Web Application Security Assessment Report

Target Application: OWASP Juice Shop

Assessment Type: Web Application Penetration Test / Secure Code Review

Assessor: Sayada Mehnaaz Hussaini

Date: August 22, 2025

1. Executive Summary

The OWASP Juice Shop application was tested against the OWASP Top 10 Web Application Security Risks. Multiple critical vulnerabilities were identified, including SQL Injection, Cross-Site Scripting (XSS), and Insecure Direct Object References (IDOR). If exploited in a real production system, these vulnerabilities could lead to data leakage, account takeover, or unauthorized system access.

2. Scope of Testing

- Target: OWASP Juice Shop (local deployment on Docker)
- Methodology: OWASP Testing Guide + Manual Penetration Testing
- Focus: OWASP Top 10 (2021 edition)

3. Findings Summary

ID	Vulnerability	Severity	Impact	Status
001	SQL Injection on Login Form	Critical	Database dump, credential theft	Open
002	Stored XSS in Product Review	High	Session hijacking, account takeover	Open
003	IDOR on User Profile Endpoint	High	Unauthorized access to other user data	Open
004	Weak Password Policy	Medium	Increases brute force risk	Open

4. Detailed Findings

4.1 SQL Injection (Critical)

Description: The login form fails to sanitize user input. By injecting `admin' OR '1'='1` into the username field, authentication is bypassed.

Impact: Full access to admin account. Possible extraction of user credentials from the database.

Proof of Concept (PoC):

```
POST /rest/user/login
{
  "email": "admin' OR '1'='1",
  "password": "anything"
}
```

Recommendation: Use parameterized queries / prepared statements. Implement input validation & sanitization. Enable WAF rules for SQL injection patterns.

4.2 Stored Cross-Site Scripting (High)

Description: User input is stored and rendered without proper escaping. Attacker can inject malicious JavaScript.

Impact: Steal user cookies/session tokens. Redirect victims to phishing pages.

Proof of Concept (PoC):

```
<script>alert('XSS')</script>
```

Recommendation: Apply output encoding before rendering user input. Implement Content Security Policy (CSP). Use libraries that auto-sanitize inputs.

4.3 Insecure Direct Object Reference (High)

Description: Changing the user ID in the request returns data of other users without authorization checks.

Impact: Unauthorized access to sensitive user details. Privacy violations and GDPR concerns.

Proof of Concept (PoC):

GET /rest/user/2 → Returns details of another user

Recommendation: Implement access control checks on server side. Never trust client-supplied identifiers. Use indirect references.

4.4 Weak Password Policy (Medium)

Description: The application allows weak passwords like '12345' and 'password'.

Impact: Increases risk of credential stuffing and brute-force attacks.

Recommendation: Enforce strong password complexity rules. Implement rate limiting / account lockout. Encourage MFA.

---

5. Conclusion

The Juice Shop application demonstrates common but critical vulnerabilities that mirror real-world risks. Mitigation should prioritize SQL Injection, XSS, and IDOR, as they could lead to account takeover, data breaches, and compliance violations.

Next Steps:

- Fix identified vulnerabilities in code
- Re-test after patching
- Integrate secure coding practices and automated security testing into the SDLC