MINI PROJECT

# Web Application Security Testing

* Name : Akash.K.S
* Date : 3/6/2025
* Task : Web Application Security Testing
* Target Application : OWASP Juice Shop (Online demo)
* Tools Used : Browser, OWASP Juice Shop Demo, Manual testing.

**1. Introduction**

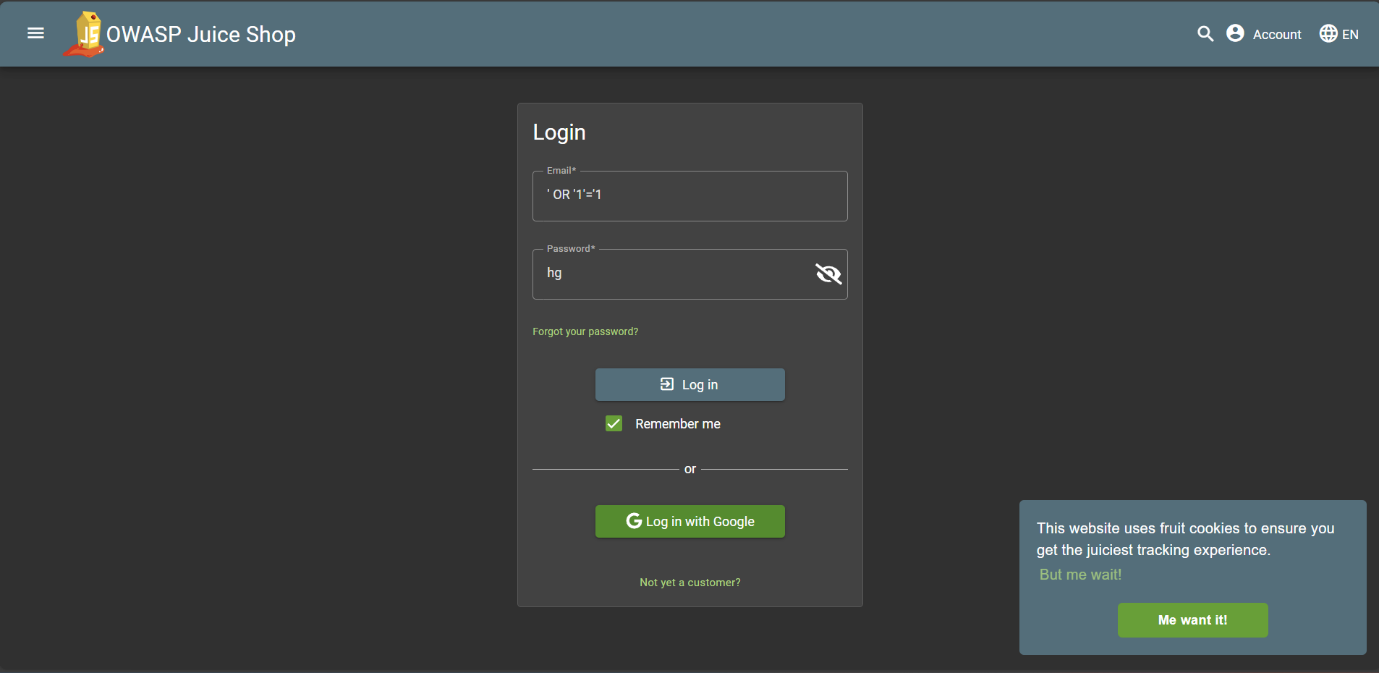
This report documents the security testing performed on the OWASP Juice Shop web application to identify common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Authentication Flaws. The testing was done manually using browser tools and crafted payloads.

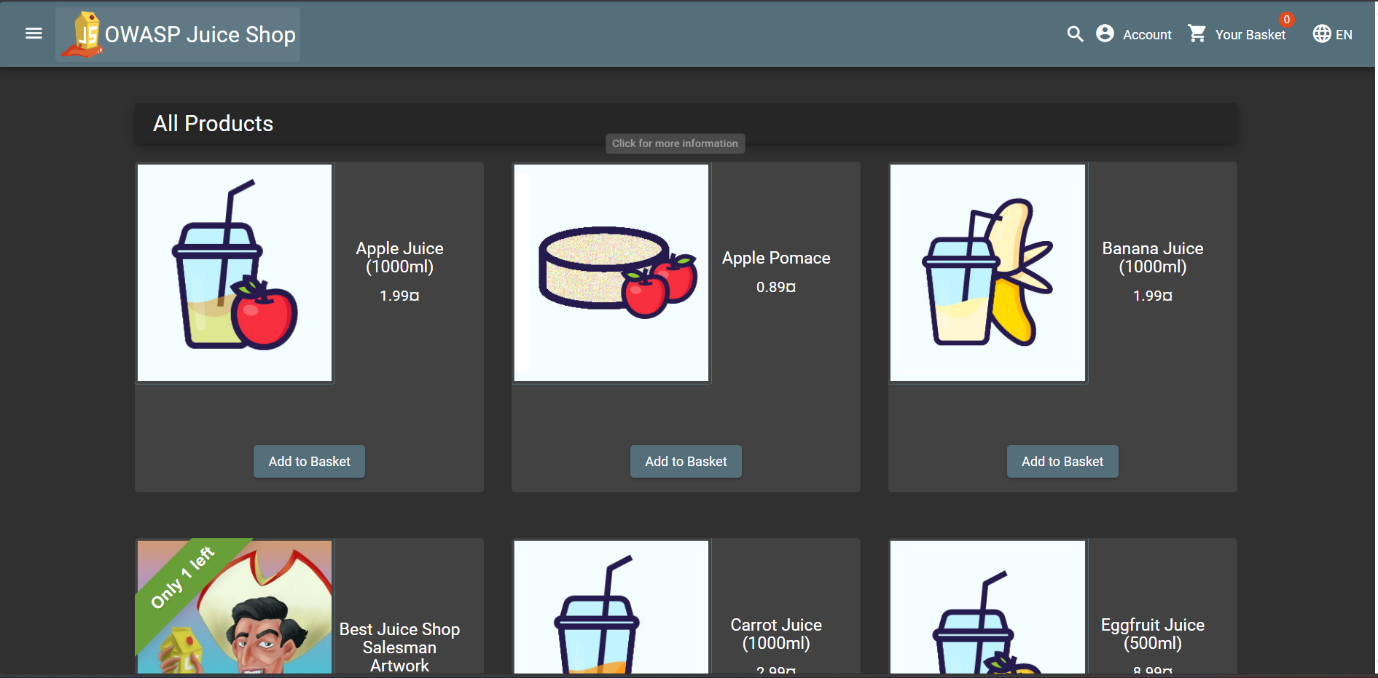
**2. Vulnerability Testing Details**

**2.1 SQL Injection**

* **Test Description:**  
  Tested login form and input fields for SQL Injection by entering payloads to bypass authentication and manipulate SQL queries.
* **Payload Used:**  
  ' OR '1'='1

**Result:**  
Successfully bypassed login/authentication.

******



I have successfully logged into the OWASP Juice shop.

* Impact:

Allows attackers to gain unauthorized access, potentially compromising sensitive data.

* Mitigation:

Use parameterized queries or prepared statements to prevent injection.

**2.2 Cross-Site Scripting (XSS)**

* Test Description:

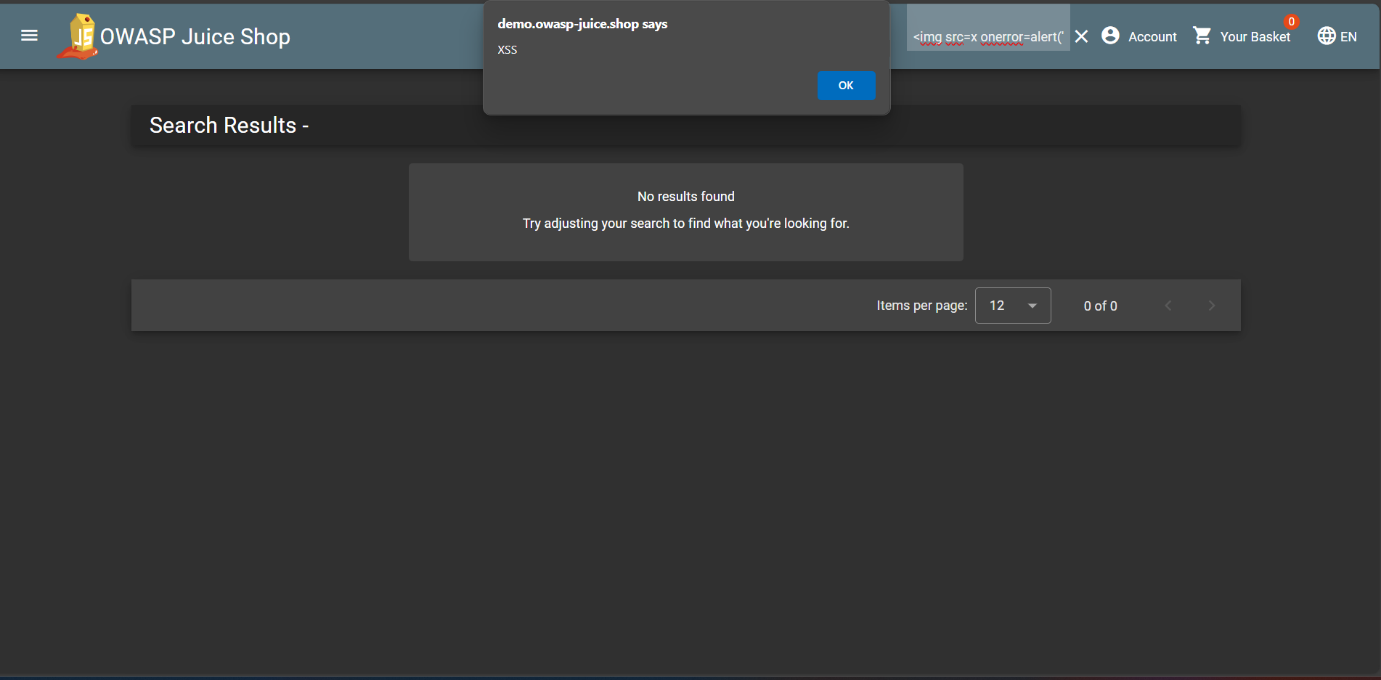
Tested input fields (search bar, feedback form) by injecting JavaScript payloads.

* Payload Used:

<img src=x onerror=alert('XSS')>

* Result:

Payload executed, showing a JavaScript alert box.



* Impact:

Attackers can execute arbitrary scripts, steal cookies, or perform actions on behalf of users.

* Mitigation:

Properly escape output and sanitize input to remove dangerous characters.

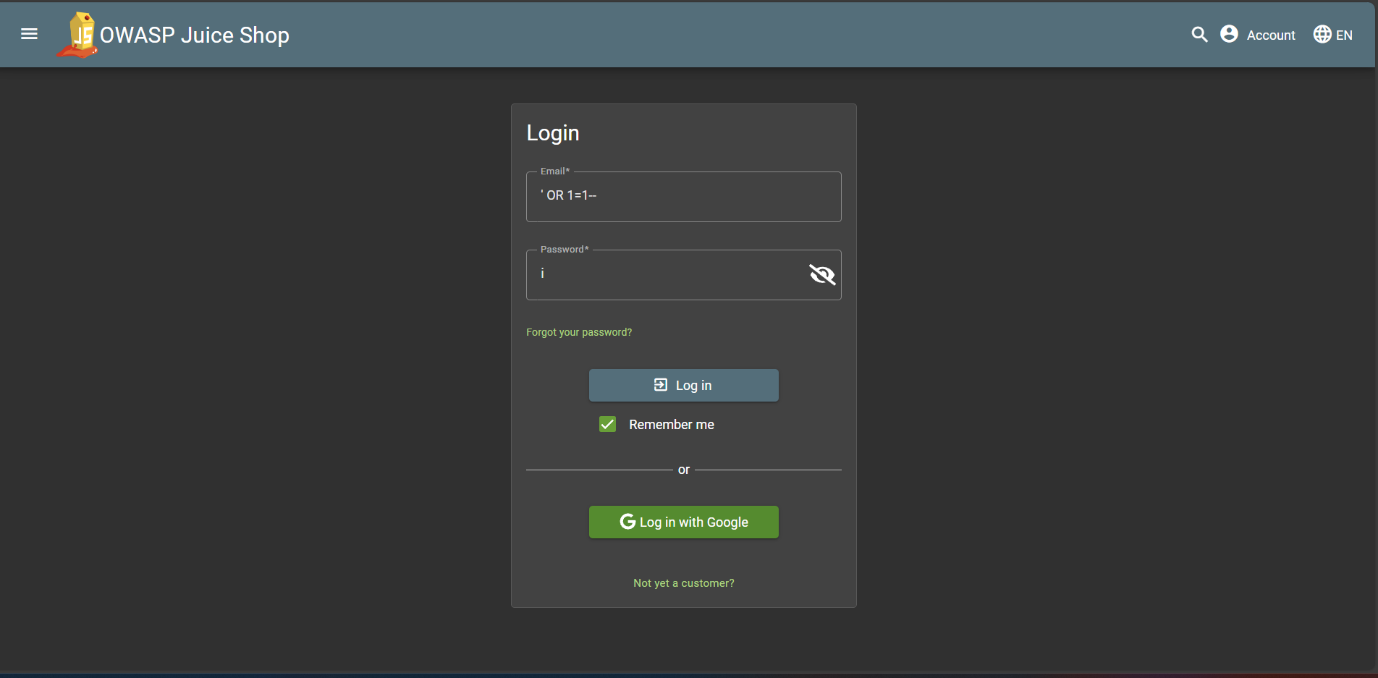
**2.3 Authentication and Access Control**

* Test Description:

Attempted login bypass using default or weak credentials, SQL Injection, and URL manipulation to access restricted pages.

* Result:

No successful login bypass or unauthorized access was observed.



* Impact:

If these flaws existed, they could lead to unauthorized access and privilege escalation.

* Mitigation:

Implement strong password policies, role-based access control, and secure session management.

**3. Tools and Methods Used**

* OWASP Juice Shop online demo
* Manual testing with crafted payloads
* Browser Developer Tools for session and request inspection

**4. Conclusion**

The web application demonstrated vulnerabilities to SQL Injection and Cross-Site Scripting, which could compromise security and user data. No authentication bypasses were found during the testing.

Implementing secure coding practices and robust access control mechanisms is recommended to mitigate these vulnerabilities.

**5. References**

\* OWASP Juice Shop: [https://owasp.org/www-project-juice-shop/](https://owasp.org/www-project-juice-shop/)

\* OWASP SQL Injection: [https://owasp.org/www-community/attacks/SQL\\_Injection](https://owasp.org/www-community/attacks/SQL\_Injection)

\* OWASP Cross-Site Scripting (XSS): [https://owasp.org/www-community/attacks/xss/](https://owasp.org/www-community/attacks/xss/)