Penetration Testing Report for OWASP Juice Shop Conducted Using Burp Suite:

1. Introduction

This report details the findings from a penetration testing exercise conducted on OWASP Juice Shop using Burp Suite. The test focused on identifying vulnerabilities, successfully exploiting them, and providing remediation strategies to mitigate risks.

2. Summary of Findings

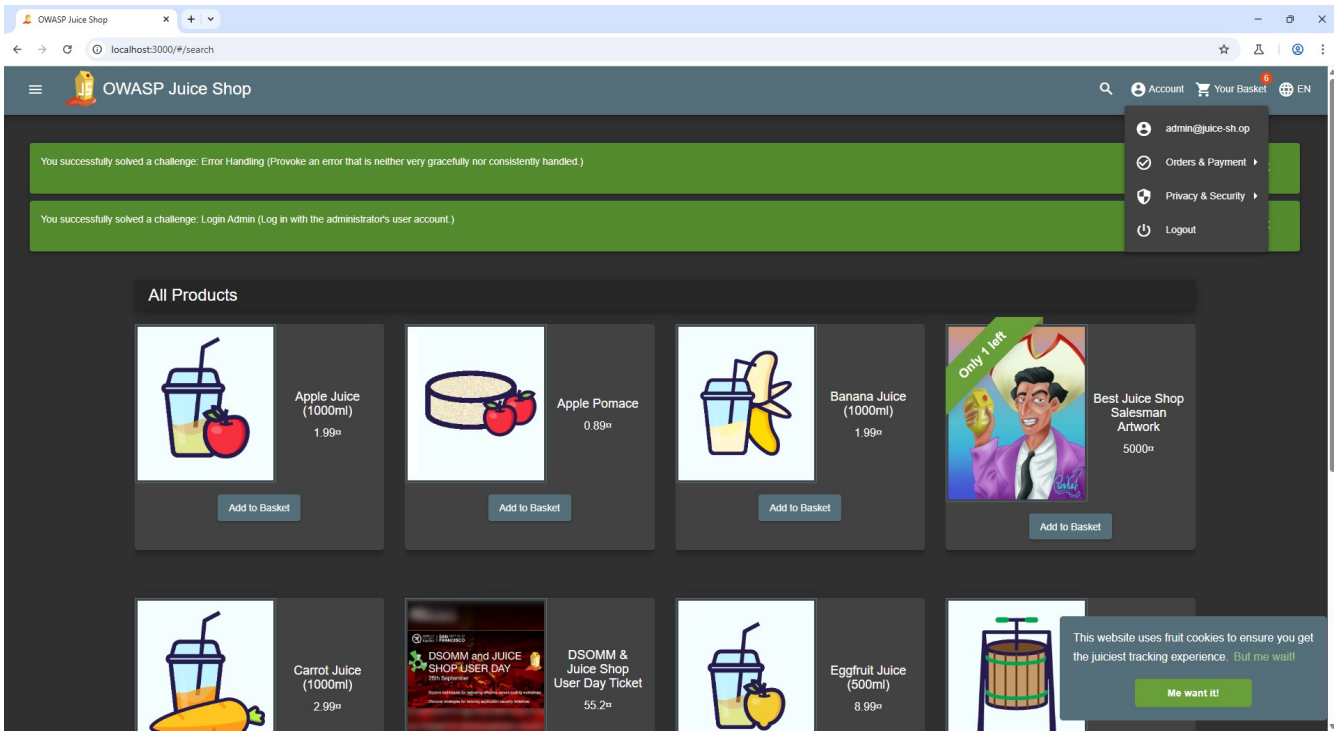
The following vulnerabilities were identified and successfully exploited:

I. **SQL Injection (SQLi) - Admin Login Bypass**
II. **Information Disclosure - Admin Email Address**
III. **Brute Force Attack - Credential Discovery**
IV. **Reflected Cross-Site Scripting (XSS) Attack**
V. **Session Hijacking via XSS - Cookie Theft**

3. Detailed Exploitation Steps and Impact

3.1 SQL Injection - Admin Login Bypass

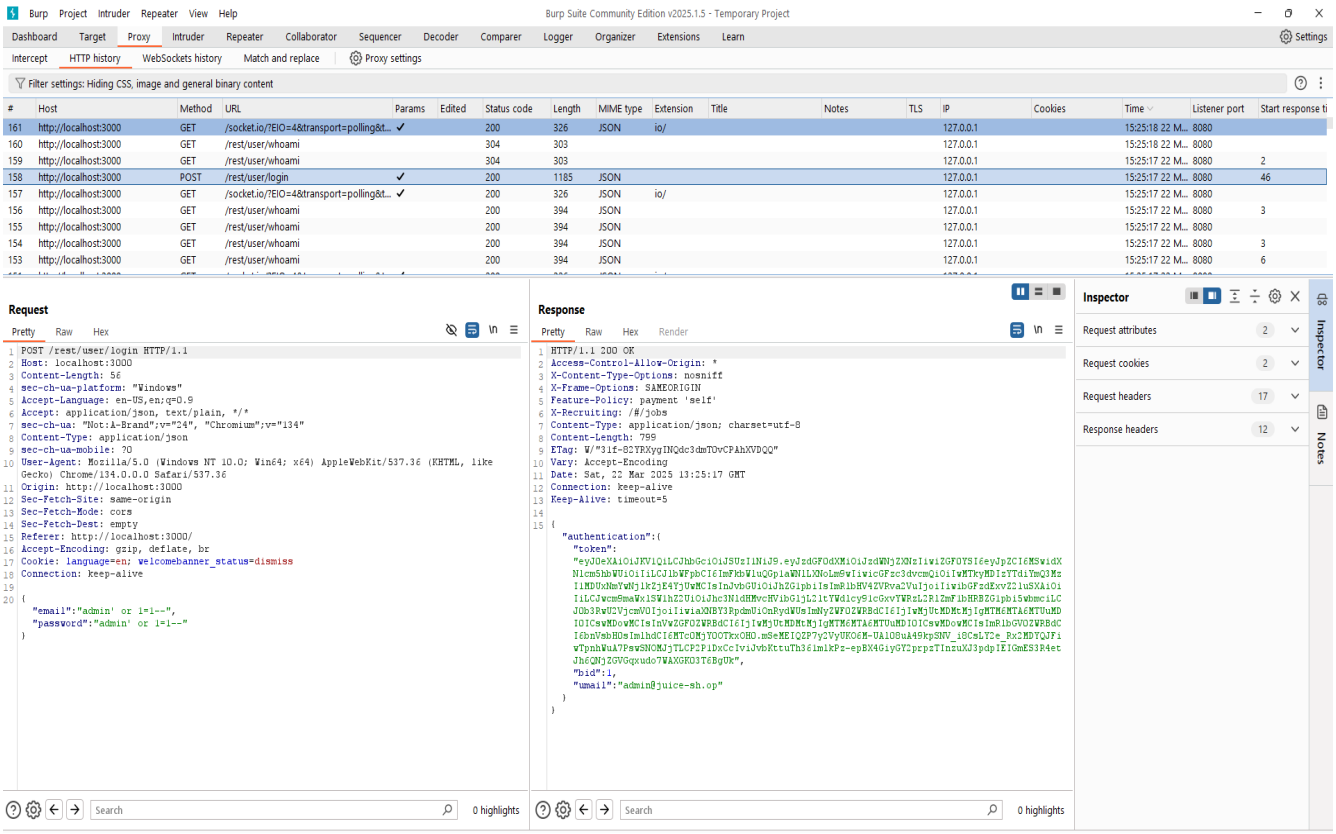
- **Description**: A SQL Injection vulnerability was identified in the login form, allowing access without valid credentials.
- **Exploitation Steps**:
    1. Used SQLi payload `' OR 1=1 --` in the username/password field.
    2. Successfully logged in as an admin.



- **Impact**: Unauthorized access to the admin account and potentially sensitive data.
- **Remediation**:
    1. Implement parameterized queries to prevent SQL injection.
    2. Use prepared statements and stored procedures.
    3. Implement input validation to filter malicious input.

3.2 Information Disclosure - Admin Email

- **Description**: The admin's email address was exposed through poorly secured API responses.
- **Exploitation Steps**:
    1. Intercepted HTTP requests using Burp Suite.
    2. Identified admin email in API response.

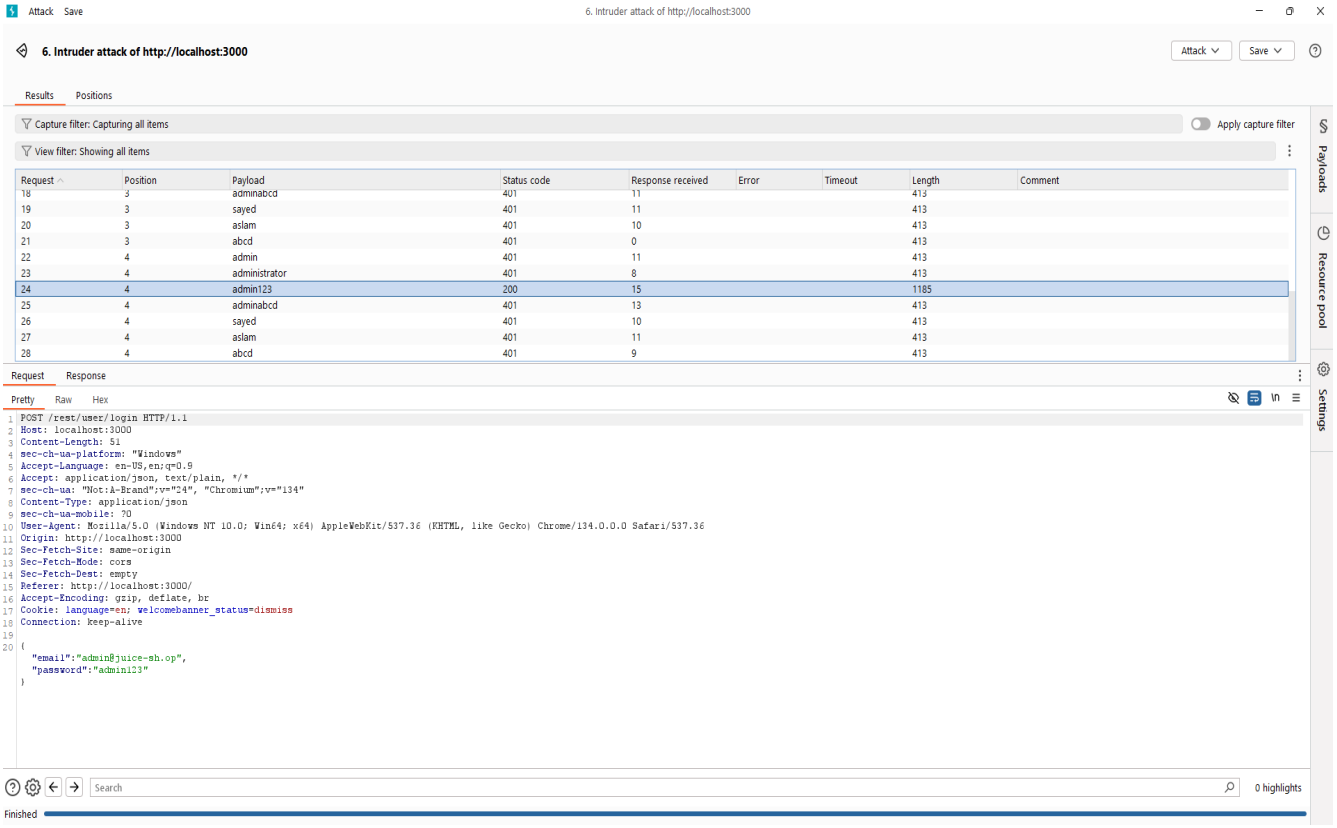


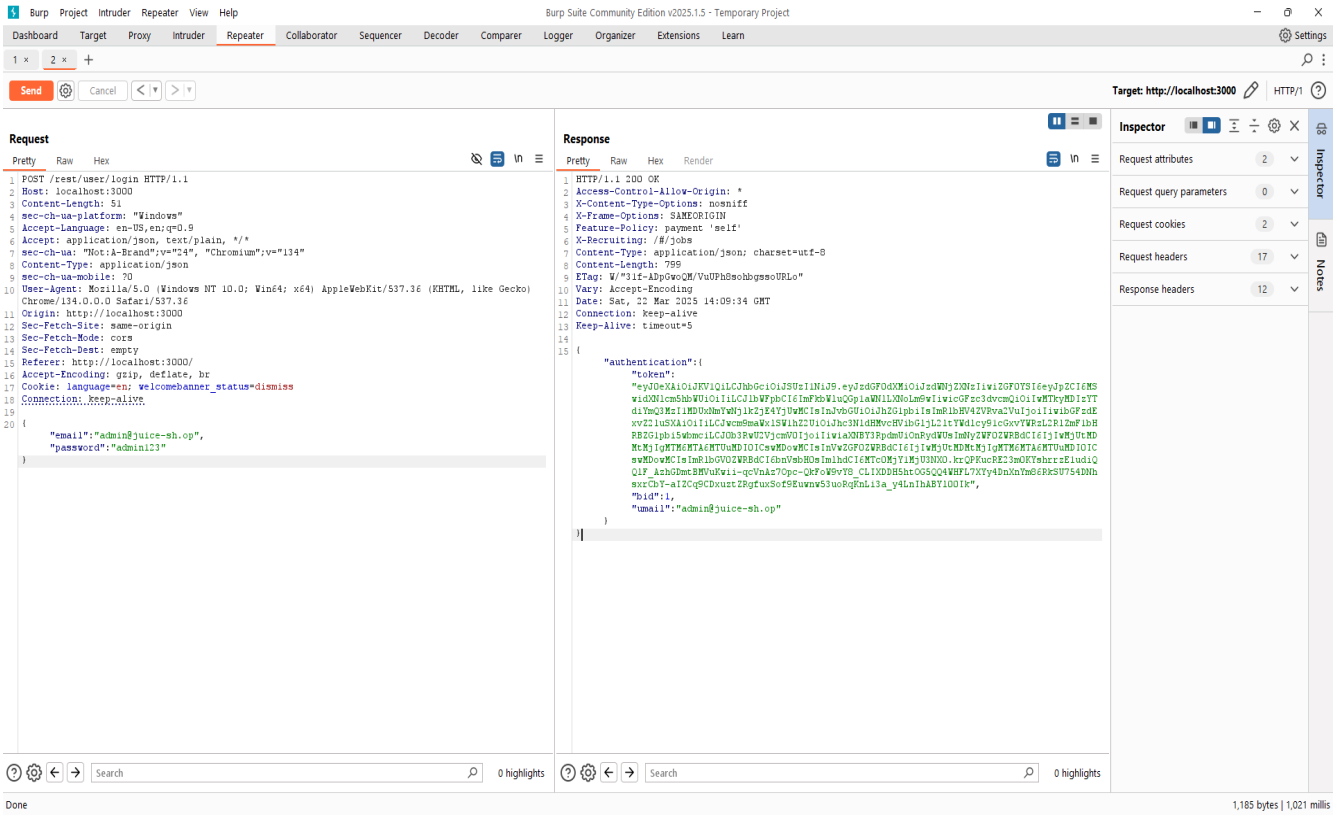
- **Impact**: Attackers can use the leaked email for phishing, brute force attacks, or social engineering.
- **Remediation**:
    1. Restrict sensitive data exposure in API responses.
    2. Implement role-based access control (RBAC).
    3. Encrypt sensitive information in storage and transit.

3.3 Brute Force Attack - Credential Discovery

- **Description**: The admin password was obtained through a brute-force attack.
- **Exploitation Steps**:
    1. Used Burp Suite's Intruder tool to automate login attempts.

2. Successfully identified the correct password.

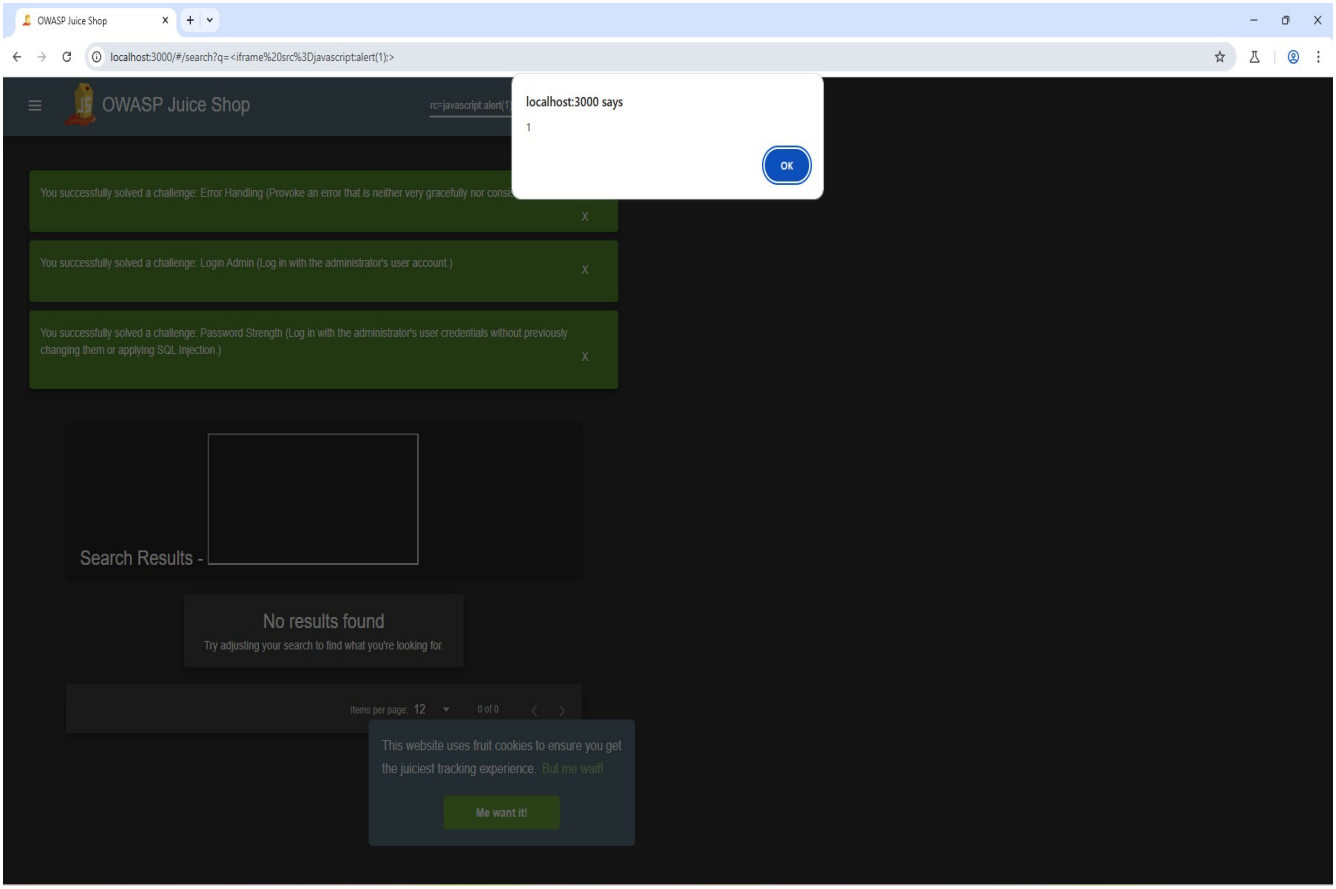




- **Impact**: Unauthorized access leading to data compromise.
- **Remediation**:
    1. Implement account lockout mechanisms after multiple failed attempts.
    2. Enforce strong password policies.
    3. Implement CAPTCHA on login forms.

3.4 Reflected Cross-Site Scripting (XSS)

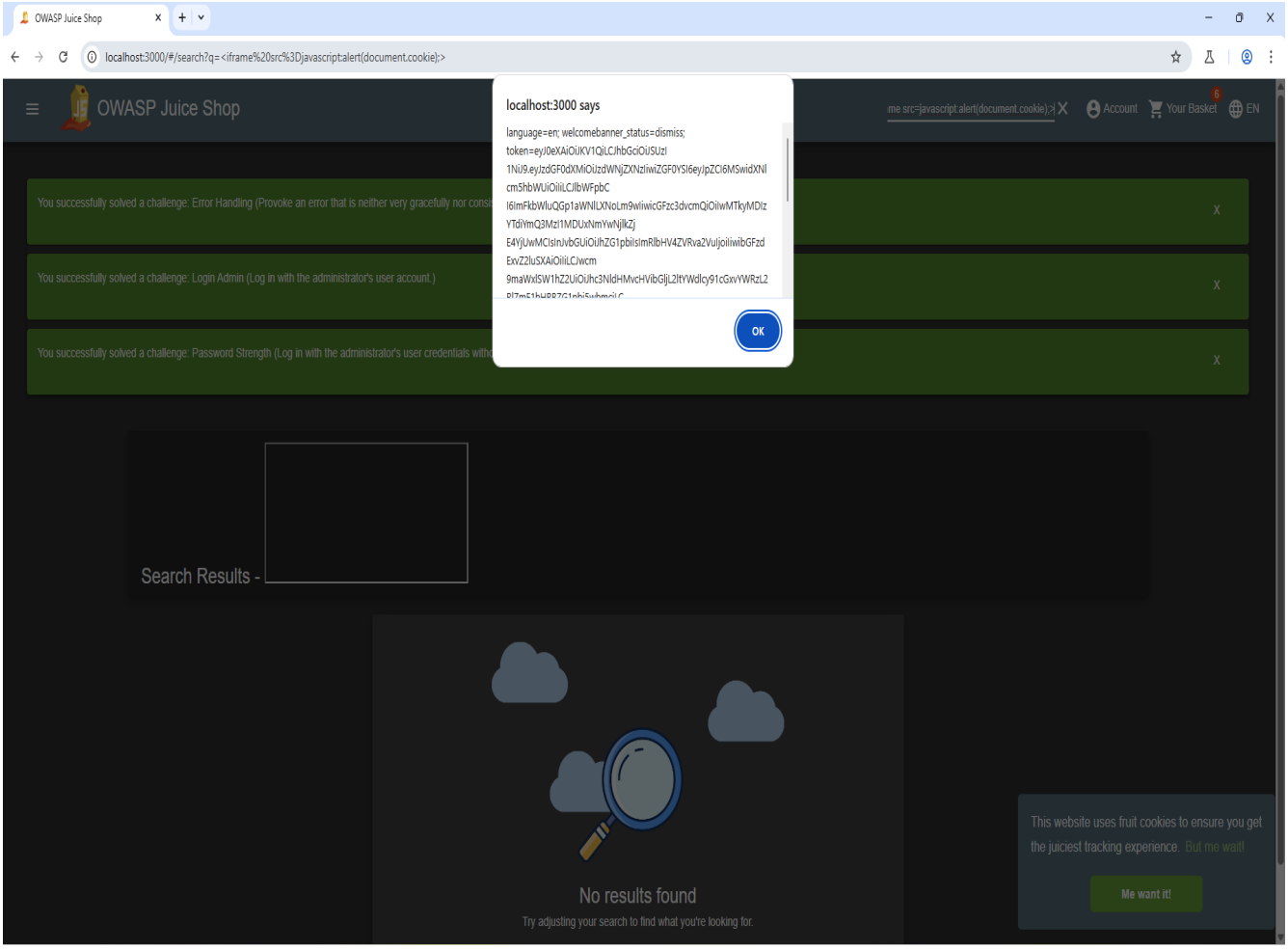
- **Description**: The application was vulnerable to reflected XSS, allowing an attacker to inject malicious JavaScript.
- **Exploitation Steps**:
    1. Injected `<iframe src=javasrcipt:alert(1);>` into a search input field.
    2. Observed script execution on the client-side.



- **Impact**: Could allow session hijacking, defacement, and phishing attacks.
- **Remediation**:
    1. Implement proper input validation and output encoding.
    2. Use Content Security Policy (CSP) to restrict script execution.
    3. Implement HTTP-only and secure cookies.

3.5 Session Hijacking via XSS - Cookie Theft

- **Description**: Using reflected XSS, the session cookie was extracted and used for session hijacking.
- **Exploitation Steps**:
    1. Injected `<iframe src=javasrcipt:alert(document.cookie);>` to extract document cookies.



- **Impact**: Complete takeover of user accounts.
- **Remediation**:
    1. Implement secure and HTTP-only cookie attributes.
    2. Use session expiration and token-based authentication.
    3. Implement Web Application Firewall (WAF) to detect and block XSS attacks.

4. Conclusion

The penetration test revealed critical security flaws in OWASP Juice Shop that could be exploited by attackers to gain unauthorized access and compromise user data. The identified vulnerabilities should be addressed by implementing secure coding practices, strengthening authentication mechanisms, and enforcing input validation techniques. Last but not the least, continuous security assessments and monitoring are recommended to mitigate future threats.