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

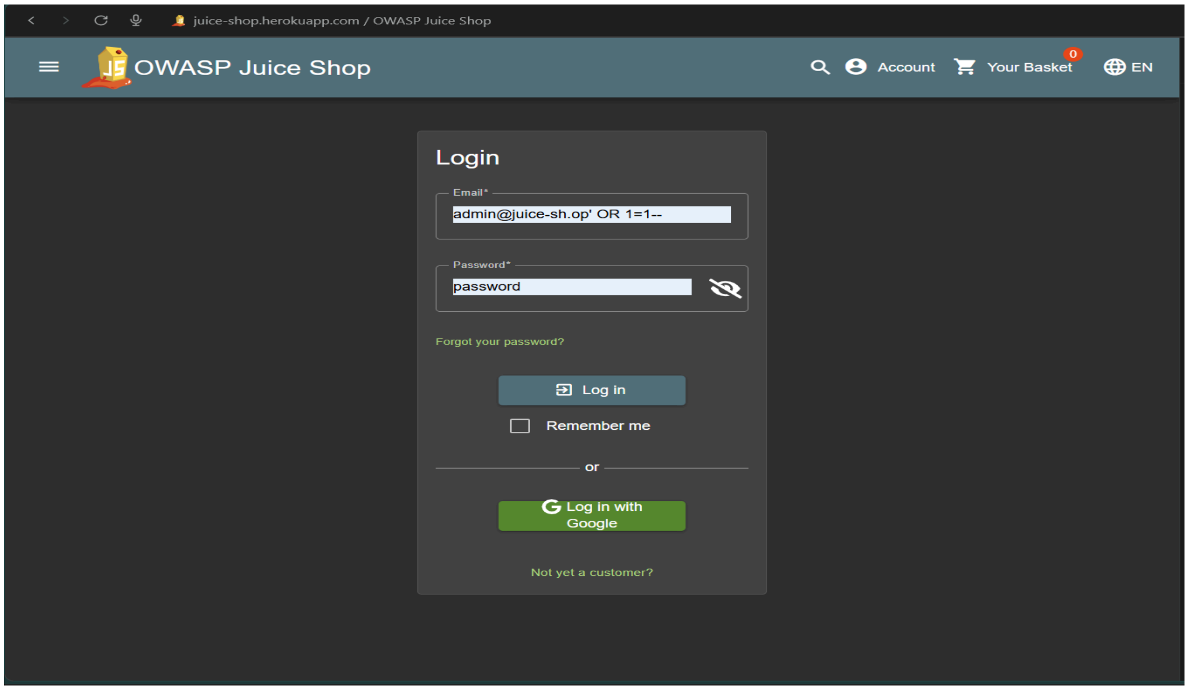
[VULNERABILITY #1] SQL INJECTION - AUTHENTICATION BYPASS

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.

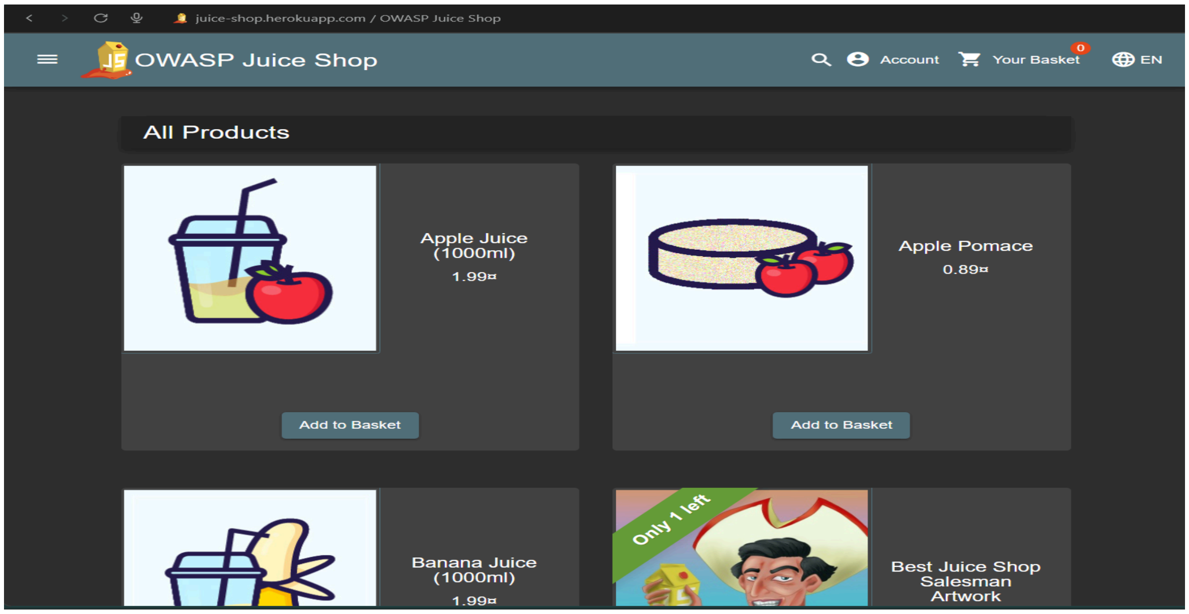
Proof of Concept:
• URL: https://juice-shop.herokuapp.com/#/login

Screenshot Evidence:



Email*: admin@juice-sh.op' OR 1=1--

Password*: password



• 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)

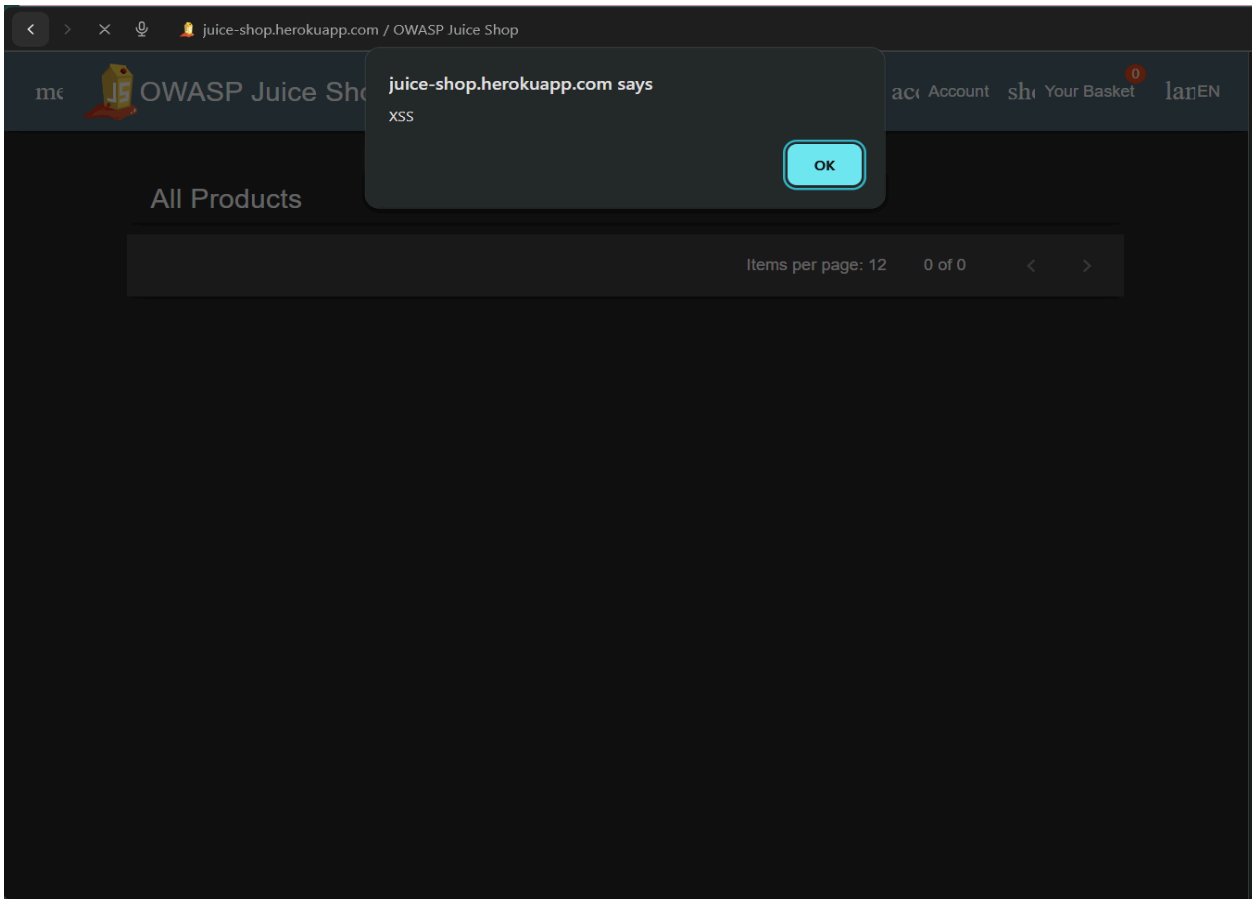
Severity: HIGH 🟠
OWASP Category: A03:2021 - Injection
CVSS Score: 7.3

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:



juice-shop.herokuapp.com says
XSS

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)

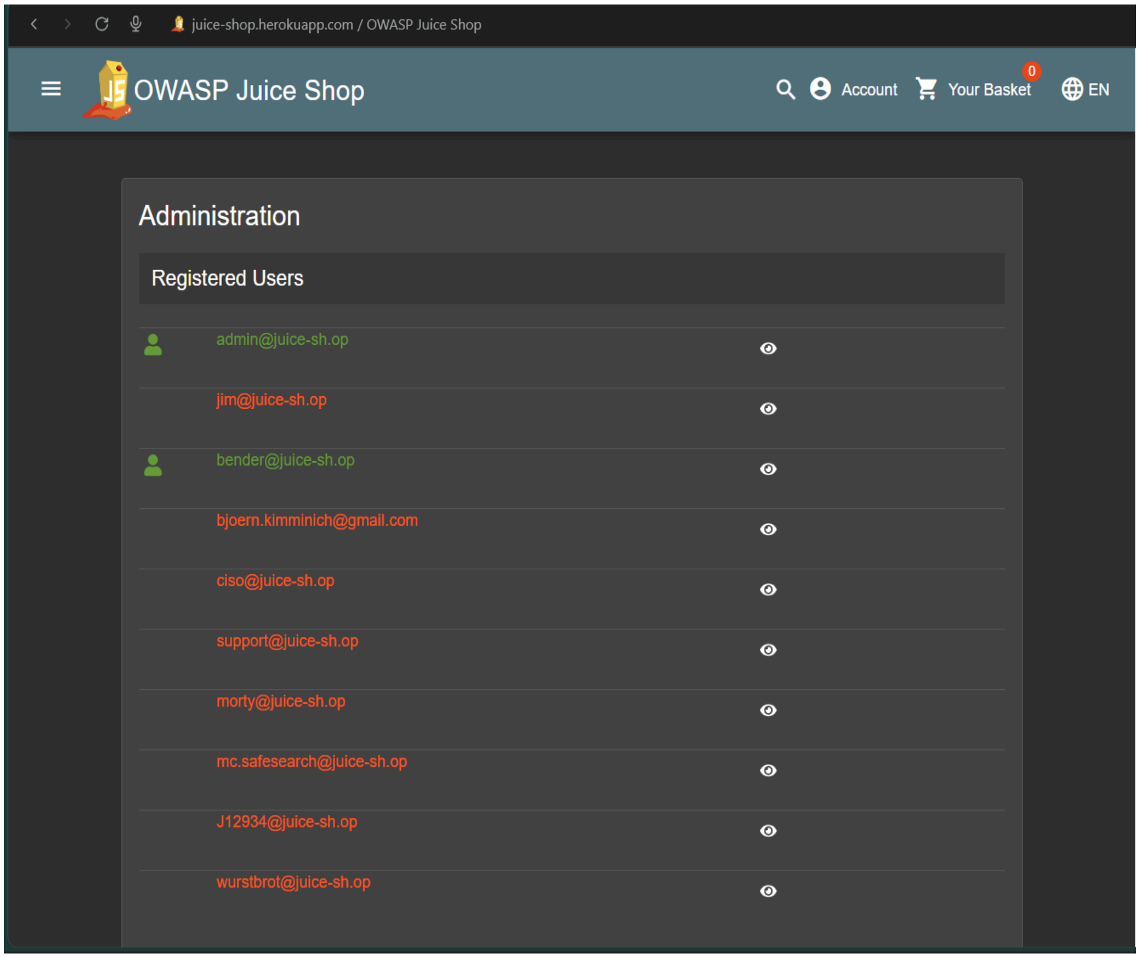
[VULNERABILITY #3] BROKEN ACCESS CONTROL

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**

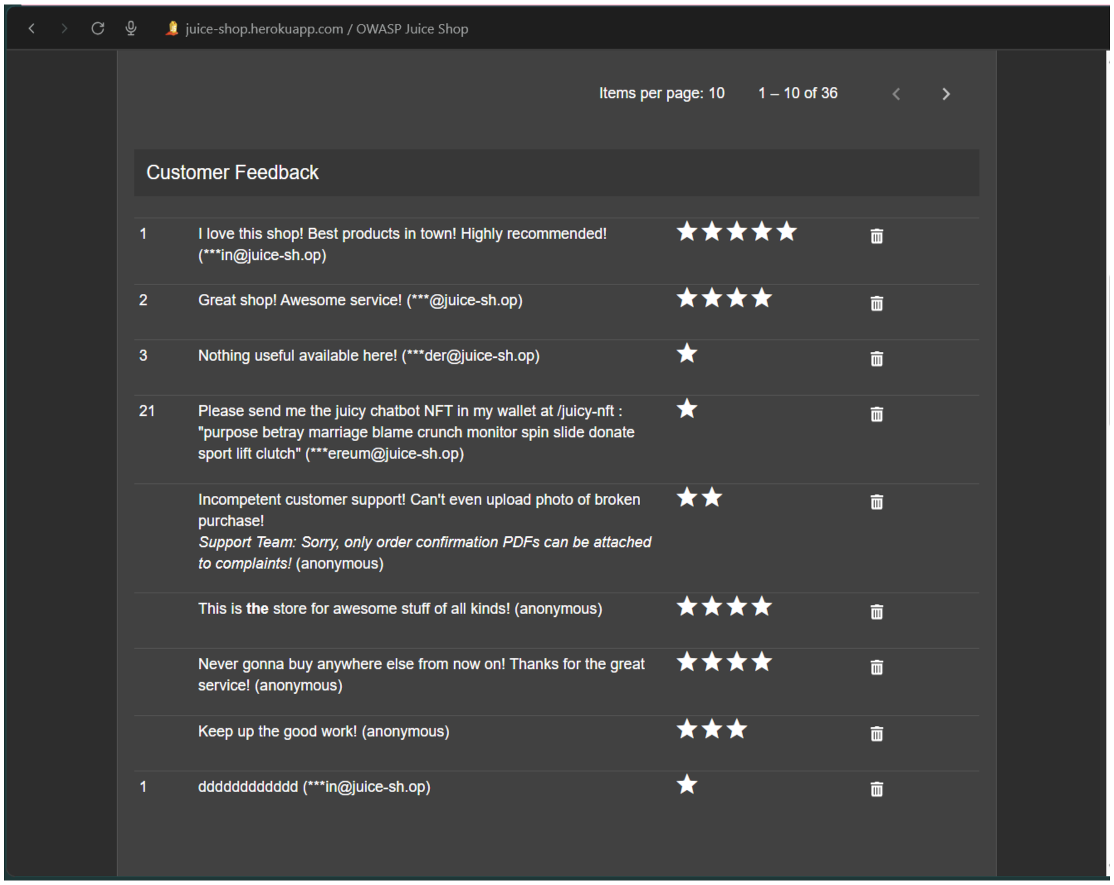
ScreenShot Evidence:



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
morty@juice-sh.op
mc.safesearch@juice-sh.op
J12934@juice-sh.op
wurstbrot@juice-sh.op



juice-shop.herokuapp.com / OWASP Juice Shop

Items per page: 10   1 – 10 of 36

Customer Feedback

1	I love this shop! Best products in town! Highly recommended! (***in@juice-sh.op)	★★★★★	🗑
2	Great shop! Awesome service! (***@juice-sh.op)	★★★★	🗑
3	Nothing useful available here! (***der@juice-sh.op)	★	🗑
21	Please send me the juicy chatbot NFT in my wallet at /juicy-nft : "purpose betray marriage blame crunch monitor spin slide donate sport lift clutch" (***ereum@juice-sh.op)	★	🗑
	Incompetent customer support! Can't even upload photo of broken purchase! *Support Team: Sorry, only order confirmation PDFs can be attached to complaints!* (anonymous)	★★	🗑
	This is **the** store for awesome stuff of all kinds! (anonymous)	★★★★	🗑
	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	★★★★	🗑
	Keep up the good work! (anonymous)	★★★	🗑
1	ddddddddddd (***in@juice-sh.op)	★	🗑

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

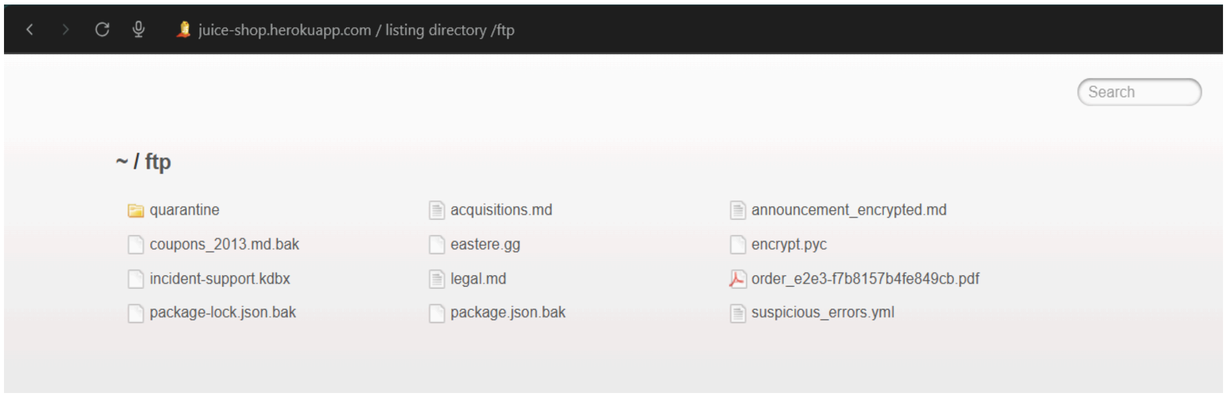
Remediation:
1. **Implement role-based access control (RBAC)**
2. **Verify user permissions on server-side**
3. **Use authentication middleware**
4. **Apply least privilege principle**
5. **Log all access attempts to sensitive resources**

[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:


juice-shop.herokuapp.com / listing directory /ftp

~ / ftp

quarantine	acquisitions.md	announcement_encrypted.md
coupons_2013.md.bak	eastere.gg	encrypt.pyc
incident-support.kdbx	legal.md	order_e2e3-f7b8157b4fe849cb.pdf
package-lock.json.bak	package.json.bak	suspicious_errors.yml

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

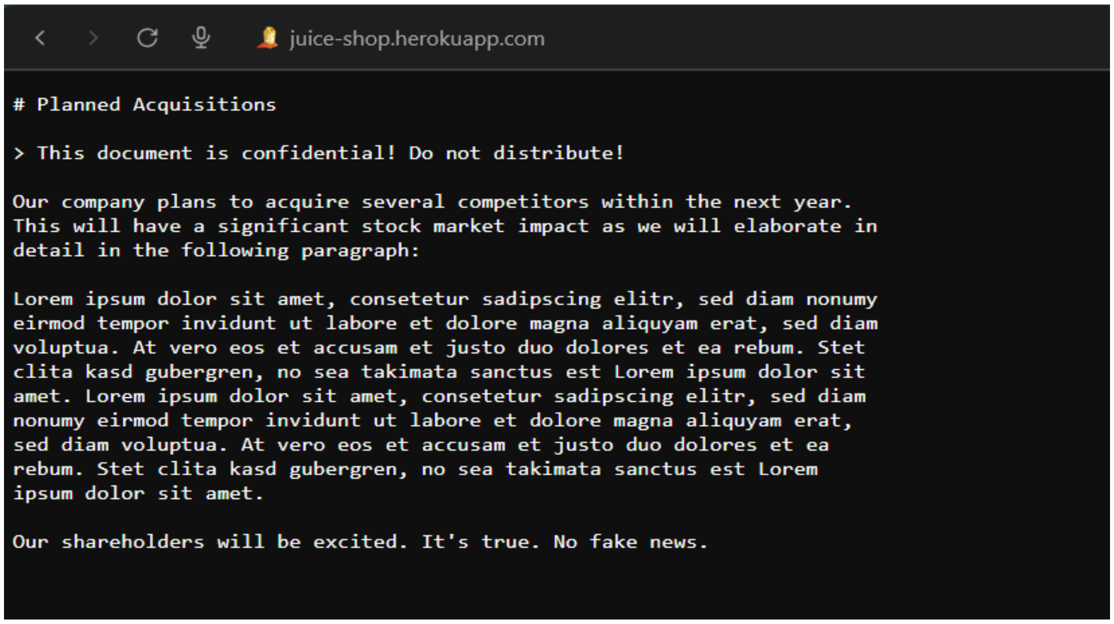
[VULNERABILITY #5] SECURITY MISCONFIGURATION - ERROR DISCLOSURE

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:



```
# Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:

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.

Our shareholders will be excited. It's true. No fake news.
```

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

OWASP TOP 10 (2021) COMPLIANCE CHECKLIST

☑ A01:2021 - Broken Access Control
Status: VULNERABLE - Admin panel accessible without authorization

☑ A02:2021 - Cryptographic Failures
Status: VULNERABLE - Sensitive files exposed publicly

☑ A03:2021 - Injection
Status: VULNERABLE - SQL Injection and XSS detected

☐ A04:2021 - Insecure Design
Status: NOT TESTED in this assessment

☑ A05:2021 - Security Misconfiguration
Status: VULNERABLE - Error messages expose system details

☐ A06:2021 - Vulnerable and Outdated Components
Status: PARTIAL - Version disclosure detected

☐ A07:2021 - Identification and Authentication Failures
Status: VULNERABLE - SQL Injection bypasses authentication

☐ A08:2021 - Software and Data Integrity Failures
Status: NOT TESTED in this assessment

☐ A09:2021 - Security Logging and Monitoring Failures
Status: NOT TESTED in this assessment

☐ A10:2021 - Server-Side Request Forgery (SSRF)
Status: NOT TESTED in this assessment

RECOMMENDATIONS & REMEDIATION TIMELINE

IMMEDIATE (0-7 days):
1. Disable public access to /ftp directory
2. Implement parameterized queries for all database operations
3. Add role-based access control to admin panel
4. Configure custom error pages (disable stack traces)

SHORT-TERM (1-4 weeks):
1. Implement input validation and output encoding
2. Add Content Security Policy headers
3. Remove all backup files from production
4. Conduct code review for similar vulnerabilities
5. Implement security logging and monitoring

LONG-TERM (1-3 months):
1. Security awareness training for development team
2. Integrate SAST/DAST tools in CI/CD pipeline
3. Regular penetration testing schedule
4. Establish secure coding standards
5. Implement Web Application Firewall (WAF)

TOOLS & METHODOLOGY

Testing Tools Used:
- OWASP ZAP - Automated vulnerability scanning
- Burp Suite Community Edition - Manual testing
- Browser Developer Tools - Request/response analysis
- Manual code review and penetration testing

Testing Methodology:
1. Reconnaissance and information gathering
2. Vulnerability identification and scanning
3. Manual exploitation and verification
4. Impact assessment

5. Documentation and remediation recommendations

Standards Followed:
• OWASP Testing Guide v4.2
• OWASP Top 10 (2021)
• CVSS v3.1 Scoring

CONCLUSION

This security assessment revealed CRITICAL vulnerabilities that require immediate attention. The application is vulnerable to:

• Authentication bypass through SQL injection
• Unauthorized access to administrative functions
• Exposure of sensitive business data
• Cross-site scripting attacks
• Information disclosure through error messages

These vulnerabilities could lead to:
- Complete system compromise
- Data breaches affecting all users
- Regulatory compliance violations
- Reputational damage
- Financial losses

RISK SUMMARY:
🔴 CRITICAL: 2 vulnerabilities
🟠 HIGH: 2 vulnerabilities
🟡 MEDIUM: 1 vulnerability

RECOMMENDATION:
Immediate remediation of all CRITICAL and HIGH severity vulnerabilities is strongly recommended before the application handles production data or customer information.

For questions or clarification on any findings, please contact the security assessment team.

END OF REPORT