WEB APPLICATION SECURITY ASSESSMENT REPORT

OWASP Juice Shop Application
Penetration Testing Assessment

Date: December 1, 2025 Assessed by: Security Analyst
Target Application: https://juice-shop.herokuapp.com

EXECUTIVE SUMMARY

This report presents the findings of a comprehensive security assessment conducted on the OWASP Juice Shop web application. The assessment identified FIVE (5) CRITICAL vulnerabilities that pose significant security risks.

OVERALL RISK RATING: CRITICAL

Key Findings: • SQL Injection allowing authentication bypass • Cross-Site Scripting (XSS) vulnerability • Broken Access Control exposing admin panel • Sensitive Data Exposure through FTP directory • Security Misconfiguration revealing error details

Immediate remediation is strongly recommended.

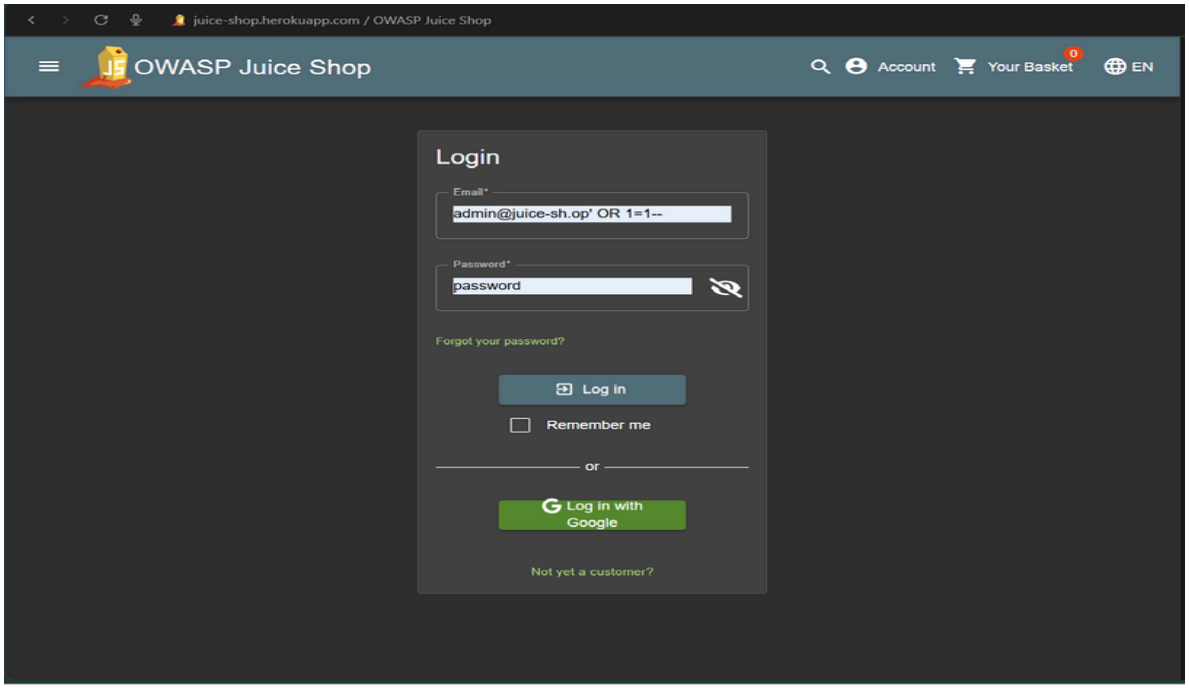
VULNERABILITY FINDINGS

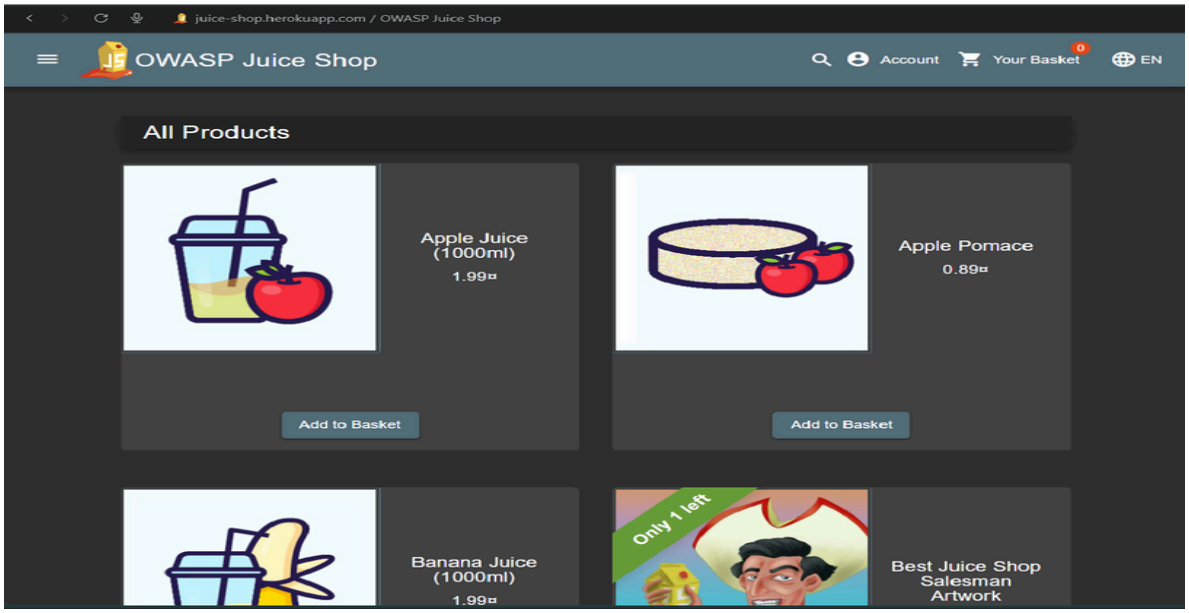
[VULNERABILITY #1] SQL INJECTION - AUTHENTICATION BYPASS
Severity: CRITICAL 🔴 OWASP Category: A03:2021 - Injection CVSS Score: 9.8

Description: The login form is vulnerable to SQL injection attacks. An attacker can bypass authentication by injecting SQL code into the email field.

Proof of Concept: •URL: https://juice-shop.herokuapp.com/#/login

Screenshot Evidence:





•Payload: admin@juice-sh.op' OR 1=1--
•Password: (any value) •Result: Successfully logged in as admin user

Impact: •Complete authentication bypass
•Unauthorized access to admin account
•Potential data theft •Database manipulation possible

Remediation: 1.Use parameterized queries/prepared statements 2.Implement input validation and sanitization 3.Use ORM frameworks (e.g., Sequelize, TypeORM) 4.Apply principle of least privilege to database accounts

Code Fix Example: //VULNERABLE CODE const query = `SELECT * FROM Users WHERE email='${email}' AND password='${password}'`;

// SECURE CODE const query = 'SELECT * FROM Users WHERE email=? AND password=?'; db.execute(query, [email, hashedPassword]);

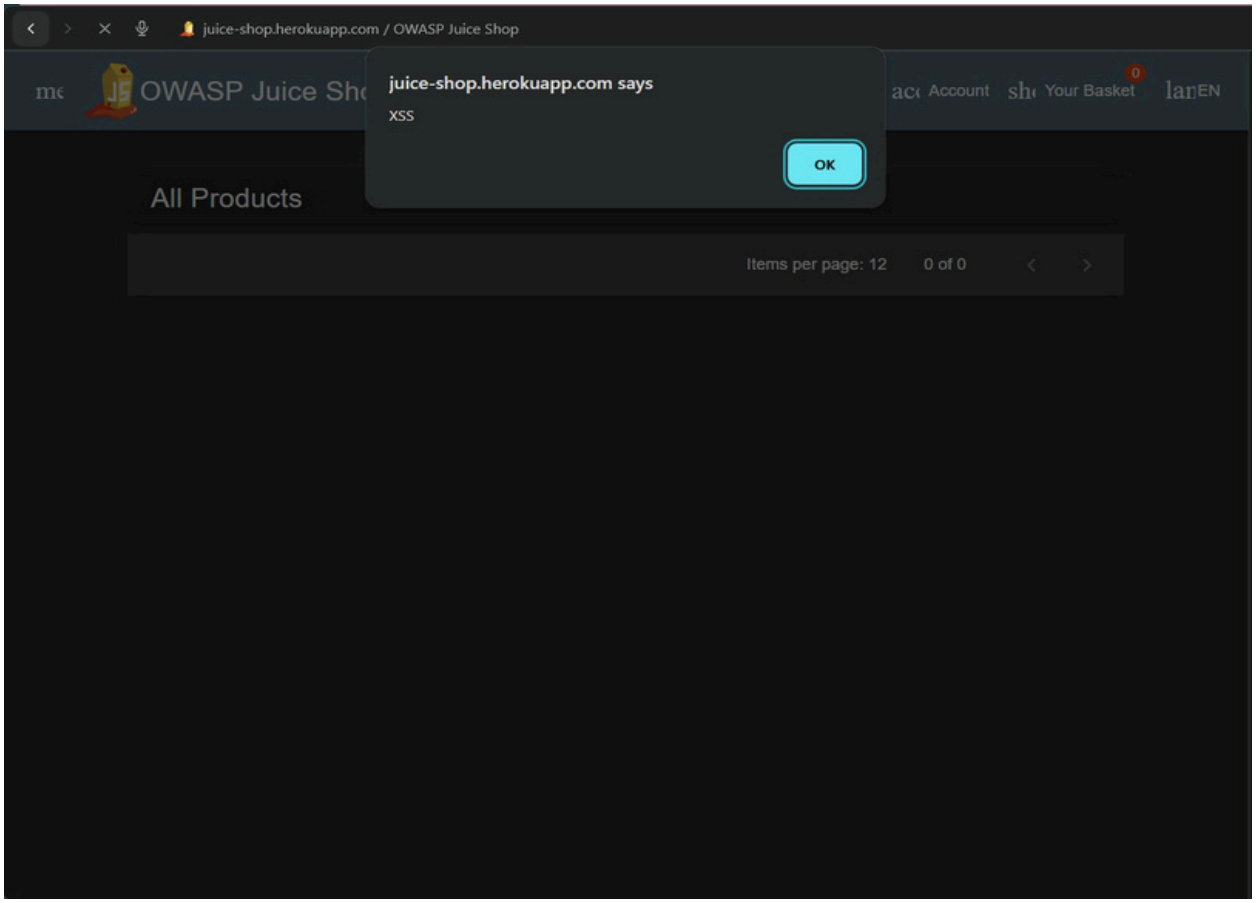
[VULNERABILITY #2] CROSS-SITE SCRIPTING (XSS)
Severity: HIGH 🟠 OWASP Category: A03:2021 - Injection CVSS Score: 7.3

Description: The search functionality reflects user input without proper sanitization, allowing execution of malicious JavaScript code.

Proof of Concept: • URL: https://juice-shop.herokuapp.com/#/search

• Payload: <iframe src="javascript:alert('XSS')"> • Result: XSS payload executed and displayed in search results

ScreenShot Evidence:



Impact: •Session hijacking through cookie theft •Credential stealing via fake login forms •Malicious redirects •Defacement of web pages

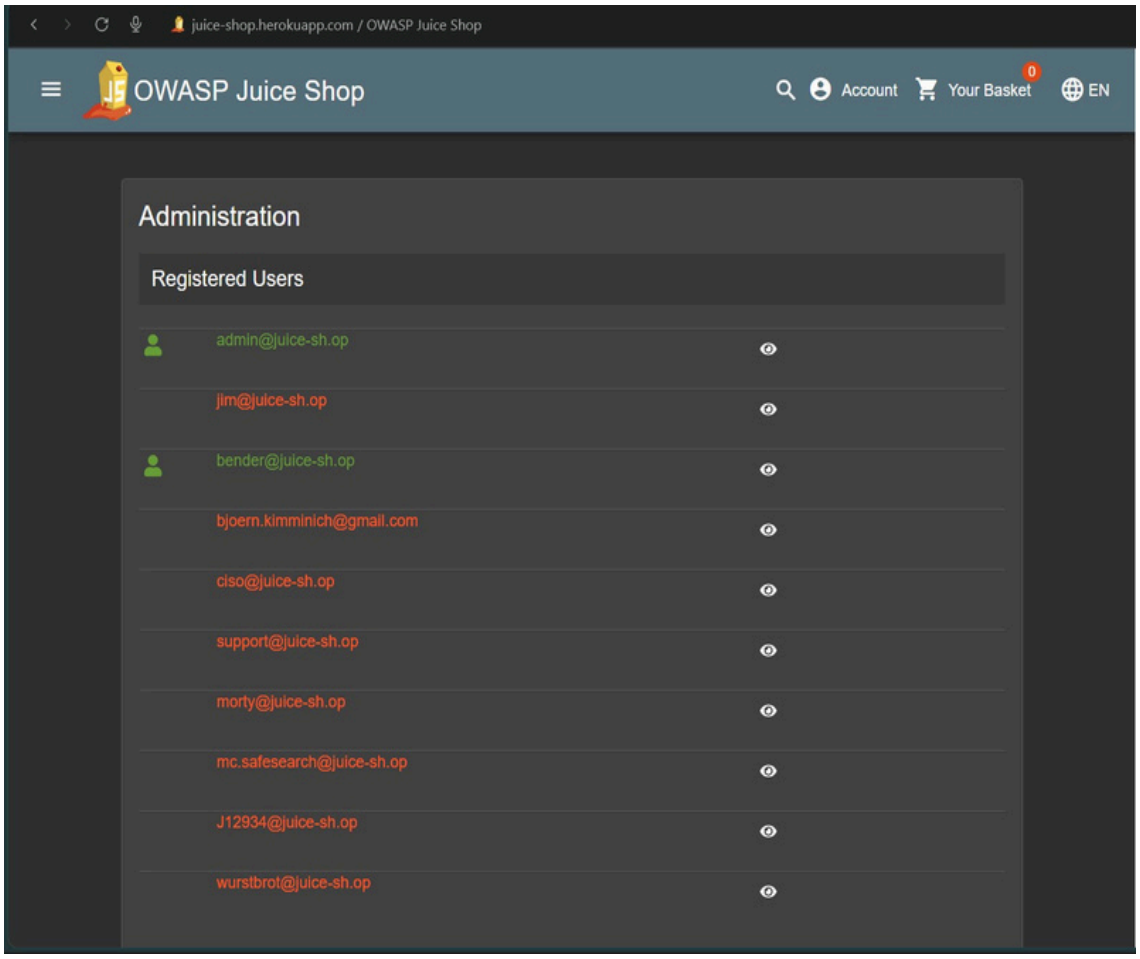
Remediation: 1.Implement output encoding/escaping 2.Use Content Security Policy (CSP) headers 3.Sanitize user input on both client and server side 4.Use security libraries (DOMPurify, OWASP Java Encoder)

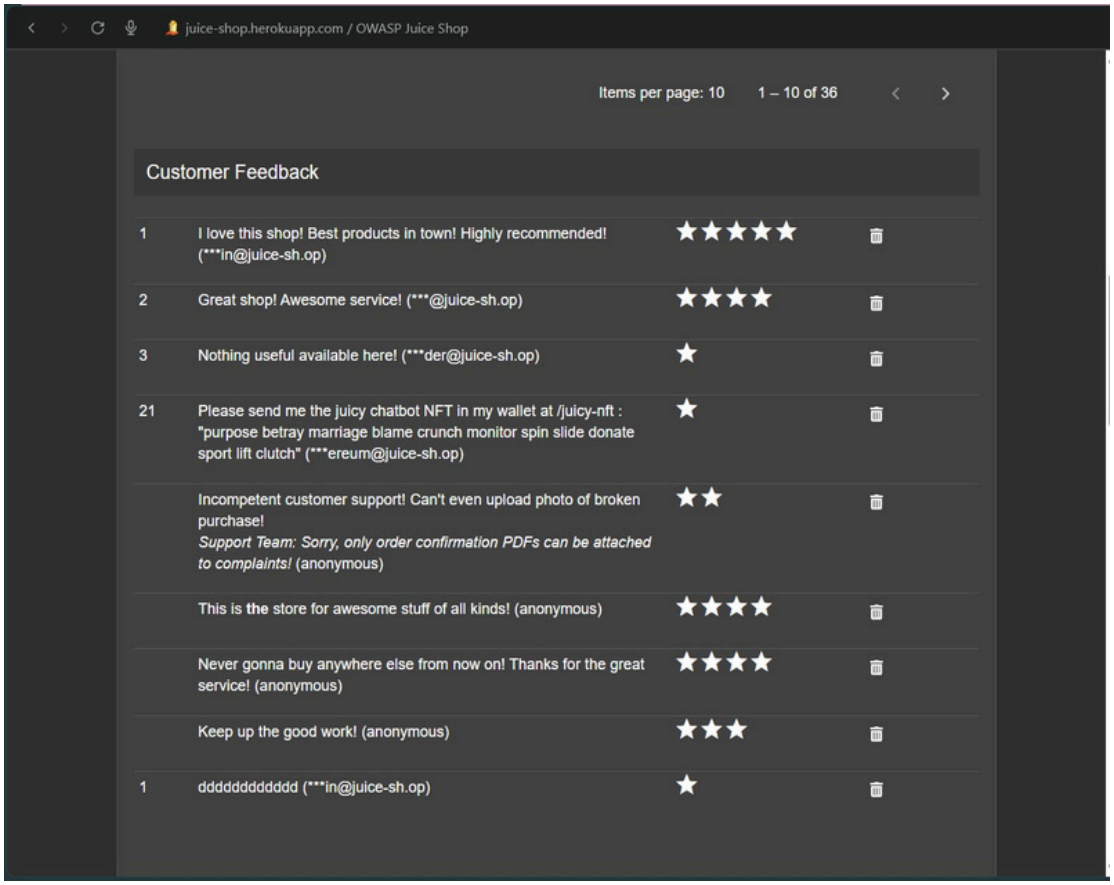
[VULNERABILITY #3] BROKEN ACCESS CONTROL
Severity: CRITICAL 🔴 OWASP Category: A01:2021 - Broken Access Control CVSS Score: 8.8

Description: The administration panel is accessible without proper authorization checks. Any authenticated user can access sensitive admin functions.

Proof of Concept: • URL: https://juice-shop.herokuapp.com/#/administration • Steps: Login as any user, navigate to /administration • Result: Full access to user database and admin panel • Exposed Data: All user emails, account details

ScreenShot Evidence:





Impact: •Unauthorized access to admin functionality •Exposure of all user data •Privilege escalation •Data breach potential

Remediation: 1.Implement role-based access control (RBAC) 2.Verify user permissions on server-side 3.Use authentication middleware 4.Apply least privilege principle 5.Log all access attempts to sensitive resources

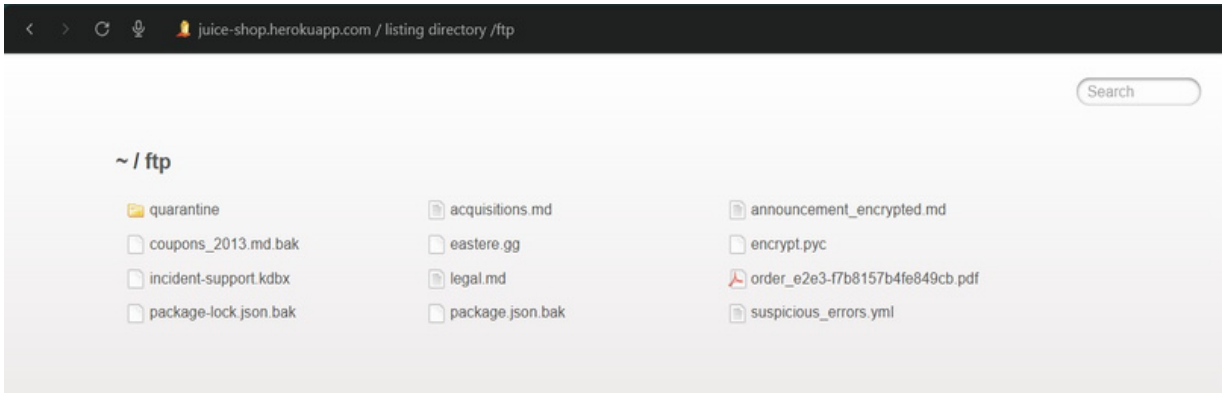
[VULNERABILITY #4] SENSITIVE DATA EXPOSURE
Severity: HIGH 🟠 OWASP Category: A02:2021 - Cryptographic Failures CVSS Score: 7.5

Description: The /ftp directory is publicly accessible without authentication, exposing confidential business documents and backup files.

Proof of Concept: • URL: https://juice-shop.herokuapp.com/ftp • Exposed Files:

- acquisitions.md (confidential business plans) - coupons_2013.md.bak (backup with potential credentials) - package.json.bak (application configuration) - legal.md (legal documents) - suspicious_errors.yml (system information)

ScreenShot Evidence:



Impact: • Exposure of confidential business information • Potential credential leakage in backup files • System architecture disclosure • Compliance violations (GDPR, PCI-DSS)

Remediation: 1. Implement authentication for all file directories 2. Remove backup files from production servers 3. Use .htaccess or web server config to deny directory listing 4. Store sensitive files outside web root 5. Encrypt sensitive data at rest

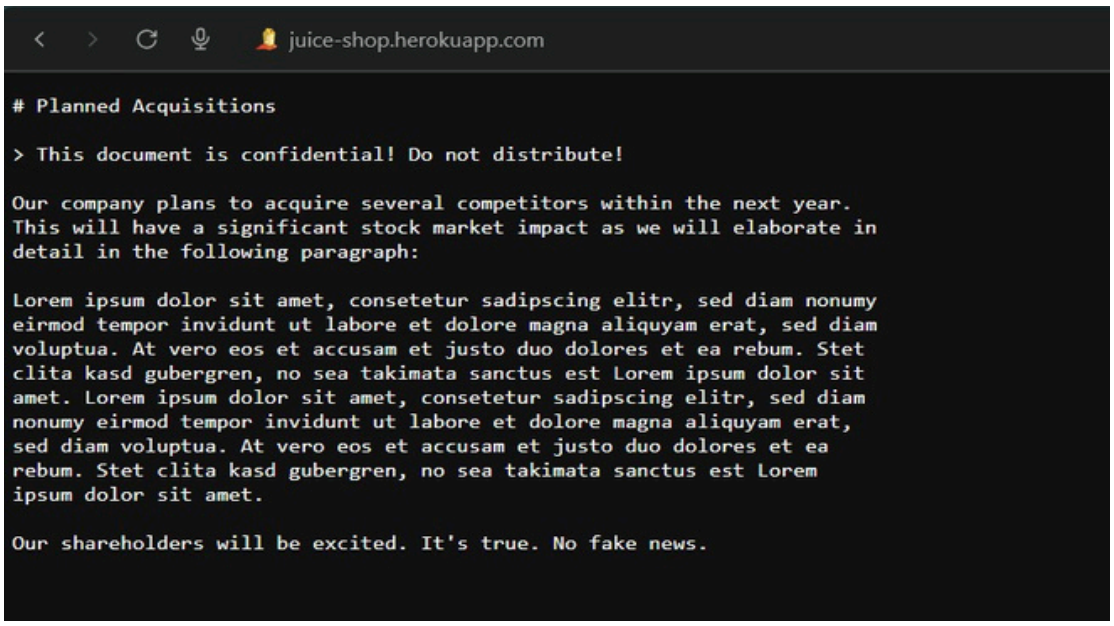
[VULNERABILITY #5] SECURITY MISCONFIGURATION - ERROR DISCLOSURE Severity: MEDIUM 🟡 OWASP Category: A05:2021 - Security Misconfiguration CVSS Score: 5.3

Description: Detailed error messages and stack traces are exposed to users, revealing internal application structure and technology stack.

Proof of Concept: • URL: https://juice-shop.herokuapp.com/ftp/package.json.bak • Error Response: Full Node.js/Express stack trace • Revealed Information:

- Express version (4.21.0) - File system paths - Internal function names - Router implementation details

ScreenShot Evidence:



Impact: •Information disclosure aiding targeted attacks •Technology stack fingerprinting •Easier identification of known vulnerabilities •Path traversal attack facilitation

Remediation: 1. Implement custom error pages 2. Disable debug mode in production 3. Configure error logging to files, not browser 4. Use generic error messages for users 5. Remove version headers from HTTP responses

---

END OF REPORT