SECURITY ASSESSMENT REPORT — TASK 1

Internship: Cyber Security Internship at Future Interns
Intern Name: Shahana Parveen
CIN ID: FIT/SEP25/CS4100
Date: 01 October 2025

1. Introduction

This report documents the results of Task 1 of my Cyber Security Internship at Future Interns. The goal was to perform a **Web Application Security Assessment** of a deliberately vulnerable application (OWASP Juice Shop), identify common security issues, and document them according to OWASP standards.

The assessment was performed using industry tools such as **OWASP ZAP** and **Burp Suite** to simulate real-world penetration testing scenarios.

2. Methodology

The following methodology was followed during the assessment:

- **Setup:** Deployed OWASP Juice Shop (deliberately vulnerable web app).
- **Scanning:** Performed an automated vulnerability scan using **OWASP ZAP**.
- **Manual Testing:** Intercepted traffic with **Burp Suite** and tested parameters manually.
- **Vulnerabilities Tested:** SQL Injection, Cross-Site Scripting (XSS), and other OWASP Top 10 flaws.
- **Documentation:** Collected logs, screenshots, and wrote findings in this report.

3. Tools Used

- OWASP ZAP (automated vulnerability scanner)
- Burp Suite (manual web testing toolkit)
- SQLMap (optional SQLi testing)
- OWASP Juice Shop (test environment)
- Windows 11 + Firefox (testing environment)

4. Findings

4.1 SQL Injection (Reflected) — /complete/search

- **Description:** The term search parameter in /complete/search is not properly sanitized. Sending the payload ' OR '1'='1 altered the server response, indicating a SQL injection vulnerability.
- **Evidence:**
    - Request: GET /complete/search?client=firefox&term=%27%20OR%20%271%27%3D%271
    - Screenshot: 04_sql_injection.png
- **Impact:** SQL Injection can allow attackers to manipulate backend queries, potentially exposing or modifying sensitive data.
- **Recommendation:**
    - Use **parameterized queries / prepared statements**
    - Validate and sanitize inputs server-side
    - Apply least-privilege to database accounts

4.2 XSS Check — /complete/search

- **Description:** Tested the term parameter with <script>alert(1)</script>. No reflected XSS was observed (payload not reflected in response).
- **Evidence:** Empty/unchanged response (no alert).
- **Impact:** No exploitable XSS found in this test case.
- **Recommendation:** Continue applying input/output sanitization to prevent XSS in other areas.

5. Conclusion

The assessment successfully demonstrated how web applications can be tested for vulnerabilities using OWASP ZAP and Burp Suite.

- A SQL Injection vulnerability was identified in the search parameter.
- No reflected XSS was observed in the tested functionality.

Through this exercise, I gained hands-on experience in vulnerability scanning, manual penetration testing, and documenting findings in a professional security report.

6. Appendix

- **Screenshots Folder:** Contains supporting screenshots (01_app_home.png, 02_zap_findings.png, 03_burp_request.png, 04_sql_injection.png).
- **Logs Folder:** Contains automated scan report (zap_report.pdf).