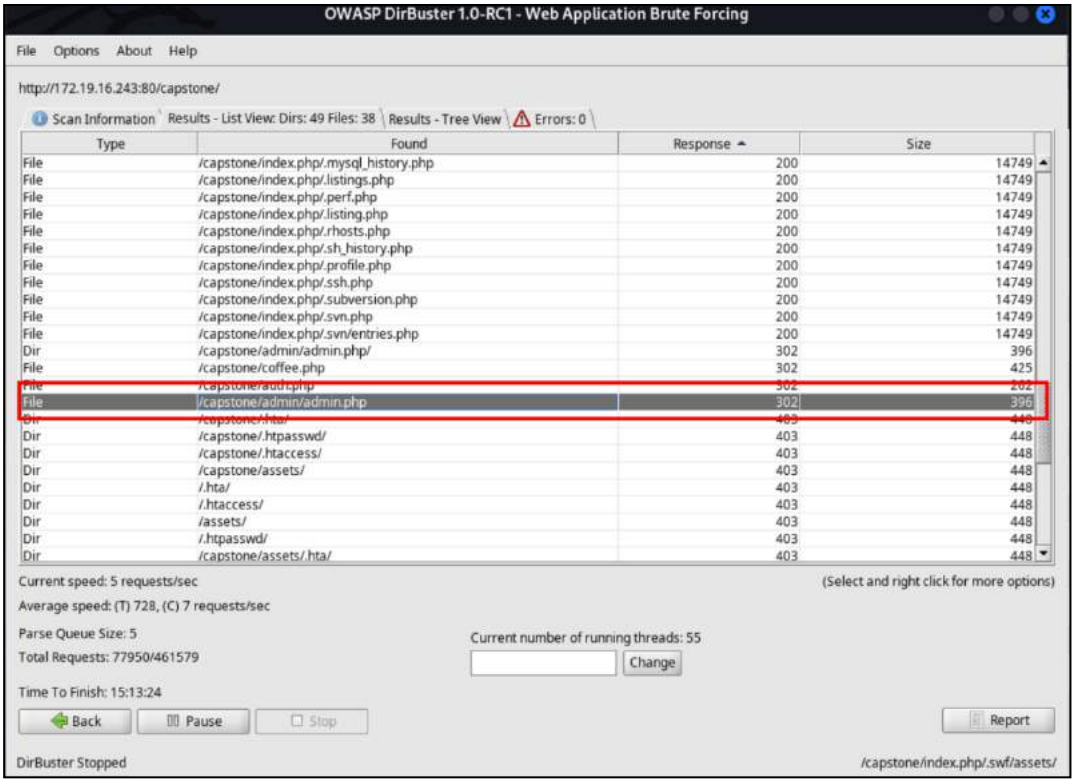

Figure 24: Dirbuster

Impact

This vulnerability cause exposure of files and directories.

Remediation

To remediate Exposure of Information through Directory Listing, disable directory indexing and ensure that sensitive files and directories are secured with proper access controls.

Cross –site-scripting(stored)

CVSS: 3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:L **CVSS Score: 7.4**

CWE-79: Cross-Site-Scripting (Stored)

Description

This vulnerability allows to inject malicious scripts into web page permanently viewed by other users. These scripts can be executed in user browser whenever anyone visits this page.

Instances

http://172.19.16.243/capstone/index.php
http://172.19.16.243/capstone/coffee.php?coffee=2

Username: abeer
Password: 123
Role: user

Steps To Reproduce

1. Firstly, we have to login using user credential.
2. Then click on kahawa(coffee type) and click on add rating and write malicious script in comment section, this is stored in the webpage permanently.
3. Then click on submit button and it will run the script when anyone visits the page again.

Evidence

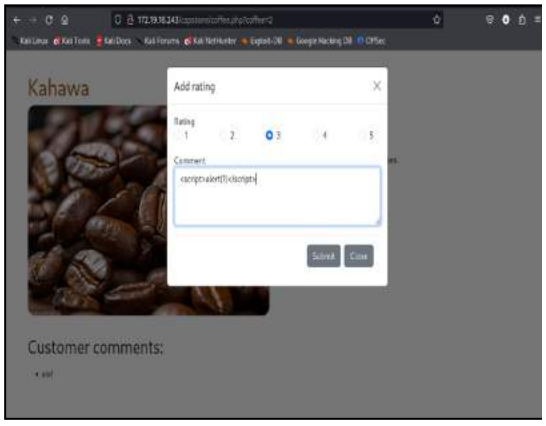

Figure 25: Rating Popup

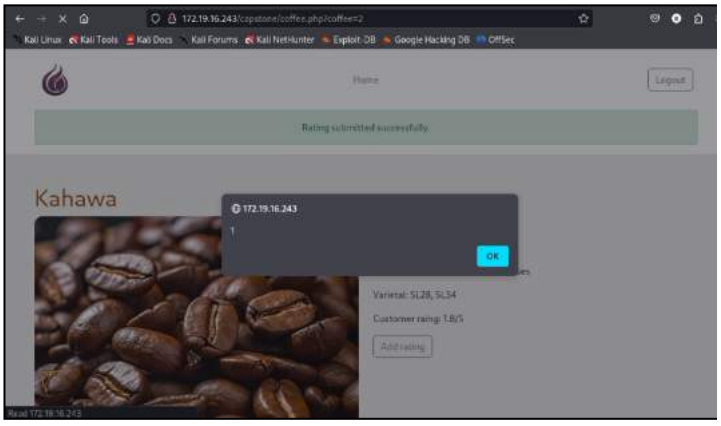

Figure 26: Script Output

Impact

Cross-Site Scripting (XSS) can lead to the theft of user data, session hijacking, and unauthorized actions, potentially compromising user privacy and system integrity. It also enables to deface websites and deliver malware to users.

Remediation

Remediation for XSS involves input validation, output encoding, and using security mechanisms.

3.3 Medium Severity Findings

Cross –site-scripting(reflected)

CVSS: 3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:L **CVSS Score: 5.4**

CWE-79: cross-site-scripting(reflected)

Description

Here we inject vulnerable malicious scripts into web pages via URL's links and then reflected back to user in web page response.

Instances

http://172.19.16.243/capstone/index.php
http://172.19.16.243/capstone/coffee.php?coffee=2

Username: abeer
Password: 123
Role: user

Steps To Reproduce

1. Firstly, we have to log out from user account.
2. Then click on URL link type the malicious script.
3. Then press the enter button and it will run the script and display alert box

Evidence

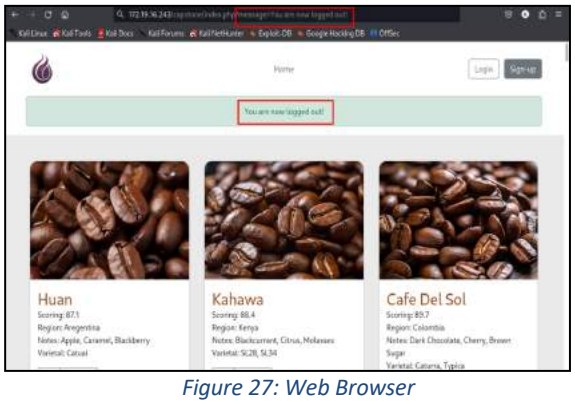

Figure 27: Web Browser

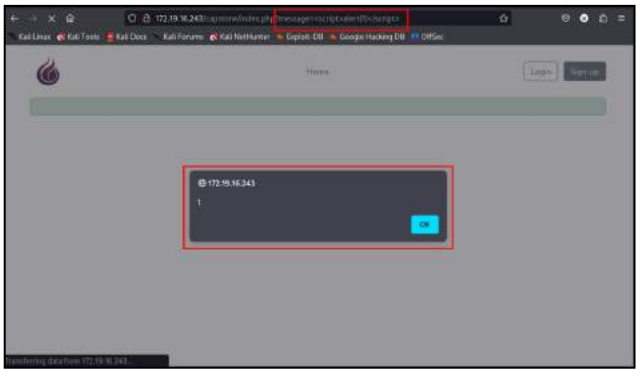

Figure 28: Script Output

Impact

Cross-Site Scripting (XSS) can lead to the theft of user data, session hijacking, and unauthorized actions, potentially compromising user privacy and system integrity. It also enables to deface websites and deliver malware to users.

Remediation

Remediation for XSS involves input validation, output encoding, and using security mechanisms.