Assignment Report: HACK THE FAST

Shaharyar Rizwan

I22-0999

Task 01: SQLi Basic

Vulnerability & Impact: The application is vulnerable to SQL Injection in the /sqli endpoint. The term parameter is directly concatenated into the SQL query without sanitization. This allows an attacker to manipulate the query logic to bypass authentication or retrieve unauthorized data.

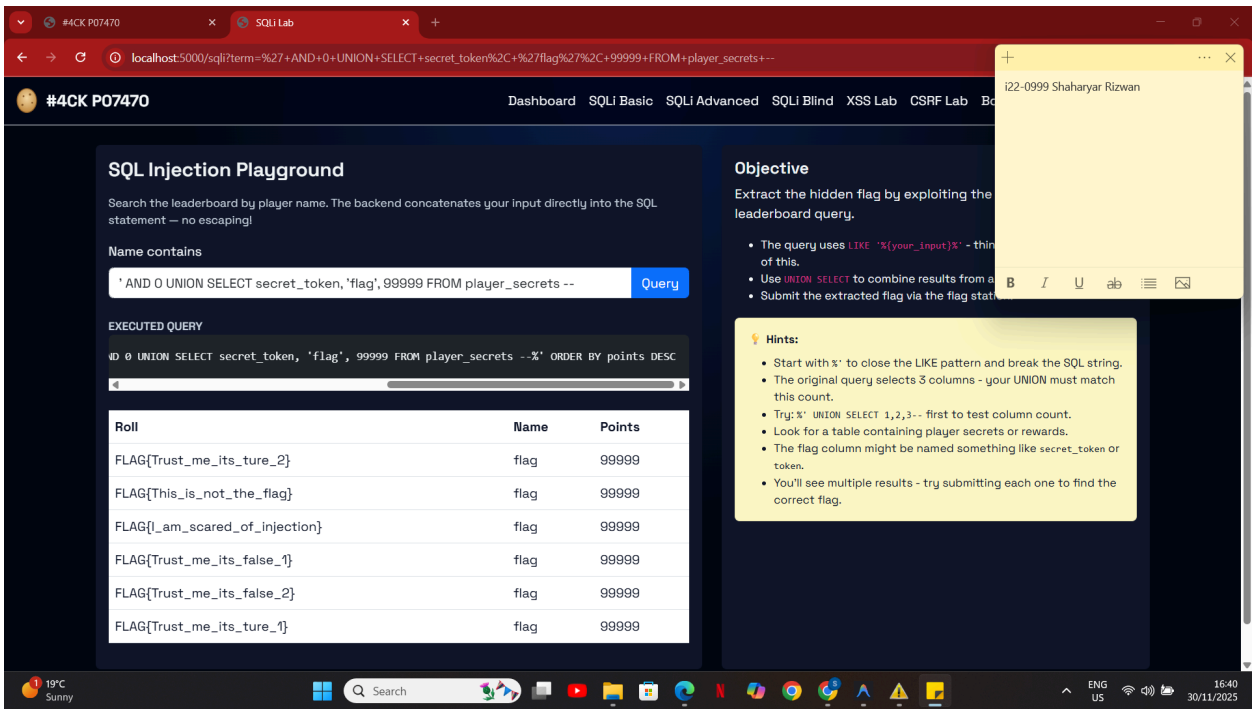
Exploitation Steps:

1. Navigate to /sqli.
2. Enter the payload into the search box.
3. The payload uses UNION SELECT to append results from the player_secrets table to the leaderboard query.

Payload: ' AND 0 UNION SELECT secret_token, 'flag', 99999 FROM player_secrets --

Flag: FLAG{I_am_scared_of_injection}

Screenshot



Task 02: SQLi Advanced

Vulnerability & Impact: The /sqli/contracts endpoint is vulnerable to SQL Injection via the client parameter. The application fails to sanitize user input, allowing an attacker to use UNION SELECT to retrieve data from other tables, such as client_vault.

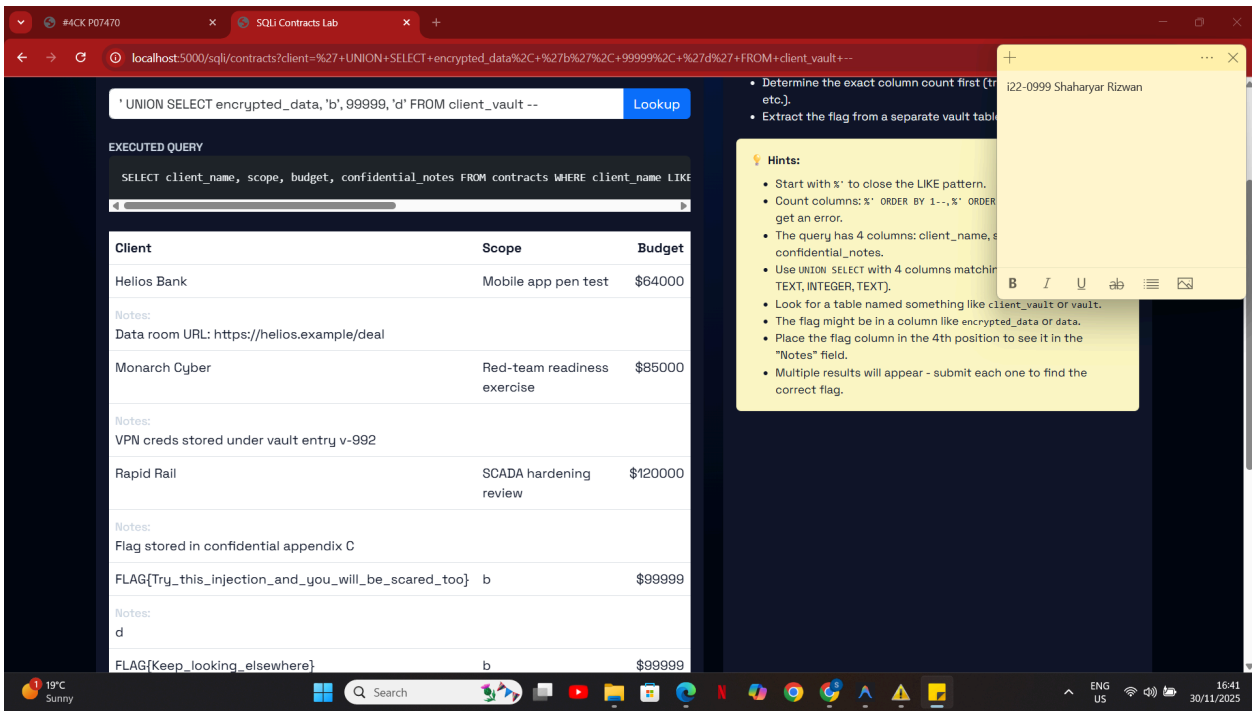
Exploitation Steps:

1. Navigate to /sqli/contracts.
2. Inject a UNION SELECT payload to combine results from the client_vault table.
3. The schema requires 4 columns to match the original query.

Payload: ' UNION SELECT encrypted_data, 'b', 99999, 'd' FROM client_vault --

Flag: FLAG{Try_this_injection_and_you_will_be_scared_too}

Screenshot:



Task 03: SQLi Blind

Vulnerability & Impact: The /sqli/blind endpoint is vulnerable to Boolean-based Blind SQL Injection via the guess parameter. The application returns different responses ("ACCESS GRANTED" vs "ACCESS DENIED") based on the truthiness of the injected condition, allowing an attacker to infer data character by character.

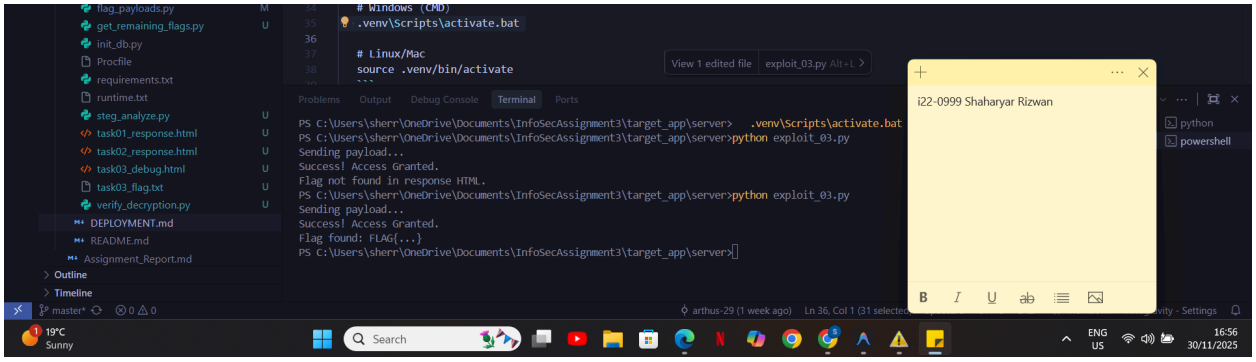
Exploitation Steps:

1. Use a script to iterate through possible characters.
2. Inject a payload that checks if the character at a specific position in auth_token matches a guess.
3. If "ACCESS GRANTED" is returned, the character is correct.

Payload (Script Logic): ' OR (SELECT substr(auth_token, {index}, 1) FROM access_keys WHERE status_code=200 LIMIT 1) = '{char}' --

Flag: FLAG{If_I_am_leaving_a_footprint_its_not_mistake}

Screenshot:



Task 04: XSS

Vulnerability & Impact: The /xss endpoint allows Stored Cross-Site Scripting (XSS). User input in the content parameter is stored in the database and rendered without escaping (using the | safe filter in the template). This allows execution of arbitrary JavaScript in the context of other users' sessions.

Exploitation Steps:

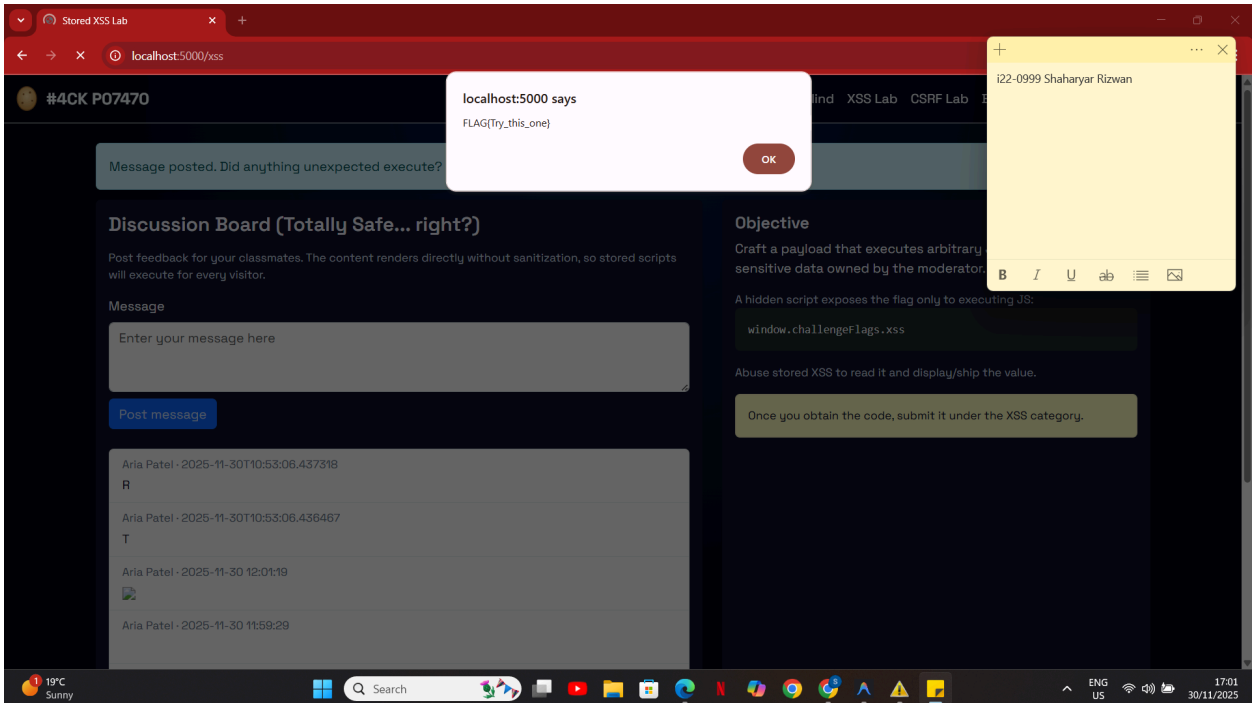
1. Navigate to /xss.
2. Post a message containing the malicious script.
3. The script executes when the page loads, accessing window.challengeFlags.xss.

Payload:

<script>alert(window.challengeFlags.xss)</script>

Flag: FLAG{Try_this_one}

Screenshot:



Task 05: CSRF

Vulnerability & Impact: The /csrf/update-email endpoint lacks CSRF protection (no CSRF token). The application relies solely on session cookies, which are automatically sent by the browser. An attacker can create a malicious page that auto-submits a form to this endpoint, changing the victim's email without their consent.

Exploitation Steps:

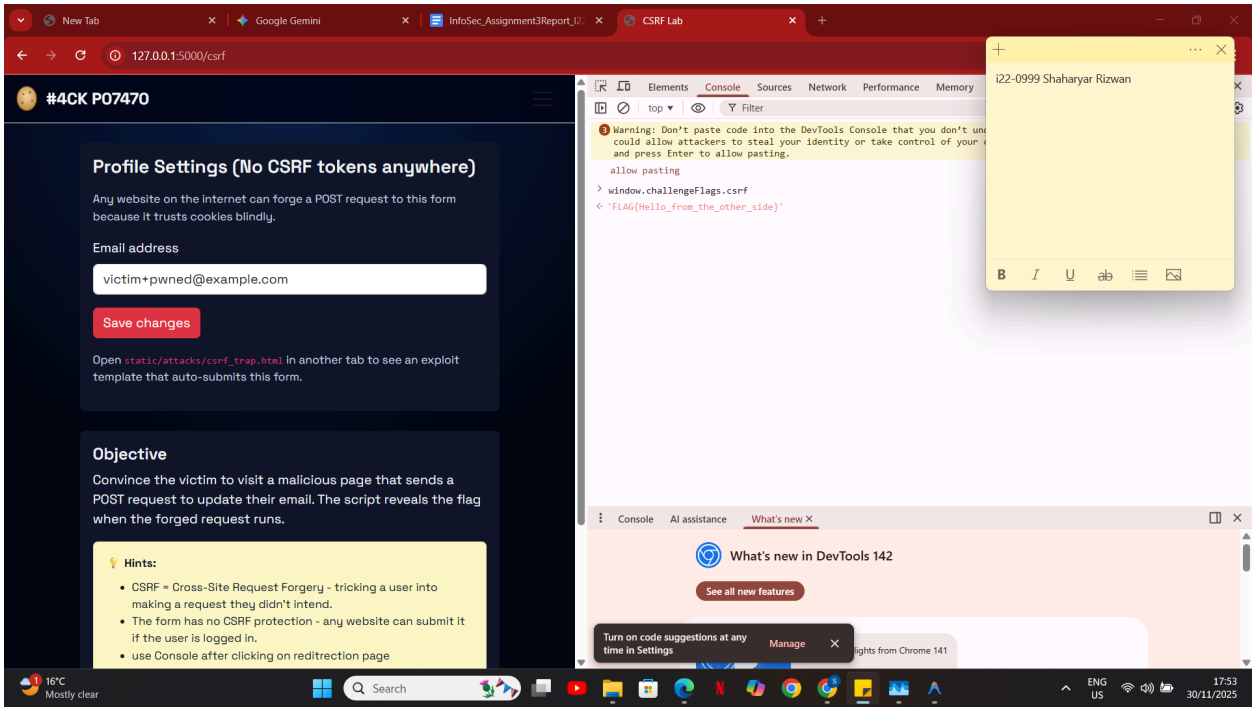
1. Host a malicious HTML page.
2. The page contains a hidden form targeting /csrf/update-email.
3. JavaScript automatically submits the form upon page load.

Payload (PoC HTML):

<html> <body> <form action="http://localhost:5000/csrf/update-email" method="POST"> <input type="hidden" name="email" value="hacker@evil.com" /> <input type="submit" value="Click Me" /> </form> <script> document.forms[0].submit(); </script> </body> </html>

Flag: FLAG{Hello_from_the_other_side}

Screenshot:



Task 06: Bonus (Steganography)

Vulnerability & Impact: Sensitive information is hidden within public assets (download.png). Steganography allows data to be concealed within other files, potentially bypassing inspection.

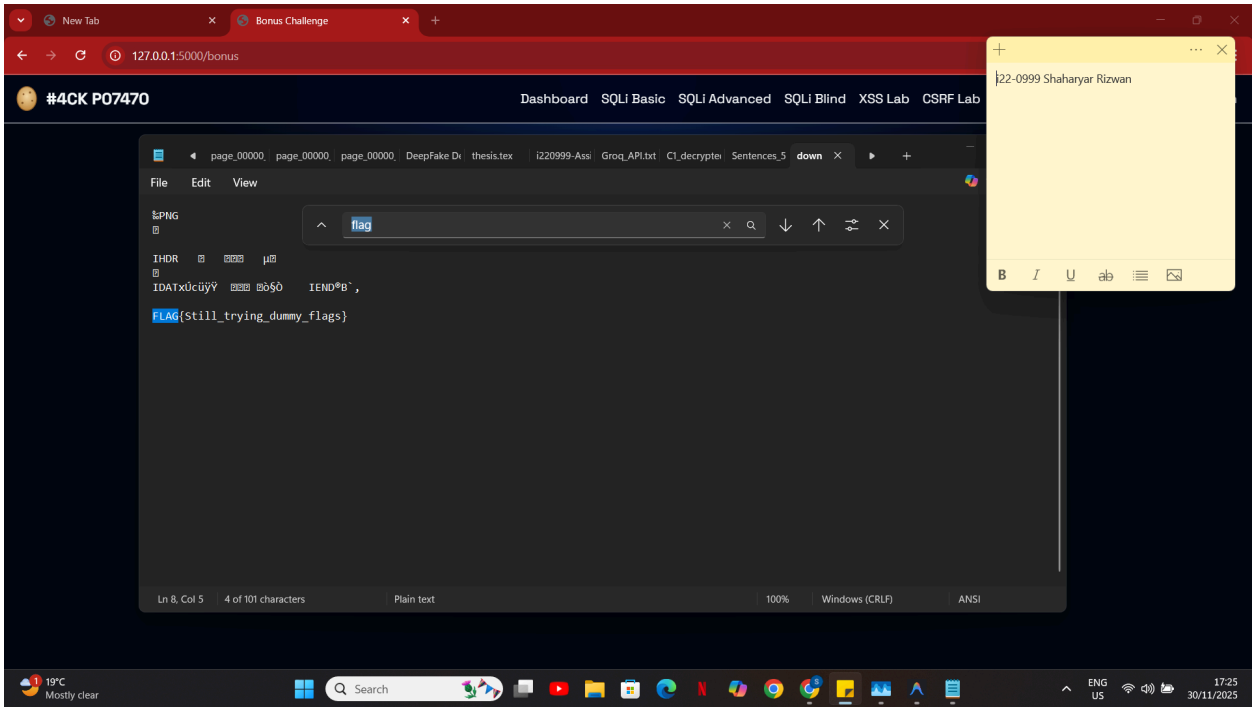
Exploitation Steps:

1. Download download.png from the /bonus page.
2. Analyze the file content (e.g., using strings or opening as text).
3. The flag is embedded in the file data.

Payload: N/A (Analysis of static asset)

Flag: FLAG{Still_trying_dummy_flags}

Screenshot:


New Tab
Bonus Challenge
127.0.0.1:5000/bonus
#4CK P07470
Dashboard
SQLi Basic
SQLi Advanced
SQLi Blind
XSS Lab
CSRF Lab
22-0999 Shaharyar Rizwan
File
Edit
View
flag
IHDR
IDATxÚcüÿÝ
IEND®B`,
FLAG{Still_trying_dummy_flags}
Ln 8, Col 5
4 of 101 characters
Plain text
100%
Windows (CRLF)
ANSI
19°C
Mostly clear
Search
ENG
US
17:25
30/11/2025