Web Application Security Assessment Report

Name: Oyekale Olamide David

Task 1: Web Application Security Testing

Program: Future Interns Cybersecurity Internship

Date: August 2025

Target Application: OWASP Juice shop (Intentionally Vulnerable App)

Task Summary

This assessment focused on the identification of security weaknesses in the OWASP Juice Shop application using Burp suite and manual validation. Several critical vulnerabilities were identified, including User enumeration and Information Leakage, which could lead to targeted attacks, credential compromise, and account takeover. The findings were mapped to the OWASP Top 10 (2021) categories and prioritized based on risk.

Tools Used :

- Kali Linux
- OWASP Juice Shop
- Burp Suite

Procedure

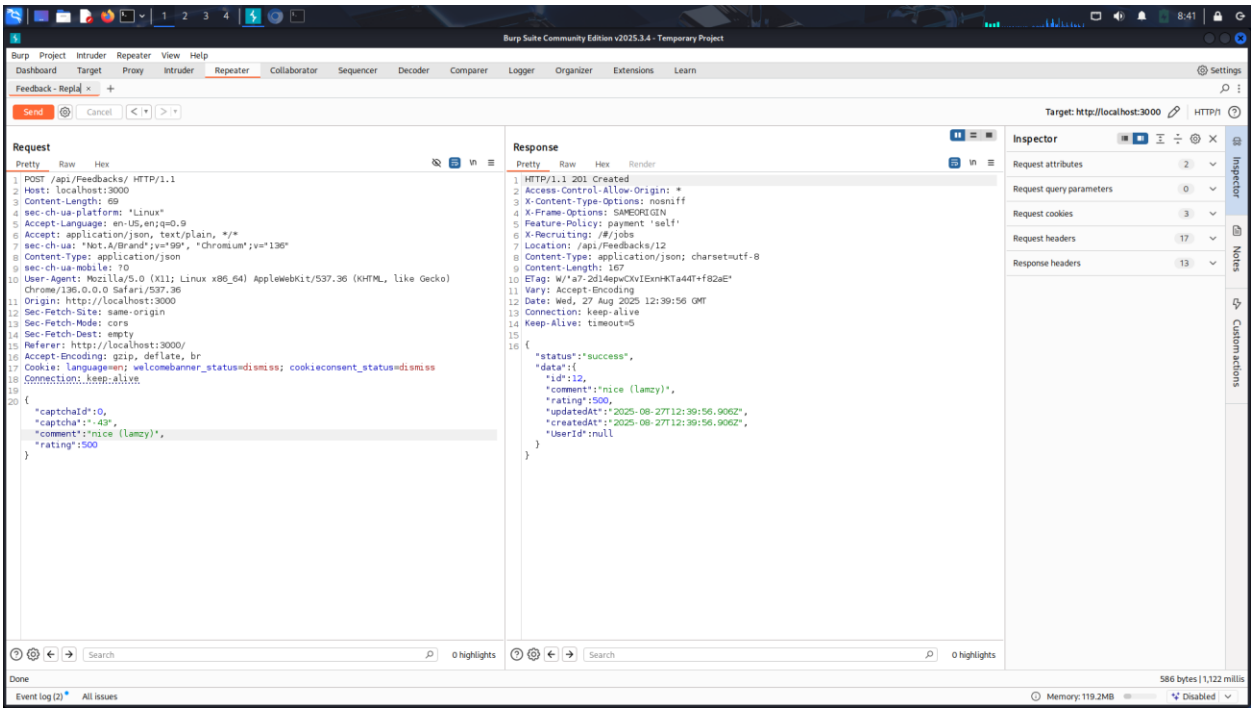
- The OWASP Juice Shop application was deployed locally on a Linux virtual machine, with all required dependencies, including Node.js and npm, installed through the Linux command-line interface, thereby providing a realistic environment for simulating web application attacks.
- The local deployment was designated as the target system for both automated scanning and manual security testing conducted with Burp Suite.
- Burp Suite was used to perform active scanning, enabling the identification of input fields, user interactions, and potential vulnerabilities within the application.
- All identified issues were analyzed and mapped to their corresponding categories in the OWASP Top 10 framework, with remediation measures recommended to address the vulnerabilities.

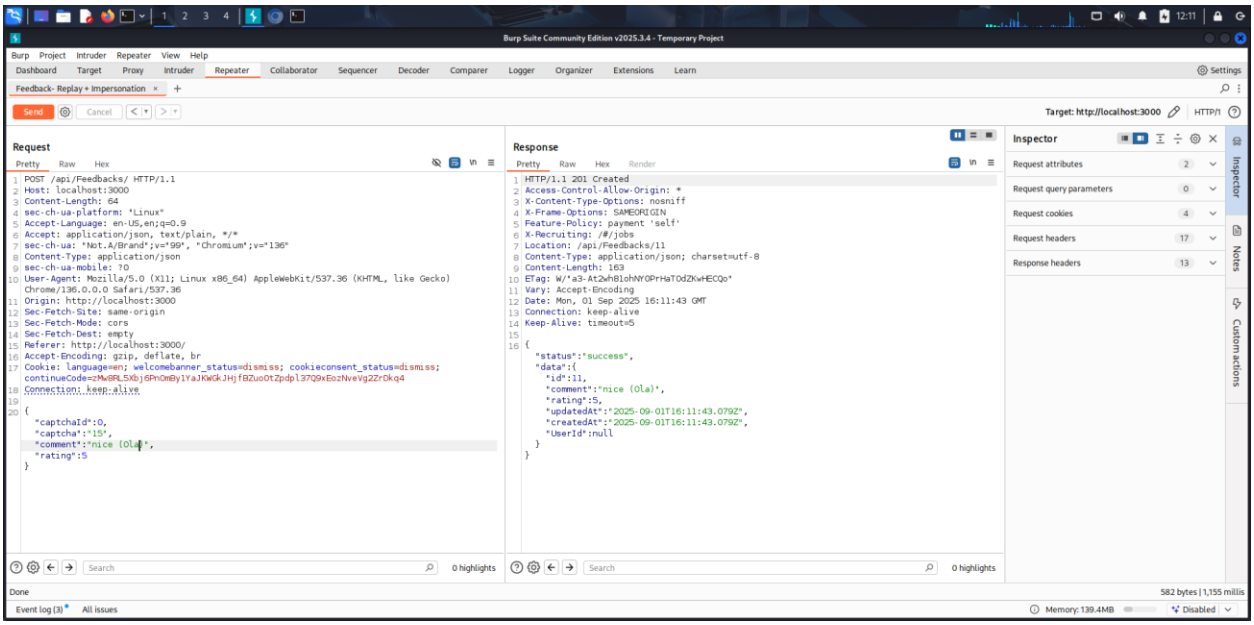
Identified Vulnerabilities

1. Replay and impersonation

The Feedback feature is vulnerable to replay attacks, as previously captured requests can be resent without validation, leading to duplicate submissions. Additionally, weak enforcement of identity validation allows impersonation, enabling attackers to submit feedback under another user's identity. Together, these issues demonstrate broken authentication and access control, exposing the application to abuse and loss of integrity.

- Impact: Enables duplicate submissions and feedback impersonation.
- Risk Rating: Medium
- Evidence:



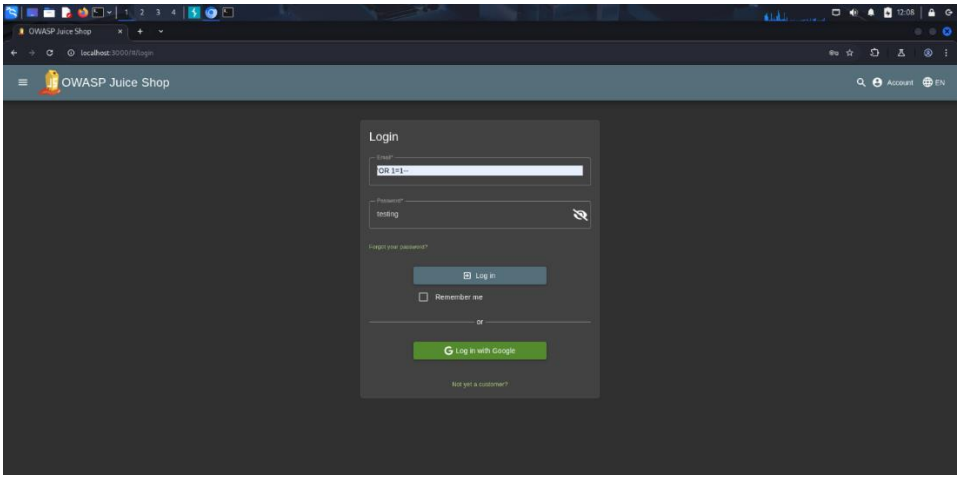


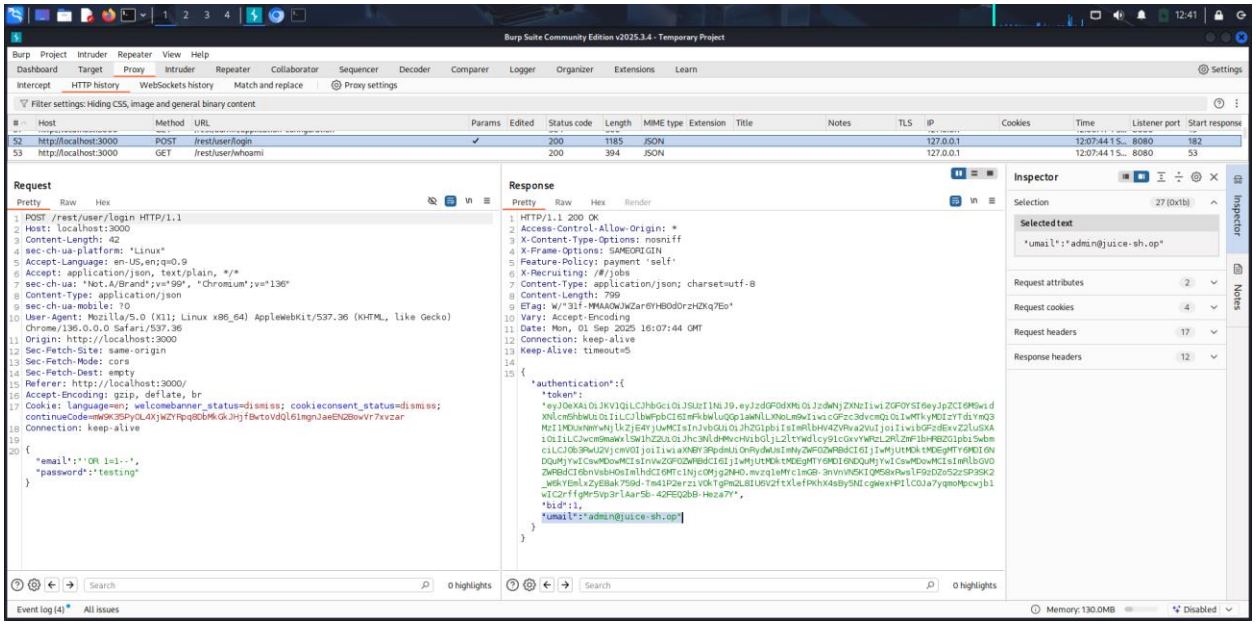
- **Mitigation Strategies:**
  - Implement strict identity validation
  - anti-replay mechanisms (nonces/timestamps)
  - enforce access control on feedback actions.

2. SQL Injection (SQLi)

An SQL Injection vulnerability exists in the login functionality, where crafted inputs (e.g., 'OR 1=1--) manipulate database queries, enabling authentication bypass and unauthorized administrative access, thereby exposing sensitive data and compromising system integrity.

- Impact: Enables authentication bypass and unauthorized administrative access.
- Risk Rating : High
- Evidence:



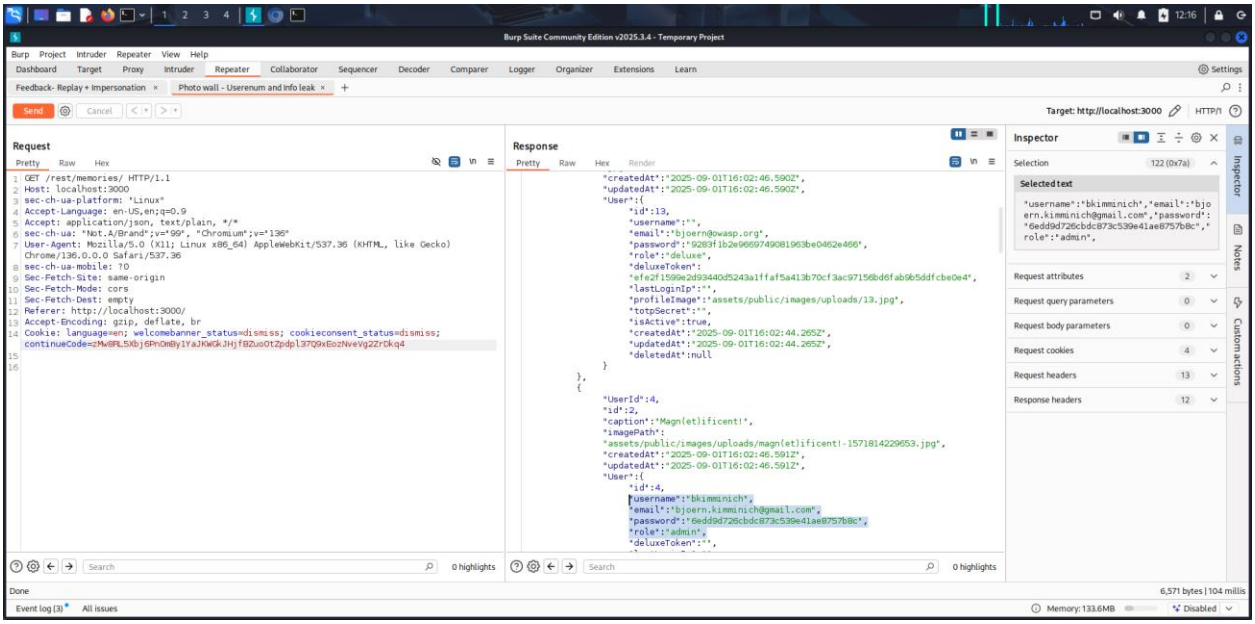


- **Mitigation Strategies:**
    - Apply parameterized queries
    - Validate inputs
    - Restrict database privileges (Apply Least privilege)

3. **User enumeration and information leakage**

   The application's Photo wall reveals valid user accounts through inconsistent responses and exposes sensitive system details, giving attackers both a target list and insights into the underlying environment. Combined, these weaknesses facilitate focused brute-force attempts, credential compromise, and exploitation of known vulnerabilities, ultimately increasing the risk of account takeover and system compromise.

   - Impact: Enables targeted attacks leading to credential compromise and account takeover.
   - Risk Rating: Medium
   - Evidence:

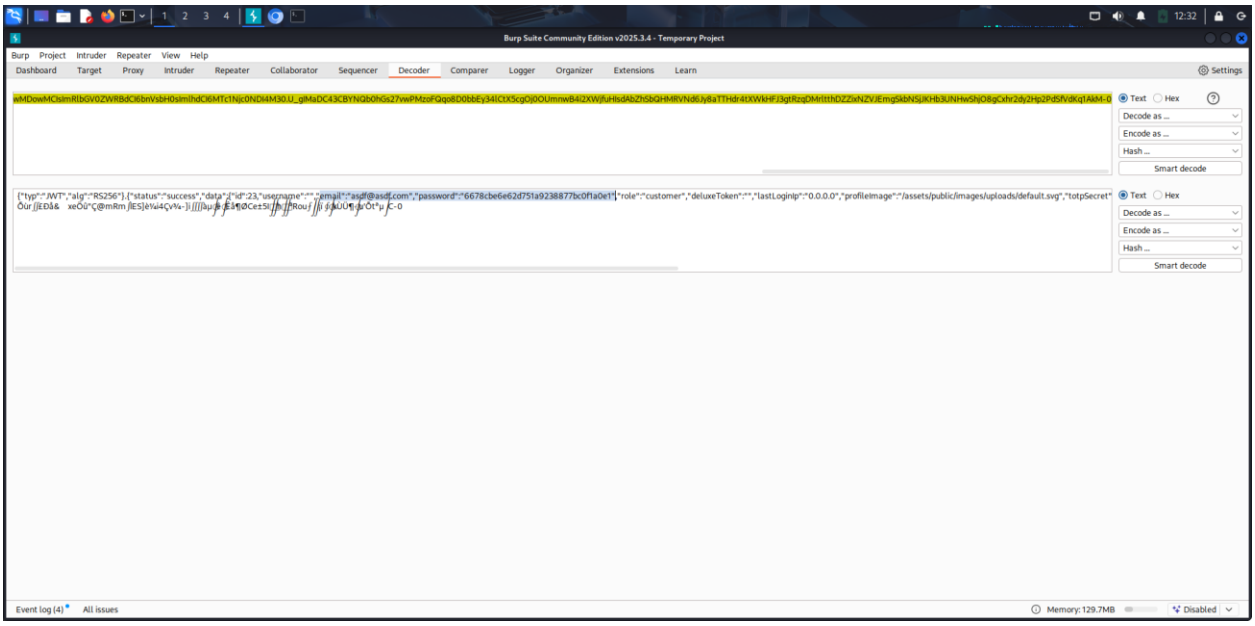


- **Mitigation Strategies:**
  - Standardize error messages to prevent revealing valid or invalid accounts.
  - Apply rate limiting and account lockout to reduce brute-force attempts.
  - Avoid exposing sensitive data in logs, API outputs, or headers.
  - Suppress stack traces, version details, and debug information in responses.

4. Insecure Token Design – Credential exposure in JWT

The application's JSON Web Token (JWT) contains user credentials hashed with MD5, a weak and reversible algorithm. This design flaw exposes sensitive information within the token, allowing attackers to recover or crack user passwords and compromise accounts.

- Impact: Exposes user credentials and enables account takeover.
- Risk Rating: High
- Evidence:

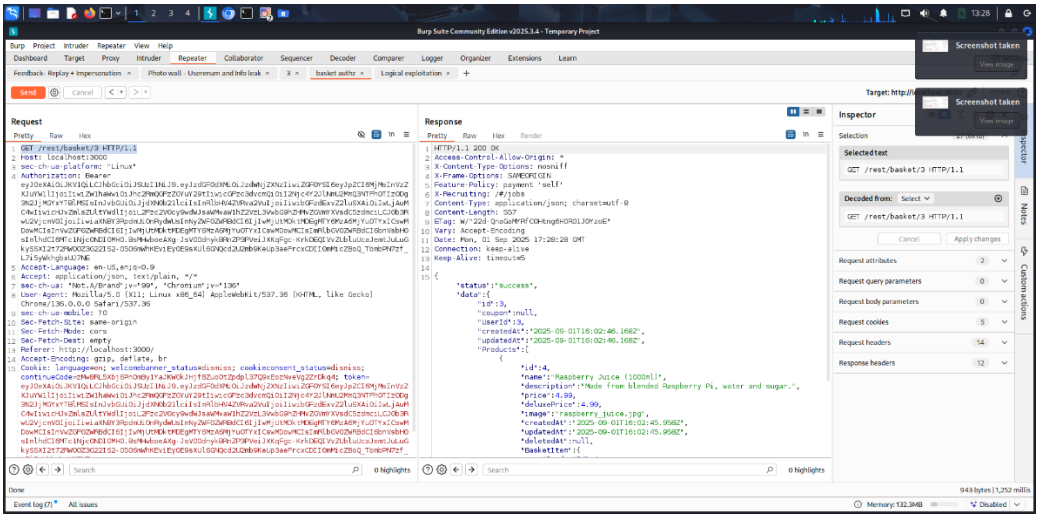


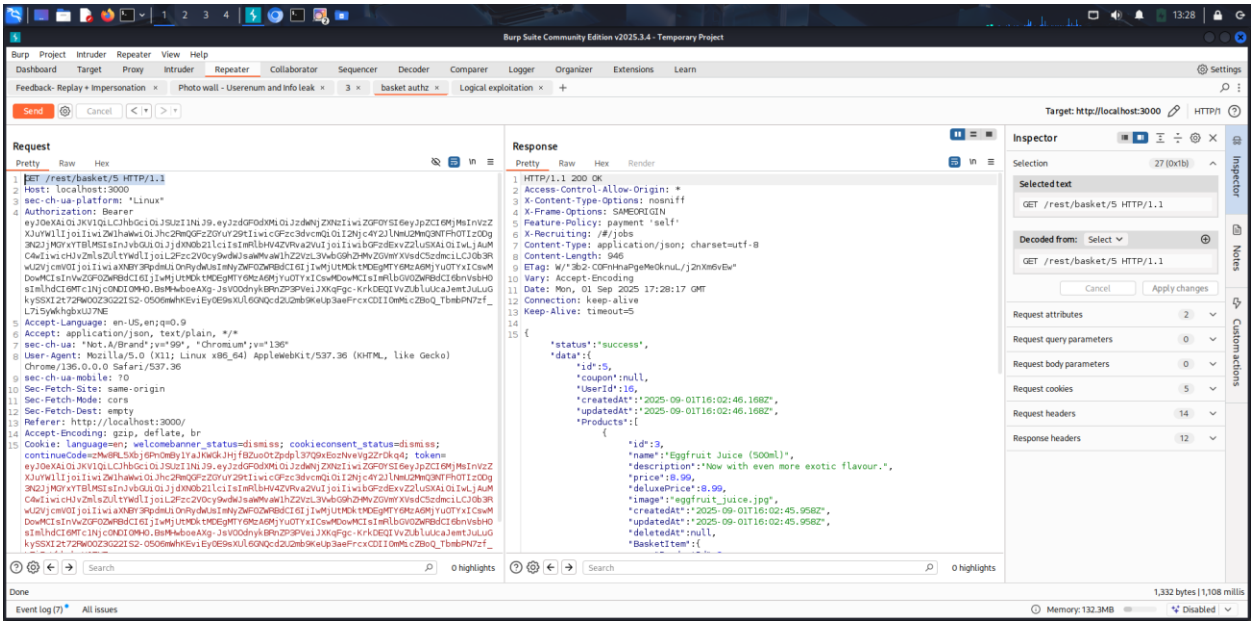
- **Mitigation Strategies:**
  - Never store credentials in tokens
  - Replace MD5 with modern password hashing (e.g., bcrypt, Argon2)
  - Ensure JWTs only contain non-sensitive claims with strong signing algorithms

5. Insecure Direct Object Reference (IDOR)

The application exposes internal object identifiers (e.g., userId, basketId) without proper access control, allowing attackers to manipulate these values to access or modify data belonging to other users.

- Impact: Grants unauthorized access to sensitive user data.
- Risk Rating: High
- Evidence:



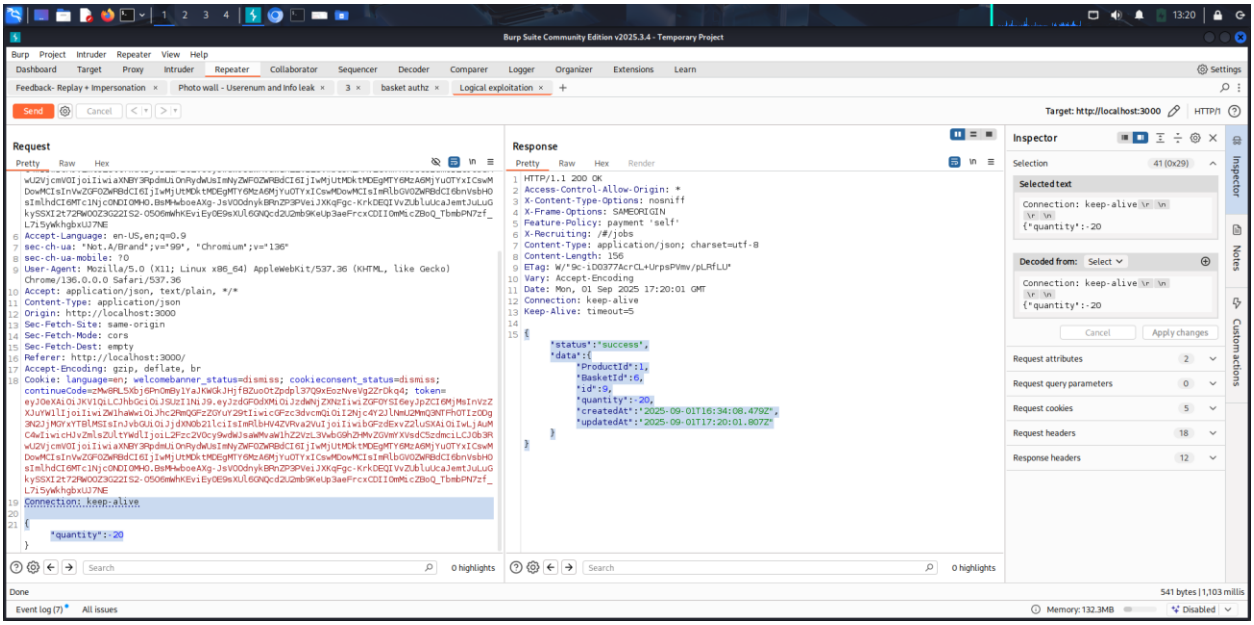


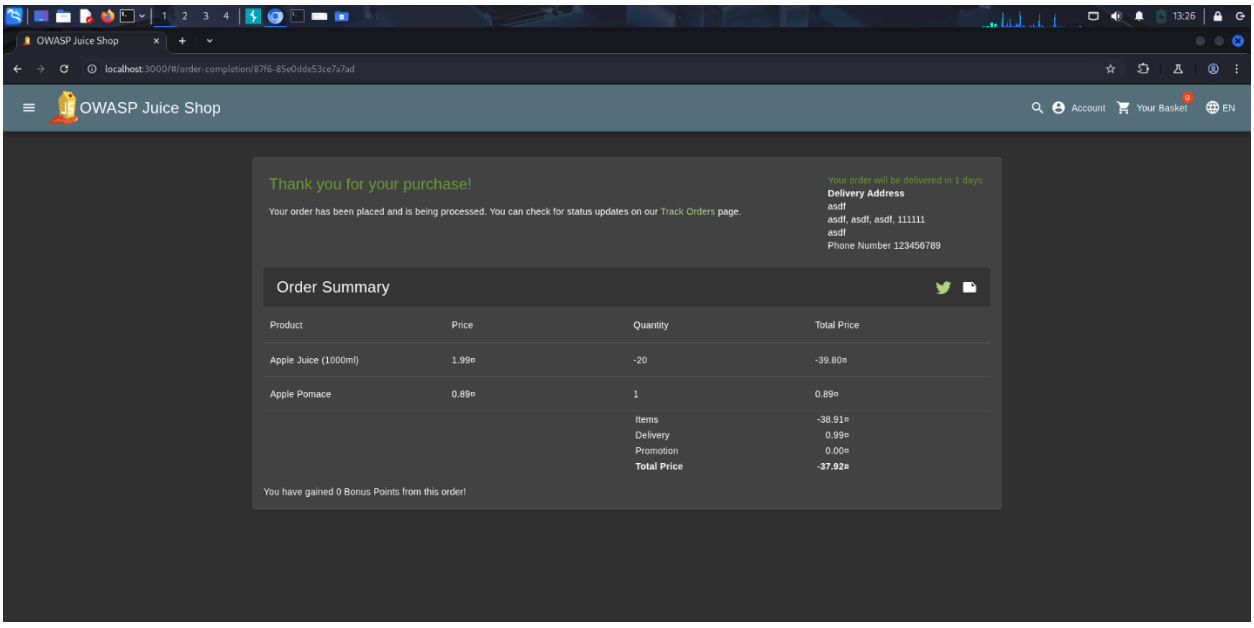
- **Mitigation Strategy:** Enforce strict authorization checks on every object access and avoid exposing predictable identifiers in requests.

6. Business Logic Exploitation – Payback Functionality

The payment/credit system in the application contains a logical flaw that can be exploited to generate a negative balance or illegitimate credit. By manipulating the refund process, attackers can credit their account beyond legitimate limits, effectively creating free funds.

- Impact: Enables unauthorized credit manipulation, leading to financial loss and abuse of the payment system.
- Risk Rating: Medium
- Evidence:





- **Mitigation Strategy:** Implement strict validation on refund and credit operations, enforce balance limits, and ensure transaction logic cannot be bypassed or abused.

OWASP Top 10 Mapping

Vulnerability	OWASP Top 10 (2021)	Risk Rating
SQL Injection in Login	A03:2021 – Injection	High
Insecure Token Design (JWT with MD5 credentials)	A02:2021 – Cryptographic Failures	High
Insecure Direct Object Reference (IDOR)	A01:2021 – Broken Access Control	High
Replay & Impersonation in Feedback	A01:2021 – Broken Access Control	Medium
User Enumeration & Information Leakage	A02:2021 – Cryptographic Failures / A05:2021 – Security Misconfiguration	Medium
Logical Exploitation – Payback Functionality	A04:2021 – Insecure Design	Medium

CONCLUSION

The assessment uncovered critical issues including injection flaws, broken access control, insecure token design, information leakage, and logic exploits. These vulnerabilities align with the OWASP Top 10 (2021) and expose risks such as account takeover, data compromise, and financial abuse. Strengthening access controls, applying secure coding practices, and using robust cryptographic standards are key to improving security posture.