# WEB APPLICATION SECURITY ASSESSMENT REPORT

**OWASP Juice Shop Vulnerability Analysis**

---

**Report Information**
**Prepared by: Divyanshu Prajapat**
**Date: December 7, 2025**
**Target Application: OWASP Juice Shop**
**Target URL: http://demo.owasp-juice.shop**
**Testing Duration: 2 hours**
**Testing Methodology: Manual penetration testing following OWASP Top 10 guidelines**

---

## EXECUTIVE SUMMARY

**This security assessment was conducted on the OWASP Juice Shop web application to identify potential vulnerabilities and security weaknesses. The assessment revealed 5 critical security vulnerabilities that could compromise the confidentiality, integrity, and availability of the application and its data.**

**Key Findings:**
- **5 vulnerabilities identified**
- **2 High Severity vulnerabilities**
- **3 Medium Severity vulnerabilities**
- **Multiple OWASP Top 10 categories affected**
- **Immediate remediation recommended for high-severity issues**

---

## VULNERABILITIES IDENTIFIED

### VULNERABILITY #1: SQL Injection Authentication Bypass

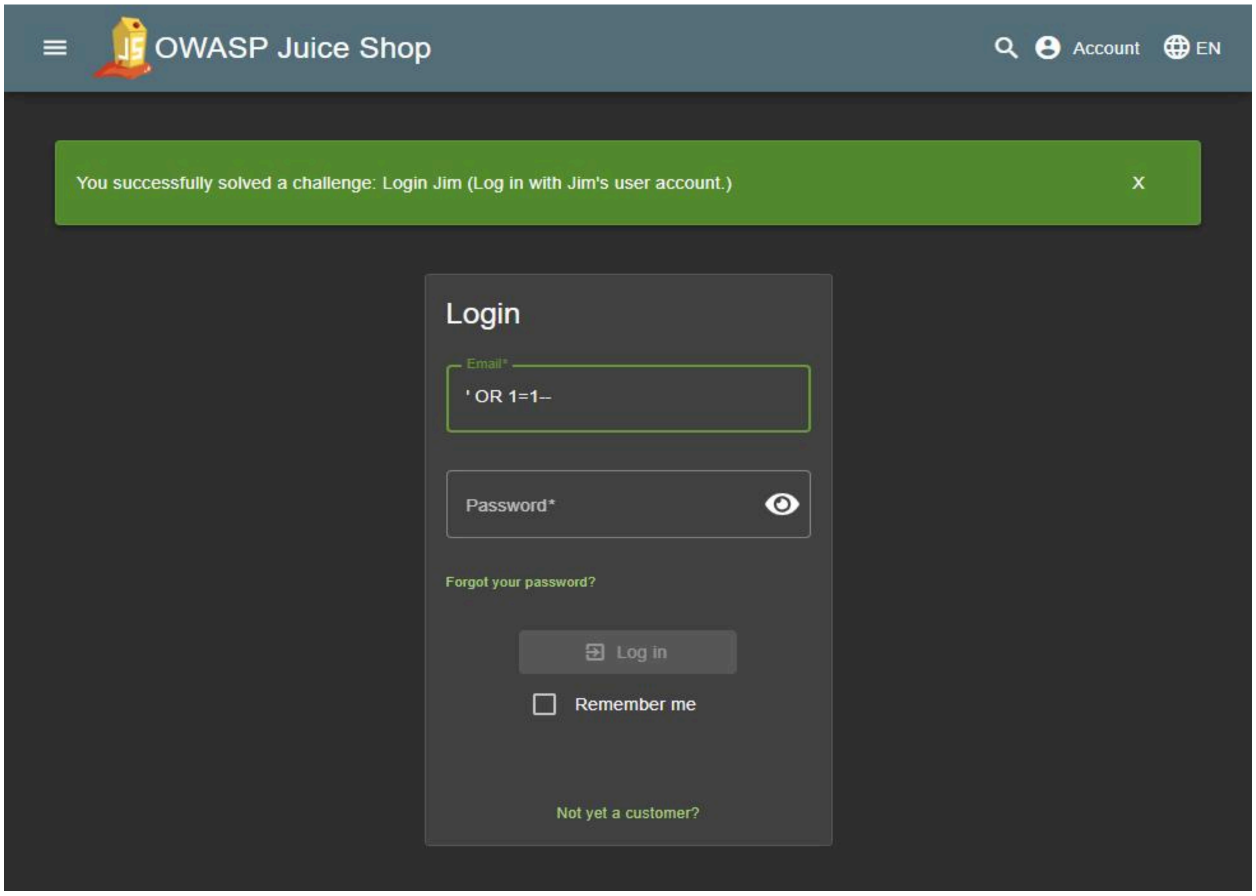
**Severity: HIGH**
**OWASP Top 10: A03:2021 - Injection**
**CVSS Score: 9.1 (Critical)**

**Description:**
**The login functionality is vulnerable to SQL injection attacks. An attacker can bypass authentication by injecting SQL commands into the email field, allowing unauthorized access to user accounts without knowing valid credentials.**

**Location:**
**Login page: http://demo.owasp-juice.shop/#/login**
**Vulnerable parameter: Email field**

**Proof of Concept:**
**Payload used: ' OR 1=1--**
**1. Navigate to the login page**
**2. Enter ' OR 1=1-- in the email field**
**3. Enter any value in the password field**
**4. Click "Log in"**
**5. Result: Successfully logged in as the first user in the database (admin account)**

**Screenshot Evidence:**



You successfully solved a challenge: Login Jim (Log in with Jim's user account.)

Email*: ' OR 1=1--

**SQL Injection - Login form showing payload ' OR 1=1-- in email field**

**Impact:**
- **Complete authentication bypass**
- **Unauthorized access to user accounts**
- **Potential access to administrative functions**
- **Data breach of user information**
- **Compromise of user sessions and sensitive data**

**Remediation:**
1. **Implement parameterized queries (prepared statements) for all database interactions**
2. **Use ORM frameworks that handle SQL injection protection**
3. **Implement input validation and sanitization**
4. **Add additional authentication factors (2FA/MFA)**
5. **Implement rate limiting on login attempts**
6. **Use security frameworks with built-in SQL injection protection**

**VULNERABILITY #2: Reflected Cross-Site Scripting (XSS)**

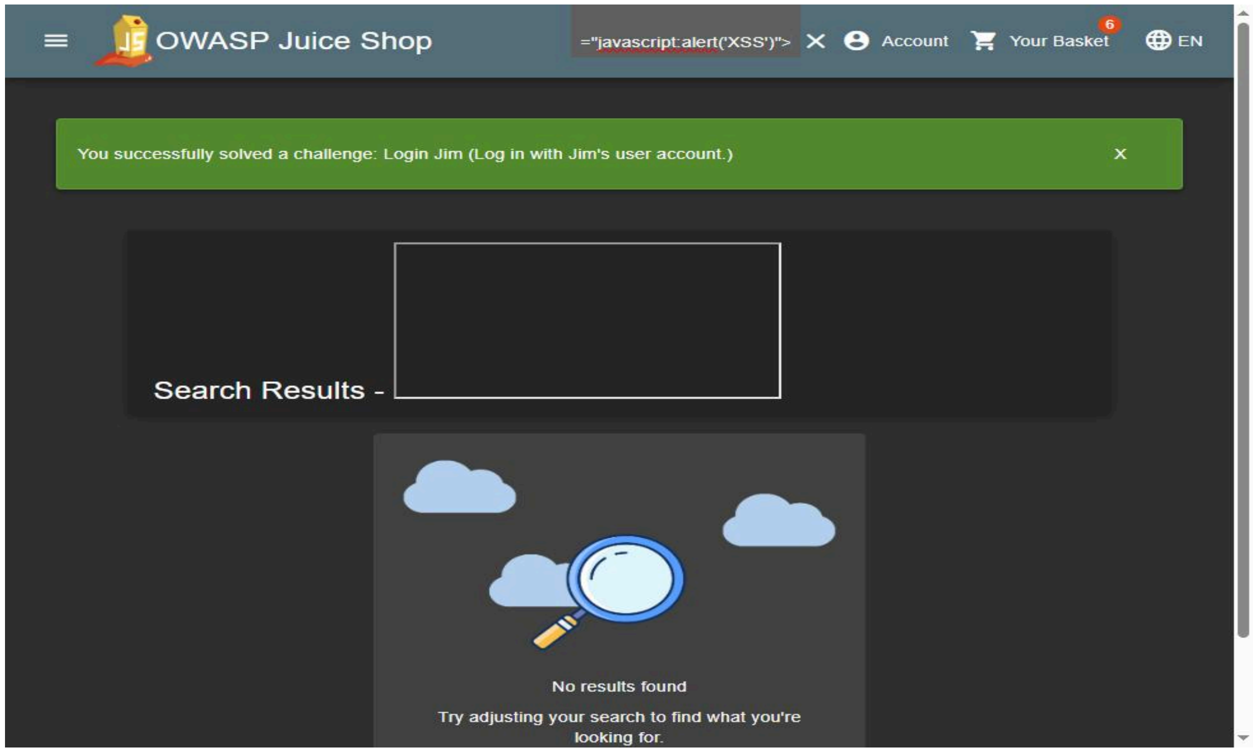
**Severity: HIGH**
**OWASP Top 10: A03:2021 - Injection**
**CVSS Score: 7.1 (High)**

**Description:**
**The search functionality reflects user input without proper sanitization or encoding, allowing attackers to inject malicious JavaScript code that executes in victims' browsers. This can lead to session hijacking, credential theft, and various client-side attacks.**

**Location:**
**Search functionality: http://demo.owasp-juice.shop/#/search**
**Vulnerable parameter: Search query (q parameter)**

**Proof of Concept:**
**Payload used: <iframe src="javascript:alert('XSS')">**
**1. Navigate to the main page**
**2. Click on the search icon**
**3. Enter: <iframe src="javascript:alert('XSS')">**
**4. Press Enter**
**5. Result: The payload is reflected in the URL and rendered in the search results page**

**Screenshot Evidence:**



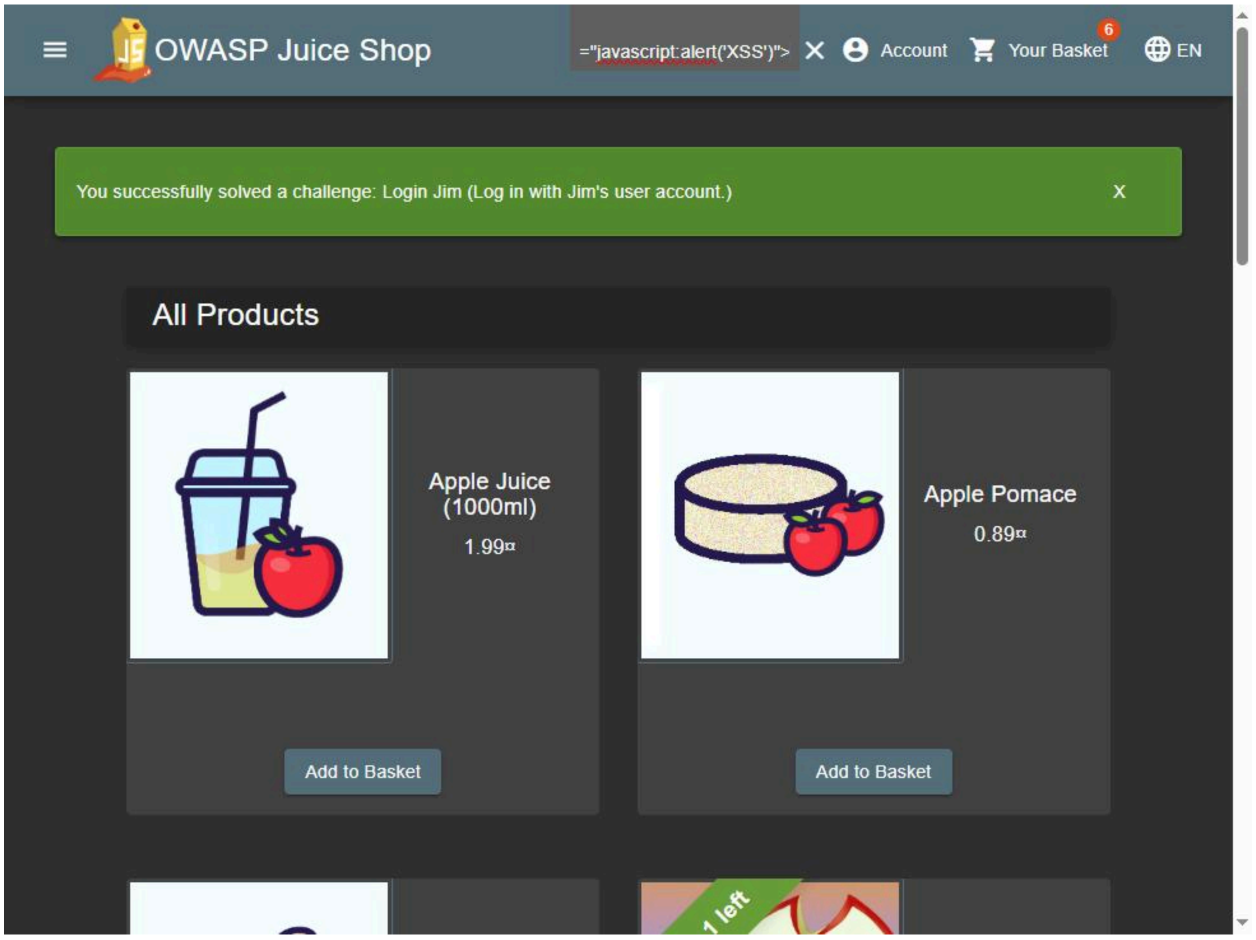
You successfully solved a challenge: Login Jim (Log in with Jim's user account.)

Search Results -

No results found

Try adjusting your search to find what you're looking for.

**XSS - Search field with malicious payload <iframe src="javascript:alert('XSS')">**



**XSS - Search results page showing reflected XSS payload in URL parameter**

**Impact:**
**• Session hijacking and cookie theft**
**• Credential harvesting through fake login forms**
**• Malware distribution**
**• Phishing attacks**
**• Defacement of web pages**
**• Keylogging and user activity monitoring**

**Remediation:**
**1. Implement output encoding for all user-supplied data**
**2. Use Content Security Policy (CSP) headers**
**3. Sanitize input on both client and server side**
**4. Use security libraries for XSS protection**
**5. Implement HTTPOnly and Secure flags on cookies**
**6. Apply context-aware output encoding**

## VULNERABILITY #3: Broken Access Control - Unauthorized Admin Access

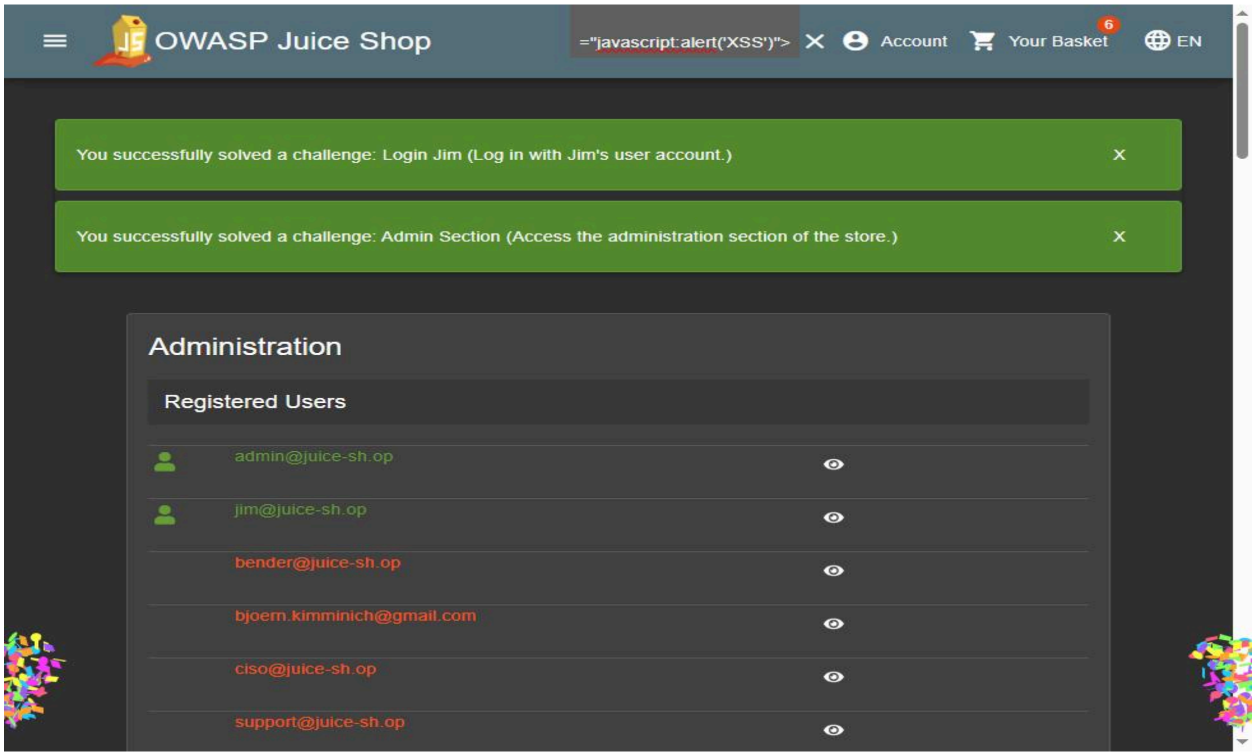
**Severity: MEDIUM**
**OWASP Top 10: A01:2021 - Broken Access Control**
**CVSS Score: 6.5 (Medium)**

**Description:**
**The application's administration panel can be accessed by any authenticated user, regardless of their privilege level. This vulnerability was exploited after gaining access through SQL injection, demonstrating how vulnerabilities can be chained for greater impact.**

**Location:**
**Administration panel: http://demo.owasp-juice.shop/#/administration**

**Proof of Concept:**
**1. Login using SQL injection (as demonstrated in Vulnerability #1)**
**2. Navigate directly to: http://demo.owasp-juice.shop/#/administration**
**3. Result: Successfully accessed admin panel showing all registered users**
**4. Exposed information includes: user email addresses, user roles, and user management functions.**

**Screenshot Evidence:**



You successfully solved a challenge: Login Jim (Log in with Jim's user account.)

You successfully solved a challenge: Admin Section (Access the administration section of the store.)

Administration

Registered Users

admin@juice-sh.op
jim@juice-sh.op
bender@juice-sh.op
bjoern.kimminich@gmail.com
ciso@juice-sh.op
support@juice-sh.op

**Admin Panel Access - Administration section showing full list of registered users with email addresses exposed**

**Impact:**
- **Exposure of all registered user information**
- **Potential user account manipulation**
- **Privacy violations (email addresses exposed)**
- **Information disclosure for further targeted attacks**
- **Privilege escalation possibilities**

**Remediation:**
1. **Implement proper role-based access control (RBAC)**
2. **Verify user permissions on server-side for every request**
3. **Use authorization middleware/guards**
4. **Implement principle of least privilege**
5. **Add audit logging for administrative actions**
6. **Regular security reviews of access control implementations**

---

**VULNERABILITY #4: Security Misconfiguration - Score Board Exposure**

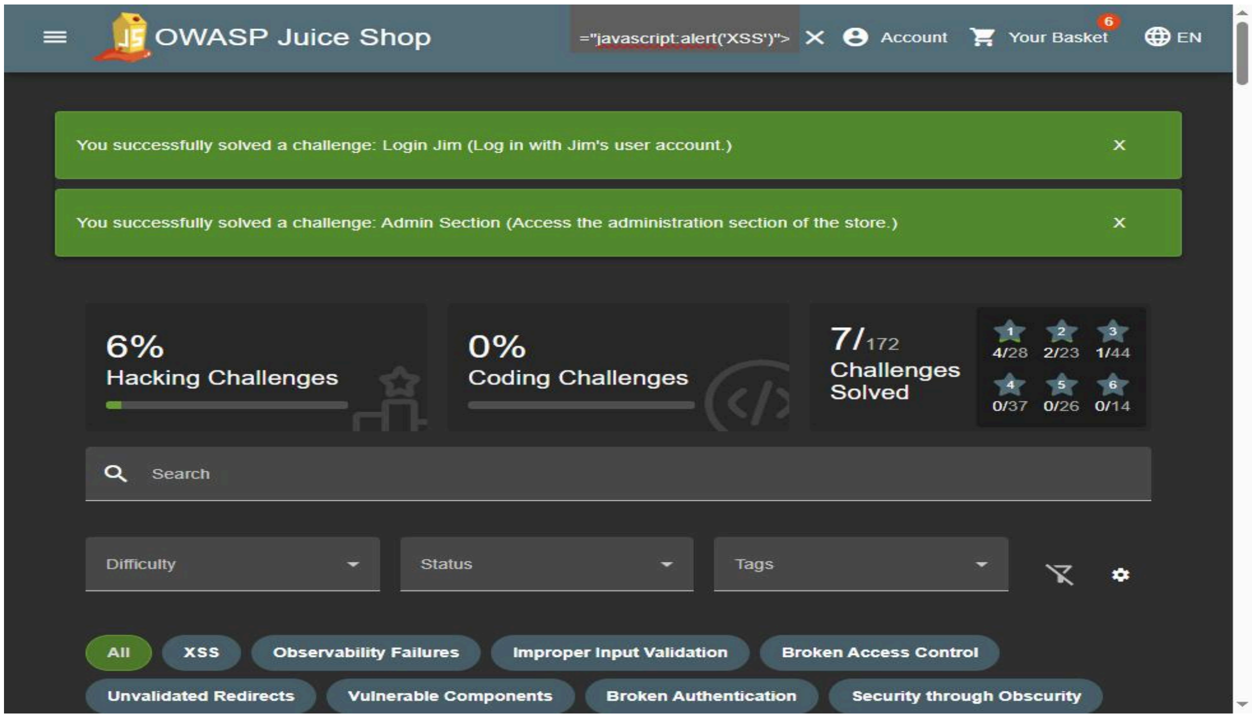
**Severity: MEDIUM**
**OWASP Top 10: A05:2021 - Security Misconfiguration**
**CVSS Score: 5.3 (Medium)**

**Description:**
**The application exposes a hidden score board that reveals the internal challenge structure, difficulty levels, and vulnerability categories. This information disclosure aids attackers in understanding the application's security weaknesses and planning targeted attacks.**

**Location:**
**Score board: http://demo.owasp-juice.shop/#/score-board**

**Proof of Concept:**
**1. Navigate to: http://demo.owasp-juice.shop/#/score-board**
**2. Result: Access granted to score board showing:**
**- All security challenges and their categories**
**- Vulnerability classifications (XSS, Injection, Broken Access Control, etc.)**
**- Difficulty ratings for each vulnerability**
**- Progress tracking (7/172 challenges visible)**

**Screenshot Evidence:**



OWASP Juice Shop ="javascript:alert('XSS')"> Account Your Basket 6 EN

You successfully solved a challenge: Login Jim (Log in with Jim's user account.)

You successfully solved a challenge: Admin Section (Access the administration section of the store.)

6% Hacking Challenges

0% Coding Challenges

7/172 Challenges Solved

| 1 | 2 | 3 |
|---|---|---|
| 4/28 | 2/23 | 1/44 |
| 4 | 5 | 6 |
| 0/37 | 0/26 | 0/14 |

**Score Board - Showing the hidden score board with 7/172 challenges, difficulty ratings, and OWASP Top 10 category mappings**

**Impact:**
**• Detailed reconnaissance information for attackers.**
**• Reveals application's security weaknesses.**
**• Maps vulnerabilities to OWASP Top 10 categories.**
**• Provides a roadmap for systematic exploitation.**
**• Discloses security posture and testing maturity.**

**Remediation:**
**1. Remove or properly secure development/testing features in production.**
**2. Implement environment-based configuration management.**
**3. Use .htaccess or server configuration to block sensitive URLs.**
**4. Implement proper authentication for debugging/admin features.**
**5. Regular security configuration reviews.**
**6. Disable debug modes and verbose error messages in production.**

**VULNERABILITY #5: Directory Listing & Sensitive Data Exposure**

**Severity: MEDIUM**
**OWASP Top 10: A01:2021 - Broken Access Control & A05:2021 - Security Misconfiguration**
**CVSS Score: 6.5 (Medium)**

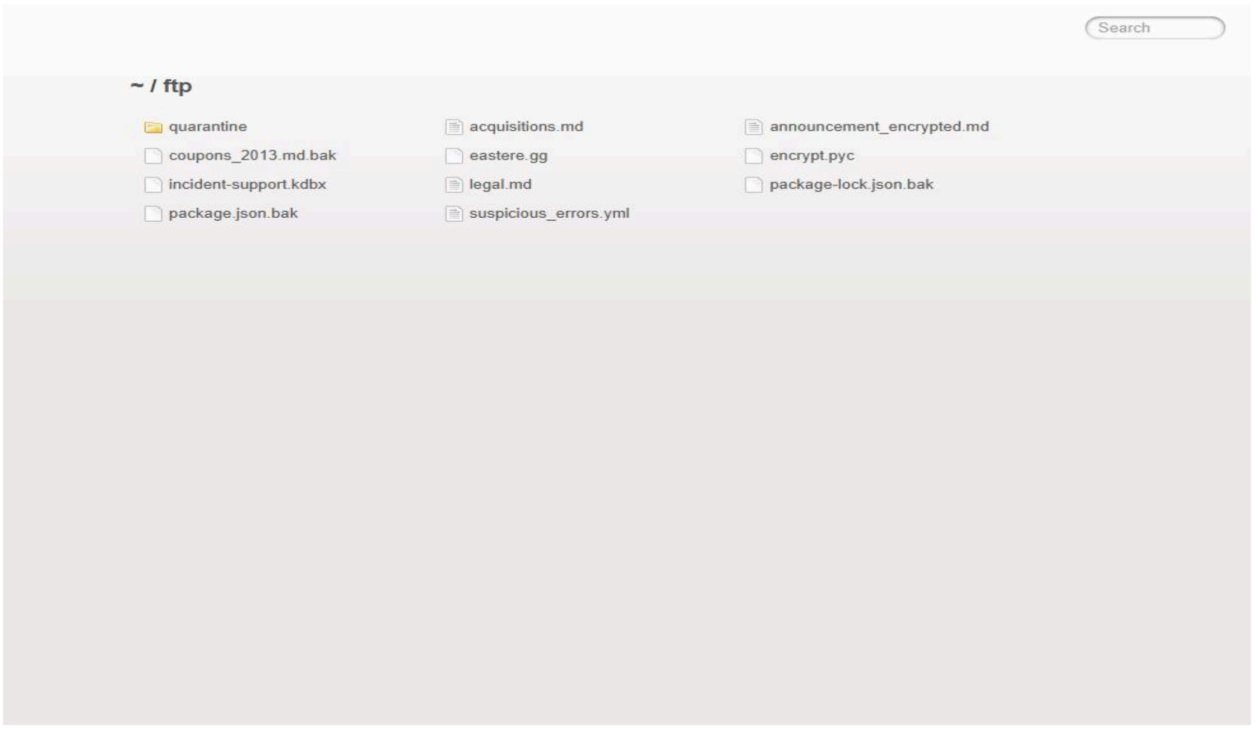
**Description:**
**The /ftp directory is publicly accessible and enables directory listing, exposing sensitive files including backup files, database files, configuration files, and potentially confidential documents. This creates multiple attack vectors for information gathering and system compromise.**

**Location:**
**FTP directory: http://demo.owasp-juice.shop/ftp**

**Proof of Concept:**
**1. Navigate to: http://demo.owasp-juice.shop/ftp**
**2. Result: Directory listing displayed showing multiple sensitive files:**
- **incident-support.kdbx (KeePass password database)**
- **package.json.bak (backup configuration file)**
- **package-lock.json.bak (backup dependency file)**
- **coupons_2013.md.bak (backup business data)**
- **legal.md (accessible legal documents)**
- **suspicious_errors.yml (error logs)**
- **acquisitions.md (business information)**
- **announcement_encrypted.md (encrypted announcements)**
- **encrypt.pyc (compiled Python encryption code)**

**3. Successfully accessed legal.md file content**

**Screenshot Evidence:**

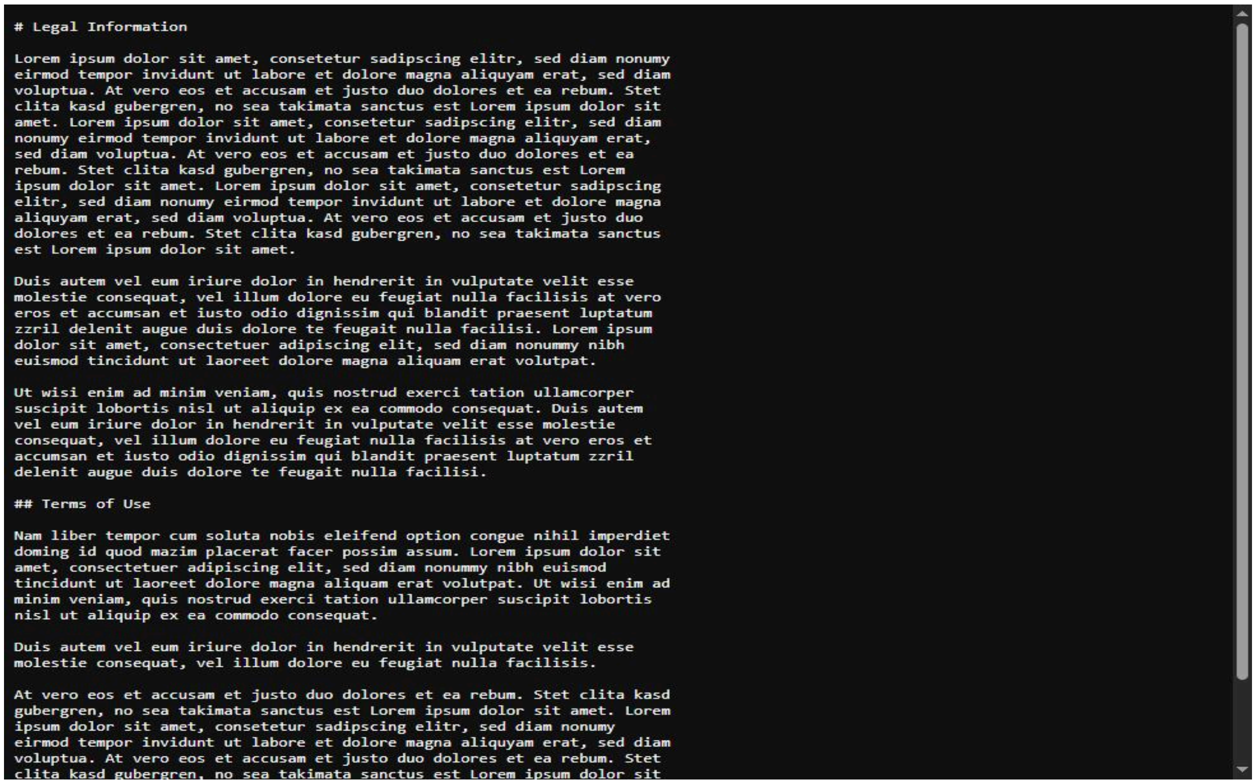


Search

~ / ftp

quarantine
acquisitions.md
announcement_encrypted.md
coupons_2013.md.bak
eastere.gg
encrypt.pyc
incident-support.kdbx
legal.md
package-lock.json.bak
package.json.bak
suspicious_errors.yml

**Directory Listing - Showing /ftp directory with publicly accessible sensitive files including KeePass database, backup files, and configuration files**



```
# Legal Information

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing
elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna
aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo
dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus
est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse
molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero
eros et accumsan et iusto odio dignissim qui blandit praesent luptatum
zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum
dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper
suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem
vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et
accumsan et iusto odio dignissim qui blandit praesent luptatum zzril
delenit augue duis dolore te feugait nulla facilisi.

## Terms of Use

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet
doming id quod mazim placerat facer possim assum. Lorem ipsum dolor sit
amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad
minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis
nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse
molestie consequat, vel illum dolore eu feugiat nulla facilisis.

At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd
gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem
ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
```

**File Access - Showing successful access to legal.md file demonstrating unrestricted file access**

**Impact:**

- **Exposure of backup files containing sensitive configuration**
- **Password database files accessible (incident-support.kdbx)**
- **Information leakage about application structure**
- **Business intelligence disclosure**
- **Potential access to encryption keys or algorithms**
- **Error logs revealing system vulnerabilities**

**Remediation:**

1. **Disable directory listing in web server configuration**
2. **Implement proper access controls on file directories**
3. **Remove or secure backup files from web-accessible locations**
4. **Use .htaccess to deny access to sensitive file types (.bak, .kdbx, etc.)**
5. **Implement file access logging and monitoring**
6. **Regular audits of publicly accessible directories**
7. **Store sensitive files outside web root**

## OWASP TOP 10 VULNERABILITIES

**The vulnerabilities identified map to the following OWASP Top 10 2021 categories:**

**A01:2021 - Broken Access Control**
**• Vulnerability #3: Unauthorized Admin Access**
**• Vulnerability #5: Directory Listing & Sensitive Data Exposure**
**Risk: HIGH - Allows attackers to access restricted resources and administrative functions**

**A03:2021 - Injection**
**• Vulnerability #1: SQL Injection Authentication Bypass**
**• Vulnerability #2: Reflected Cross-Site Scripting (XSS)**
**Risk: CRITICAL - Enables complete system compromise and data theft**

**A05:2021 - Security Misconfiguration**
**• Vulnerability #4: Score Board Exposure**
**• Vulnerability #5: Directory Listing & Sensitive Data Exposure**
**Risk: MEDIUM - Reveals system architecture and aids reconnaissance**

---

## COMPLETE OWASP TOP 10 2021 REFERENCE

**For comprehensive security coverage, below is the complete OWASP Top 10 2021 list with descriptions:**

**A01:2021 - Broken Access Control ✓ FOUND**
**Description: Failures related to restrictions on authenticated users. Attackers exploit flaws to access unauthorized functionality and/or data.**
**Common Weaknesses: Missing access controls, bypassing checks, elevation of privilege**
**Found in Report: Vulnerabilities #3 and #5**

**A02:2021 - Cryptographic Failures**
**Description: Failures related to cryptography which often lead to sensitive data exposure.**
**Common Weaknesses: Weak encryption, storing sensitive data in clear text, inadequate key management**
**Status: Not tested in this assessment**

**A03:2021 - Injection ✓ FOUND**
**Description: User-supplied data is not validated, filtered, or sanitized. Includes SQL, NoSQL, OS command, ORM, LDAP, and Expression Language injection.**
**Common Weaknesses: SQL injection, XSS, command injection**
**Found in Report: Vulnerabilities #1 (SQL Injection) and #2 (XSS)**

**A04:2021 - Insecure Design**
**Description: Represents missing or ineffective control design. Different from insecure implementation.**
**Common Weaknesses: Lack of threat modeling, insecure design patterns, missing security requirements**
**Status: Not specifically tested in this assessment**

**A05:2021 - Security Misconfiguration ✓ FOUND**
**Description: Missing appropriate security hardening or improperly configured permissions.**
**Common Weaknesses: Default credentials, verbose error messages, missing security headers, exposed admin interfaces**
**Found in Report: Vulnerabilities #4 (Score Board) and #5 (Directory Listing)**

**A06:2021 - Vulnerable and Outdated Components**
**Description: Using components with known vulnerabilities, unsupported versions, or not staying current with security patches.**
**Common Weaknesses: Outdated libraries, unpatched systems, unused dependencies**
**Status: Not specifically tested (requires dependency scanning)**

**A07:2021 - Identification and Authentication Failures**
**Description: Confirmation of user identity, authentication, and session management is critical to protect against authentication-related attacks.**
**Common Weaknesses: Credential stuffing, weak passwords, broken session management.**
**Status: Authentication bypass found via SQL injection (Vulnerability #1)**

**A08:2021 - Software and Data Integrity Failures**
**Description: Code and infrastructure that does not protect against integrity violations.**
**Common Weaknesses: Insecure CI/CD pipelines, auto-update without integrity verification, insecure deserialization.**
**Status: Not specifically tested in this assessment.**

**A09:2021 - Security Logging and Monitoring Failures**
**Description: Without logging and monitoring, breaches cannot be detected.**
**Common Weaknesses: Insufficient logging, no alerting, inadequate log protection.**
**Status: Not specifically tested (requires operational assessment)**

**A10:2021 - Server-Side Request Forgery (SSRF)**
**Description: Occurs when a web application fetches a remote resource without validating user-supplied URL.**
**Common Weaknesses: Unvalidated URL parameters, internal network scanning, service enumeration.**
**Status: Not specifically tested in this assessment.**

---

## RECOMMENDATIONS & REMEDIATION TIMELINE

**Immediate Actions (Within 24 hours):**
1. Disable directory listing on /ftp directory
2. Implement rate limiting on login endpoints
3. Add input validation on all user input fields
4. Review and restrict access to administrative panels

**Short-term Actions (Within 1 week):**
1. Implement parameterized queries for all database operations
2. Deploy Content Security Policy (CSP) headers
3. Conduct code review of authentication and authorization logic
4. Remove or secure all development/testing features from production
5. Implement comprehensive logging and monitoring

**Medium-term Actions (Within 1 month):**
1. Implement Web Application Firewall (WAF)
2. Conduct security training for development team
3. Establish secure coding guidelines
4. Implement automated security testing in CI/CD pipeline
5. Regular penetration testing schedule

---

## TOOLS USED

• Manual Testing: Browser Developer Tools
• Application: OWASP Juice Shop (Intentionally Vulnerable Application)
• Testing Framework: OWASP Top 10 2021 Guidelines
• Documentation: Screenshots captured during testing

---

## CONCLUSION

This security assessment of OWASP Juice Shop identified 5 significant vulnerabilities across multiple OWASP Top 10 categories. The most critical findings include SQL Injection and Cross-Site Scripting vulnerabilities that could lead to complete application compromise.

Key Observations:
1. Multiple injection vulnerabilities present serious security risks
2. Broken access control allows unauthorized access to sensitive areas
3. Security misconfigurations expose internal application structure
4. Sensitive data is inadequately protected

The identified vulnerabilities require immediate attention, particularly the SQL injection and XSS issues. Implementation of the recommended remediation steps will significantly improve the application's security posture and protect against common web application attacks.

All findings have been documented with proof-of-concept demonstrations, OWASP Top 10 mappings, impact assessments, and detailed remediation guidance.

---

End of Report
Prepared by: Divyanshu Prajapat
Date: December 7, 2025