

SQL Injection Penetration Test Report on OWASP Juice Shop

Prepared by: Elroy Fernandes

Date: January 13, 2026

Environment: Kali Linux (VM)

Target: OWASP Juice Shop v19.1.1

Test Type: SQL Injection Assessment

Table of content

Executive Summary.....	3
Scope and Objectives.....	3
Scope.....	3
Objectives.....	3
Technical Finding: SQL Injection (Auth Bypass).....	4
Steps to Reproduce (Proof of Concept).....	4
Impact.....	6
Technical Analysis.....	6
Remediation recommendations.....	7
Conclusion.....	7

Executive Summary

This report documents a successful SQL injection attack against the OWASP Juice Shop login page conducted in a controlled, local environment. The vulnerability allowed authentication bypass using a simple SQL payload (' OR 1=1--), granting unauthorized administrative access. This assessment was performed for educational purposes to demonstrate common web application vulnerabilities and remediation strategies.

Key Finding: High Severity SQL Injection (CWE-89) on login functionality.

Impact: Complete authentication bypass, full admin privileges.

Status: Proof of Concept Achieved

Scope and Objectives

Scope

- **Application:** OWASP Juice Shop (Intentionally vulnerable web application)
- **Environment:** Local Kali Linux deployment
- **Target Component:** User Login page
- **Testing Method:** Manual penetration testing

Objectives

- Identify and exploit authentication vulnerabilities
- Demonstrate SQL injection techniques
- Document findings and remediation steps
- Showcase ethical security testing methodology

Technical Finding: SQL Injection (Auth Bypass)

Description

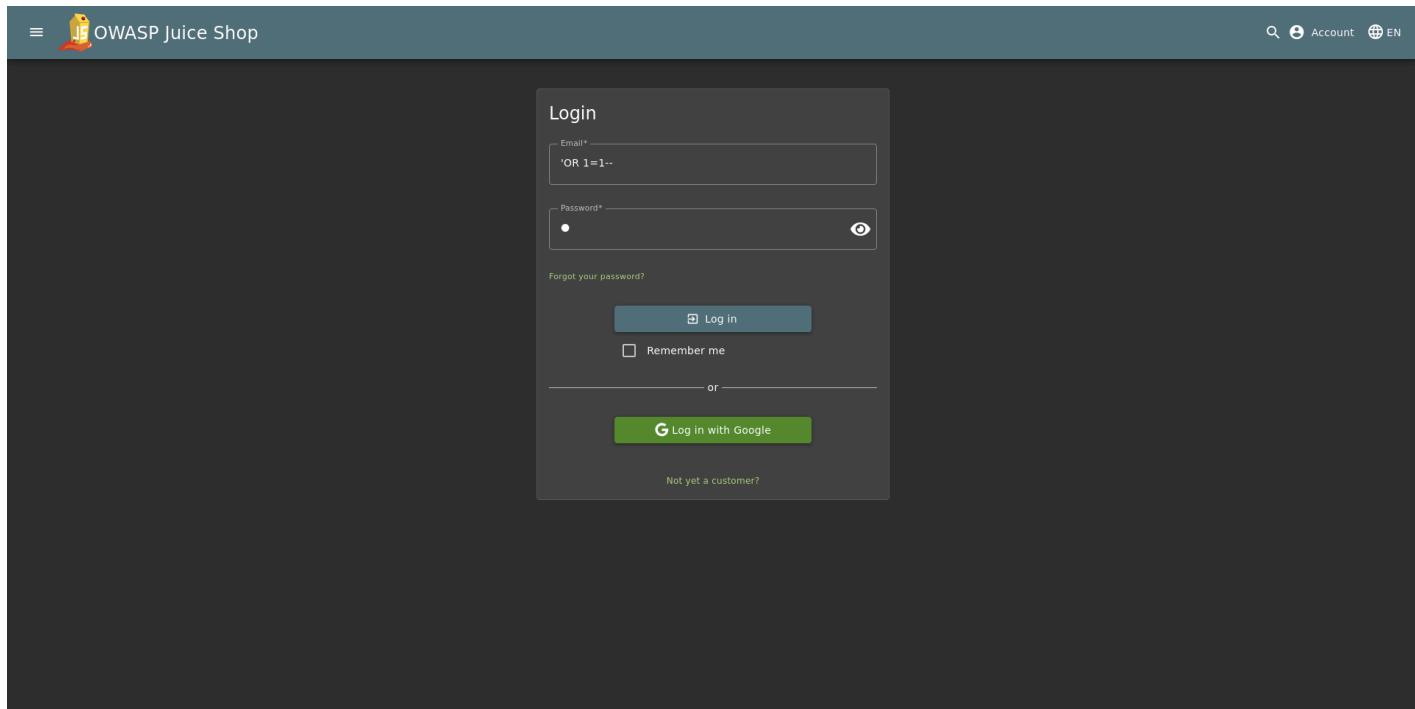
The login form does not use parameterized queries or prepared statements when validating user credentials. By injecting SQL syntax into the email/username field, an attacker can easily manipulate the back-end database query logic to always return “True”, effectively logging the attacker into the first account in the database (typically the Admin)

Steps to Reproduce (Proof of Concept)

1. Navigate to the login page of the OWASP Juice Shop.

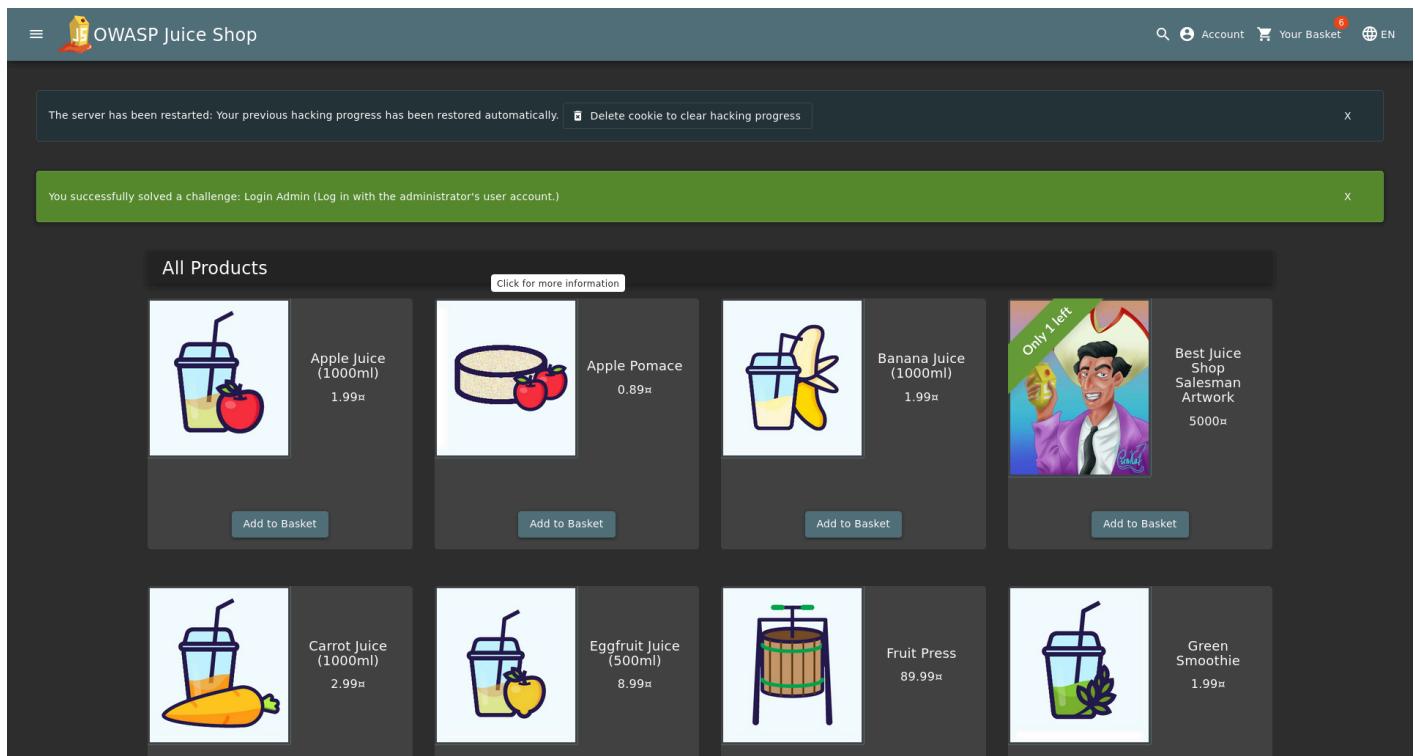
The screenshot shows the OWASP Juice Shop login interface. At the top, there is a navigation bar with a menu icon, the logo, and the text "OWASP Juice Shop". On the right side of the header are search, account, and language selection ("EN") icons. The main content area has a dark background. A modal dialog box titled "Login" is centered. Inside the dialog, there is an error message: "Invalid email or password." Below this, there are two input fields: "Email*" containing "admin" and "Password*" containing a series of dots (...). To the right of the password field is an "Eye" icon for password visibility. Below the inputs is a link "Forgot your password?". At the bottom of the dialog are two buttons: "Log in" (disabled) and "Remember me" (unchecked). A horizontal line with the word "or" separates this from a green button labeled "Log in with Google" with a "G" icon. At the very bottom of the dialog, there is a link "Not yet a customer?".

2. In the Email/Username field, enter the following payload ‘ OR 1=1 --



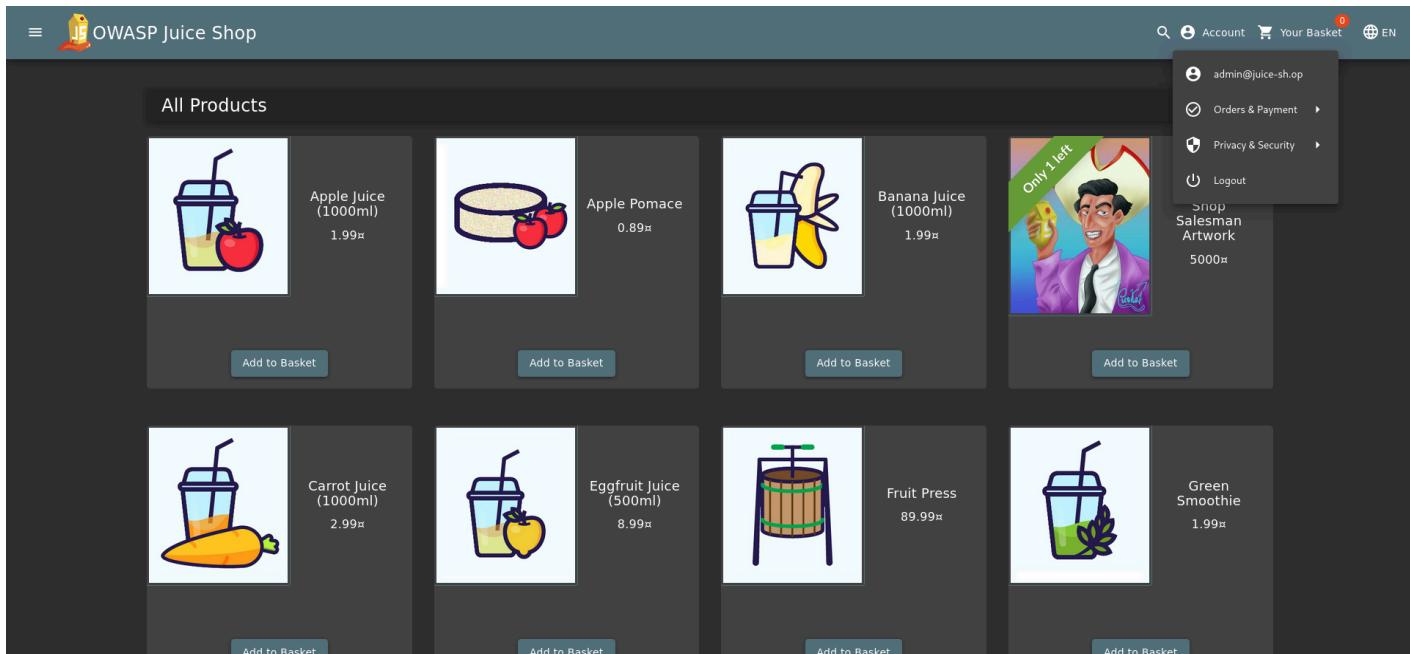
The screenshot shows the OWASP Juice Shop login page. The URL bar at the top displays "OWASP Juice Shop". The main content is a "Login" form with a dark background. The "Email*" field contains the payload "'OR 1=1--". The "Password*" field has a single character, a dot. Below the fields are "Forgot your password?", "Log in" (with a checkbox for "Remember me"), and "Log in with Google". A "Not yet a customer?" link is at the bottom.

3. Enter any arbitrary string in the Password field (e.g: SomethingPassword1)
4. Click Login.
5. **Result:** The Application processes the query, ignores the password check due to the comment operator (- -), and authenticates the user as the admin@juice-sh.op



The screenshot shows the OWASP Juice Shop home page after a successful login. The header includes the logo, "OWASP Juice Shop", a search icon, "Account", "Your Basket" (with a red notification badge), and "EN". A message at the top says "The server has been restarted: Your previous hacking progress has been restored automatically." and a "Delete cookie to clear hacking progress" button. A green banner below it says "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)". The main content is a grid of "All Products".

All Products	
 Apple Juice (1000ml) 1.99¤ Add to Basket	 Apple Pomace 0.89¤ Add to Basket
 Banana Juice (1000ml) 1.99¤ Add to Basket	 Only 1 left! Best Juice Shop Salesman Artwork 5000¤ Add to Basket
 Carrot Juice (1000ml) 2.99¤ Add to Basket	 Eggfruit Juice (500ml) 8.99¤ Add to Basket
 Fruit Press 89.99¤ Add to Basket	 Green Smoothie 1.99¤ Add to Basket



Impact

- **Severity:** Critical
- **Confidentiality:** High (Access to all user data and PII)
- **Integrity:** High (Ability to modify products, reviews, and user account)
- **Availability:** Medium (Potential to delete records or drop tables)

Technical Analysis

The SQL injection vulnerability exists because:

1. **No Input Validation:** Username field accepts special SQL characters (' , - ,etc.)
2. **No Parameterized Queries:** Application uses string concatenation instead of prepared statements.

Remediation recommendations

To prevent this, the development team should implement the following:

- 1. Use PreparedStatements:** This ensures the database treats the input as data only, not as executable code.
- 2. Input Validation:** Whitelist allowed characters in username field (alphanumeric + underscore only). Reject inputs containing SQL keywords (SELECT, OR, --,etc.)

Conclusion

This penetration test successfully demonstrated a critical SQL injection vulnerability in the OWASP Juice shop authenticated mechanism. The vulnerability allows an unauthenticated attacker to bypass login controls and gain full administrative access using a simple payload. Implementation of parameterized queries and input validation will effectively remediate this issue. This assessment highlights the importance of secure coding practices and the need for regular security testing in application deployment.

Report Status: Complete

Severity Level: High

Remediation: Timeline: Immediate