

# **VULNERABILITY ASSESSMENT & PENETRATION TESTING (VAPT)**

Target Application: OWASP Juice Shop (v19.1.1)

Date: 07 December 2025

Tester: SREESHNAV K

Tools Used: Kali Linux, Burp Suite, OWASP ZAP

## **1. EXECUTIVE SUMMARY**

A security assessment was performed on the OWASP Juice Shop web application to identify security flaws. The assessment followed standard ethical hacking procedures. During the test, three (3) critical vulnerabilities were identified:

- SQL Injection: Allowing unauthorized administrative access.
- Cross-Site Scripting (XSS): Allowing execution of malicious scripts.
- Broken Access Control: Exposing sensitive internal pages.

These vulnerabilities pose a high risk to the application and require immediate remediation.

## **2. VULNERABILITY FINDINGS**

Finding 1: SQL Injection (Authentication Bypass)

- Severity: Critical
- OWASP Category: A03:2021 – Injection
- Description: The login form is vulnerable to SQL Injection. By injecting a standard payload, an attacker can manipulate the database query to log in as the Administrator without a valid password.
- Proof of Concept:
  - Payload: ' OR 1=1 --
  - Result: The application accepted the payload and granted administrative access.

- Evidence:

The screenshot shows a web browser window for the OWASP Juice Shop application. The URL is `localhost:3000/#/search`. A green notification bar at the top says "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)". Below this, there is a section titled "All Products" displaying two items:

Image	Name	Price
	Apple Juice (1000ml)	1.99¤
	Apple Pomace	0.89¤

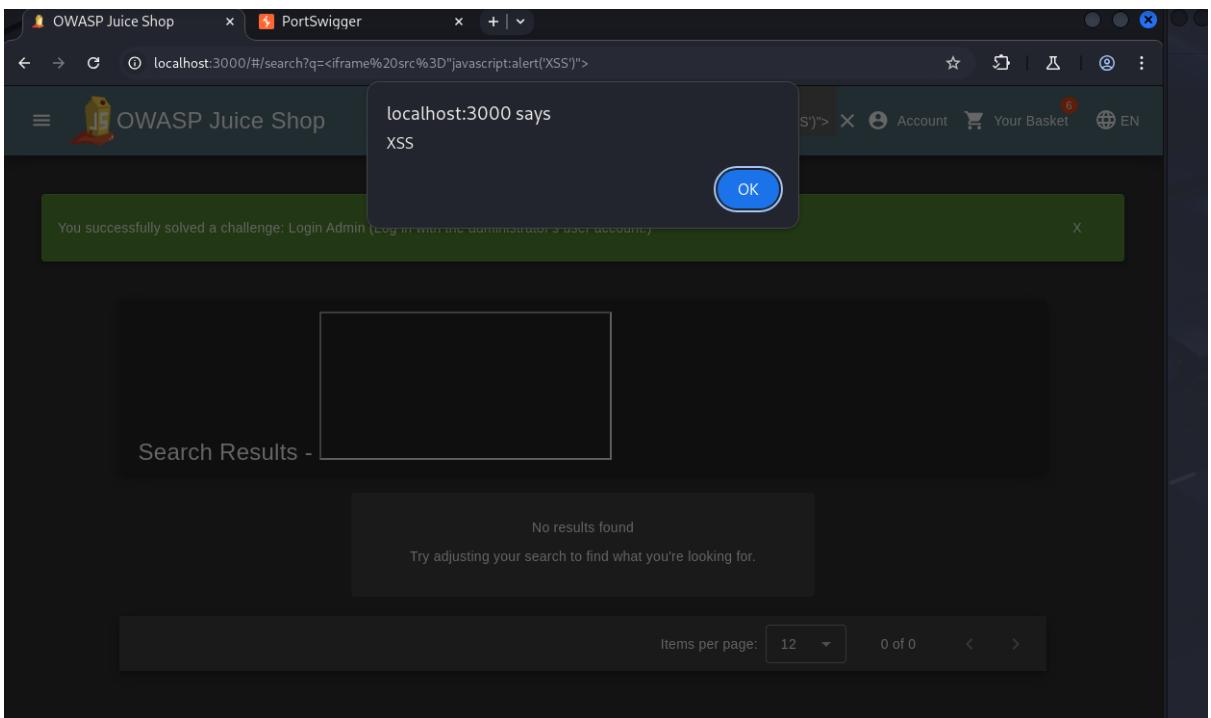
Each item has a "Add to Basket" button below it.

- Remediation: Implement parameterized queries (Prepared Statements) for all database inputs to prevent code injection.

## Finding 2: Reflected Cross-Site Scripting (XSS)

- Severity: High
- OWASP Category: A03:2021 – Injection
- Description: The application's search function fails to sanitize user input. This allows an attacker to inject JavaScript code that executes in the victim's browser.
- Proof of Concept:
  - Payload: `<iframe src="javascript:alert('XSS')">`
  - Result: The application executed the script, displaying an alert popup.

- Evidence:



- Remediation: Implement strict input validation and output encoding (sanitization) for all user-supplied data.

### Finding 3: Broken Access Control (Sensitive Page Exposure)

- Severity: Medium
- OWASP Category: A01:2021 – Broken Access Control
- Description: The application relies on "security by obscurity" to hide sensitive pages. The "Score Board" page is accessible to any unauthenticated user who guesses the correct URL, bypassing access controls.
- Proof of Concept:
  - URL: <http://localhost:3000/#/score-board>
  - Result: The user successfully accessed the restricted internal scoreboard.

- Evidence:

The screenshot shows the OWASP Juice Shop application interface. At the top, there are two green success messages: "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)" and "You successfully solved a challenge: Score Board (Find the carefully hidden 'Score Board' page.)". Below this, the main dashboard displays progress bars for "Hacking Challenges" (2%) and "Coding Challenges" (0%). It shows "2/122 Challenges Solved" with a star rating of 1/2 stars (1/23, 0/44). The navigation bar includes filters for "All", "XSS", "Observability Failures", "Improper Input Validation", "Broken Access Control", "Unvalidated Redirects", "Vulnerable Components", "Broken Authentication", "Security through Obscurity", and "Insecure Deserialization", with "All" currently selected. The main content area lists challenges under categories like "Miscellaneous" and "XSS". One challenge, "Score Board", is highlighted with a green dot. Another challenge, "DOM XSS", is described as "Perform a DOM XSS attack with <iframe src='javascript:alert('xss')'>". A third challenge, "Bonus Payload", is described as "Use the bonus payload <iframe width='100%' height='166%' scrolling='no' src='https://www.youtube.com/embed/dQw4w9WgXcQ?autoplay=1'>". A fourth challenge, "Privacy Policy", is described as "Read our privacy policy.".

- Remediation: Implement proper role-based access control (RBAC) checks on the server side to ensure only authorized users can load restricted pages.