Cybersecurity Vulnerability Assessment Report

Target: OWASP Juice Shop (Test Environment)
Tool Used: Burp Suite Professional
Submitted By: Gourav Swaroop
Date: 23/08/2025

Executive Summary

A penetration test was conducted on the OWASP Juice Shop application within a controlled lab environment to identify vulnerabilities aligned with the OWASP Top 10 (2021).

The assessment revealed four critical vulnerabilities:

- **SQL Injection (SQLi) – High**
- **Cross-Site Scripting (XSS) – High**
- **Cross-Site Request Forgery (CSRF) – High**
- **Security Misconfiguration – Medium**

These vulnerabilities demonstrate typical web application weaknesses that could allow attackers to bypass authentication, execute malicious scripts, or manipulate user sessions. Screenshots demonstrating PoCs are available in the project repository.

Methodology

The assessment followed the OWASP Testing Guide:

1. **Environment Setup**
    - OWASP Juice Shop deployed locally (http://127.0.0.1:3000).
    - Burp Suite Professional configured as proxy (127.0.0.1:8080).
2. **Reconnaissance & Scoping**
    - Target scope defined in Burp Suite.
    - Manual exploration of input vectors (login forms, search bar, feedback forms).
3. **Automated Scanning**
    - Performed Crawl + Audit (Balanced Mode) in Burp Suite.
    - Filtered results for SQLi, XSS, CSRF, and misconfigurations.
4. **Manual Verification & Exploitation**
    - Verified vulnerabilities using Burp Repeater, Proxy, and browser tests.

- Created proof-of-concepts (PoCs) for XSS and CSRF.

5. **Documentation**
    - Mapped findings to OWASP Top 10 (2021) categories.
    - Captured screenshots as evidence.
    - Provided mitigation recommendations.

Findings

1. SQL Injection (SQLi)

- **Description:** Login form did not sanitize inputs, allowing SQL payloads to bypass authentication.
- **PoC:** {"email":'" OR '1'='1'--","password":"admin"} → 200 OK with valid authentication token
- **Screenshot:** Figure 1 – Finding-SQLi.png
- **Impact:** Bypass login, access admin accounts, or extract sensitive data.
- **Risk:** High
- **Mitigation:** Parameterized queries, server-side validation, WAF.

---

2. Cross-Site Scripting (XSS)

- **Description:** Search functionality failed to sanitize inputs, allowing JavaScript injection.
- **PoC:** <script>alert(1)</script> → Browser alert
- **Screenshot:** Figure 2 – Finding-XSS.png
- **Impact:** Steal session cookies, deface content, execute scripts.
- **Risk:** High
- **Mitigation:** Output encoding, DOMPurify sanitization, strict CSP.

---

3. Cross-Site Request Forgery (CSRF)

- **Description:** State-changing requests lack anti-CSRF tokens; relies solely on cookies.
- **PoC:** Malicious HTML form successfully changed account state without re-authentication.

- **Screenshot:** Figure 3 – Finding-CSRF.png
- **Impact:** Authenticated users may perform unwanted actions (e.g., change password).
- **Risk:** High
- **Mitigation:** Anti-CSRF tokens, SameSite cookies, validate Referer/Origin headers.

---

4. Security Misconfiguration

- **Description:** Missing security headers; application accessible via HTTP.
- **PoC:** Missing headers: Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options
- **Screenshot:** Figure 4 – Finding-SecurityMisconfig.png
- **Impact:** Exposed to clickjacking, MIME sniffing, MITM attacks.
- **Risk:** Medium
- **Mitigation:** Enable HTTPS/HSTS, configure headers (X-Frame-Options: DENY, X-Content-Type-Options: nosniff, strict CSP), harden server.

OWASP Top 10 Mapping

Vulnerability	OWASP Top 10 Category (2021)	Risk Level
SQL Injection	A03: Injection	High
Cross-Site Scripting (XSS)	A07: XSS	High
Cross-Site Request Forgery	A08: Software & Data Integrity Failures	High
Security Misconfiguration	A05: Security Misconfiguration	Medium

---

Conclusion

The penetration test identified multiple critical vulnerabilities that could compromise the application. Exploiting these weaknesses may allow attackers to bypass authentication, inject malicious scripts, or manipulate user sessions. Implementing the recommended mitigations—including input validation, anti-CSRF measures, strict headers, and secure configurations—will significantly improve the application's resilience against OWASP Top 10 threats.

Screenshots of PoCs for each vulnerability are available in the project repository for reference.