Web Application Penetration Testing Report

Project Title: OWASP Juice Shop Web Application Vulnerability Assessment and Penetration Testing Report
Target Application: OWASP Juice Shop
Tester: Yuvaraj S
Date: February 2026

1. Abstract

This project presents a security assessment of the OWASP Juice Shop web application to identify and analyze critical vulnerabilities based on the OWASP Top 10:2025 framework. Multiple vulnerabilities including Broken Access Control, Security Misconfiguration, Cross-Site Scripting (XSS), and SQL Injection were identified and exploited to evaluate their impact and propose remediation strategies. The assessment demonstrates common real-world web security weaknesses and highlights secure coding and configuration practices.

2. Objective

The objective of this project is to:

- Identify vulnerabilities in a web application
- Demonstrate exploitation techniques
- Analyze security risks and impacts
- Recommend mitigation strategies

3. Scope of Testing

- Target Application: OWASP Juice Shop
- Environment: Localhost (http://localhost:3000)
- Testing Type: Black-box and Gray-box web application testing

4. Methodology

The assessment followed a structured penetration testing methodology:

1. Reconnaissance & Enumeration
2. Vulnerability Identification
3. Exploitation (Proof of Concept)
4. Impact Analysis
5. Remediation Recommendations

5. Tools Used

- Kali Linux
- Burp Suite
- Firefox Developer Tools
- OWASP Juice Shop
- Manual HTTP Request Manipulation

6. Findings Summary

ID	OWASP Category	Vulnerability	Severity
A01	Broken Access Control	IDOR (Horizontal Privilege Escalation)	High
A01	Broken Access Control	Vertical Privilege Escalation (Admin Panel Access)	Critical
A02	Security Misconfiguration	Directory Listing & Sensitive File Exposure	High
A03	Injection	Reflected Cross-Site Scripting (XSS)	High
A05	Injection	SQL Injection (Authentication Bypass)	Critical

7. Detailed Findings

7.1 A01:2025 – Broken Access Control (IDOR)

Technical Summary

The application uses user-supplied IDs in URLs to fetch profile data without validating authorization, allowing users to access other users' private information.

Proof of Concept

http://localhost:3000/#/user/profile?id=6

Change to:

http://localhost:3000/#/user/profile?id=1

Impact

- Exposure of emails and user data
- User enumeration and privacy violation

Remediation

- Enforce server-side authorization
- Use session-based identity instead of URL parameters

7.2 A01:2025 – Broken Access Control (Vertical Privilege Escalation)

Technical Summary

Administrative endpoints are not protected by server-side role checks.
UI hiding is used instead of backend authorization.

Proof of Concept

http://localhost:3000/#/administration

Impact

- Full access to user database
- Ability to delete feedback and modify data

Remediation

- Implement Role-Based Access Control (RBAC)
- Enforce deny-by-default policy

7.3 A02:2025 – Security Misconfiguration (Directory Listing)

ETechnical Summary
The /ftp directory allows directory listing and exposes sensitive backup files.
Proof of Concept
http://localhost:3000/ftp
Exposed Files
- package.json.bak
- coupons_2023.json.bak
- acquisition.md

Impact
- Disclosure of dependencies and internal configuration
- Possible credential leakage

Remediation
- Disable directory indexing
- Remove backup files from web root
- Use allow-list file handling

7.4 A03:2025 – Cross-Site Scripting (Reflected XSS)

Technical Summary
User input in the search bar is reflected without sanitization, allowing JavaScript execution.
Payload
<iframe src="javascript:alert(1)"></iframe>
Impact
- Session hijacking via document.cookie
- Phishing via malicious iframes
- Unauthorized actions in user context

Remediation
- Context-aware output encoding
- Implement Content Security Policy (CSP)
- Use .textContent instead of .innerHTML

7.5 A05:2025 – SQL Injection (Authentication Bypass)

Technical Summary

The login form concatenates user input directly into SQL queries, allowing injection.

Payload

' OR 1=1 --

Impact

- Admin account takeover
- Full database extraction via UNION queries
- Privilege escalation

Remediation

- Use prepared statements
- Implement ORM frameworks
- Validate user input
- Enforce least-privilege DB accounts

8. Risk Matrix

Vulnerability	Confidentiality	Integrity	Availability	Overall Risk
IDOR	High	Low	Low	High
Vertical Access Control	Critical	High	Medium	Critical
Security Misconfiguration	High	Medium	Low	High
XSS	High	High	Low	High
SQL Injection	Critical	Critical	Medium	Critical

9. Conclusion

The OWASP Juice Shop application contains multiple critical security vulnerabilities that can lead to full system compromise, data leakage, and unauthorized administrative control. The findings highlight the importance of implementing strong access control mechanisms, secure coding practices, and hardened server configurations. Immediate remediation is recommended to mitigate real-world exploitation risks.

10. References

- OWASP Top 10:2025
- OWASP Testing Guide
- NIST SP 800-53 Security Controls