🛡️ Web Application Security Assessment Report

📄 Task: Web Application Security Testing

🧪 Target: OWASP Juice Shop

🛠️ Tools Used: Burp Suite, OWASP ZAP, SQLMap, Browser Developer Tools

👤 Intern: Sparsh Agarwal

📆 Date: 02-May-2025

---

1. Introduction

This report documents the findings of a security assessment performed on OWASP Juice Shop, an intentionally vulnerable web application used for ethical hacking and security training. The objective was to identify common web vulnerabilities such as SQL Injection, XSS, and broken authentication.

---

2. Environment Setup

- **Platform**: Kali Linux
- **Juice Shop Version**: Prebuilt standalone release
- **Access URL**: http://localhost:3000
- **Tools Used**:
  - Burp Suite
  - OWASP ZAP
  - SQLMap

3. Testing Methodology

The testing followed the OWASP Top 10 principles. The app was examined for input validation flaws, session handling issues, insecure direct object references, and other exploitable weaknesses using a combination of automated scans and manual testing.

4. Findings & Vulnerabilities

🔹 Finding 1: SQL Injection

- **Location**: Login Form
- **Payload**: `' OR '1'='1`
- **Impact**: Authentication bypass
- **Severity**: High
- **Tool**: Manual testing + SQLMap
- **Mitigation**: Use parameterized queries and input sanitization

🔹 Finding 2: Stored XSS

- **Location**: Feedback Form
- **Payload**: `<img src=x onerror=alert(1)>`
- **Impact**: Stored XSS executed for all users viewing feedback
- **Severity**: Medium
- **Tool**: Manual browser test
- **Mitigation**: Escape HTML characters, use Content Security Policy (CSP)

🔹 Finding 3: Broken Authentication

- **Location**: JWT Token Manipulation
- **Impact**: Changed user role to admin via JWT tampering
- **Severity**: High
- **Tool**: Burp Suite
- **Mitigation**: Use signed JWTs with server-side role verification

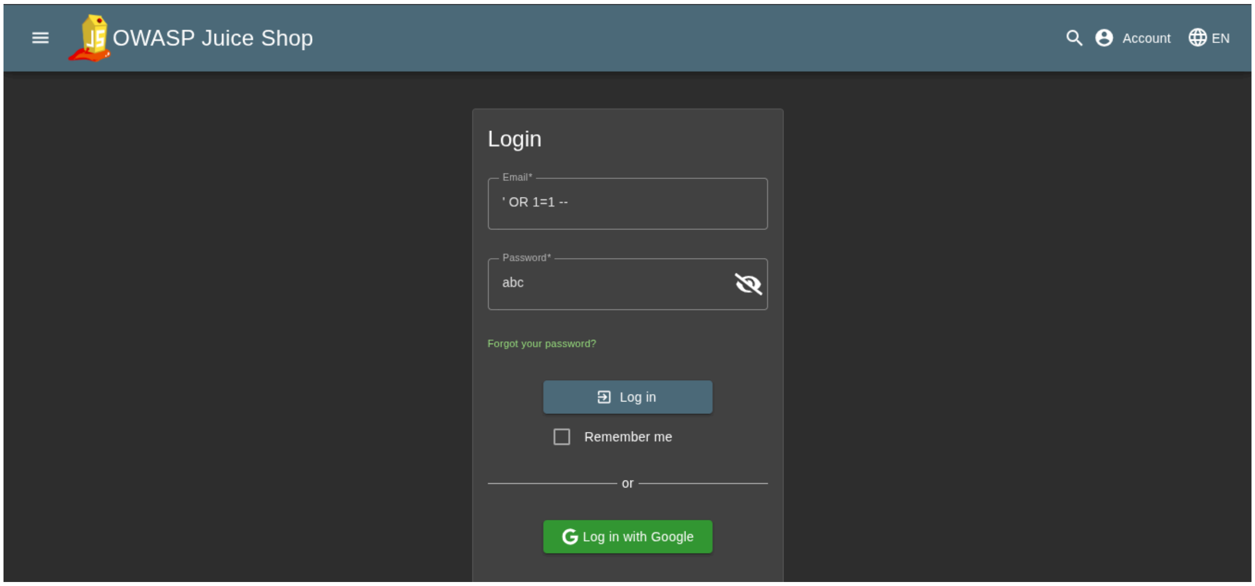
---

🔹 Finding 4: Sensitive Data in Responses

- **Location**: `/rest/user/login` response body
- **Impact**: Partial user information exposed in plain text
- **Severity**: Medium
- **Tool**: Burp Suite
- **Mitigation**: Avoid exposing unnecessary details in responses

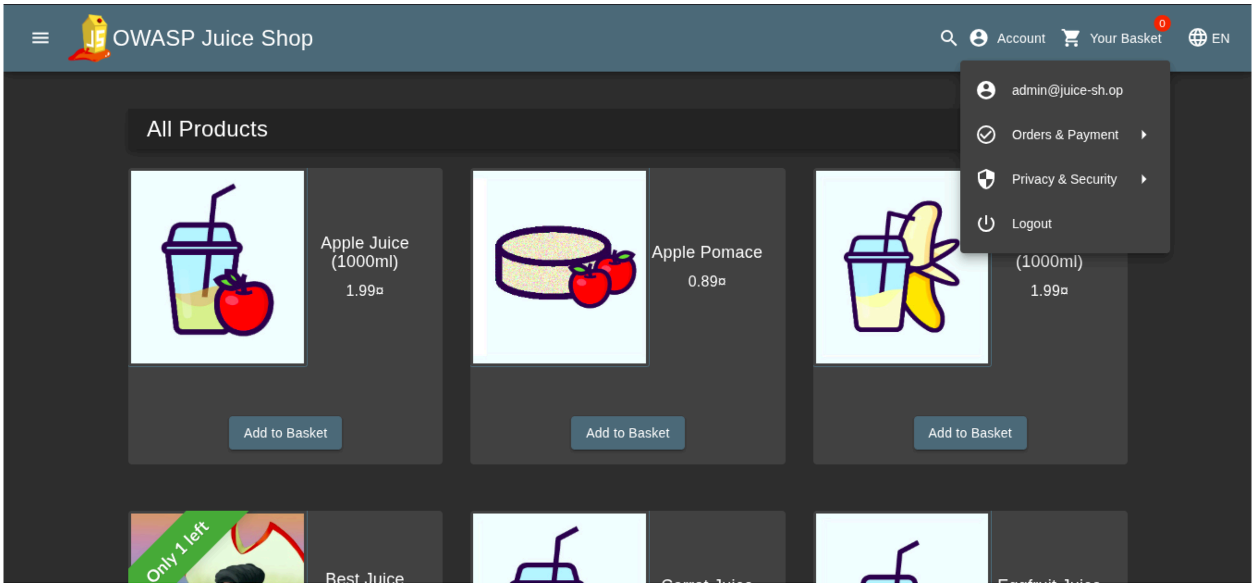
---

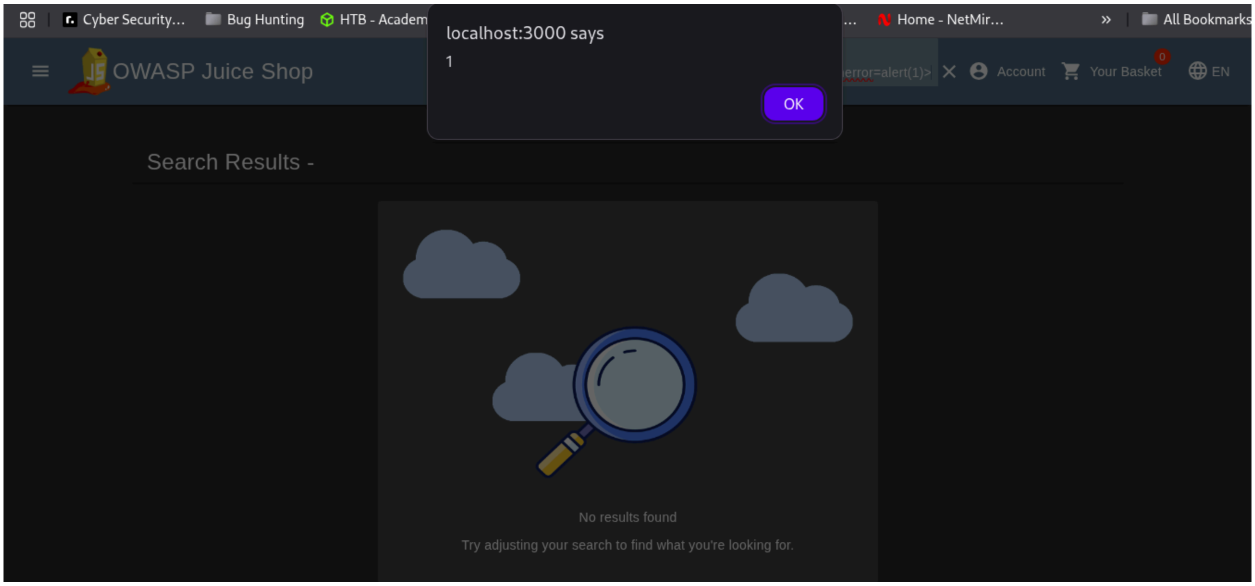
5. Screenshots



Email*
' OR 1=1 --

Password*
abc





localhost:3000 says

1

Search Results -

No results found

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

---

6. Recommendations

- Implement server-side input validation
- Use secure password storage (e.g., bcrypt)
- Sanitize all user-generated content
- Enable HTTPS and enforce secure headers
- Rotate and sign JWT tokens securely

---

7. Conclusion

The OWASP Juice Shop application exhibited several high-risk vulnerabilities commonly found in real-world web apps. This exercise demonstrated key penetration testing skills such as SQL injection exploitation, XSS detection, and session manipulation.