Security Assessment Report

Week 1: Security Assessment of Juice Shop Application

Objective:

To perform basic vulnerability assessment on the Juice Shop application by identifying common vulnerabilities and documenting the findings.

1. Application Setup:

- Application: Juice Shop (mock web-based application for security testing)
- Setup Steps:
  - Installed dependencies using npm install
  - Started the application using npm start
  - Accessed the application at http://localhost:3000

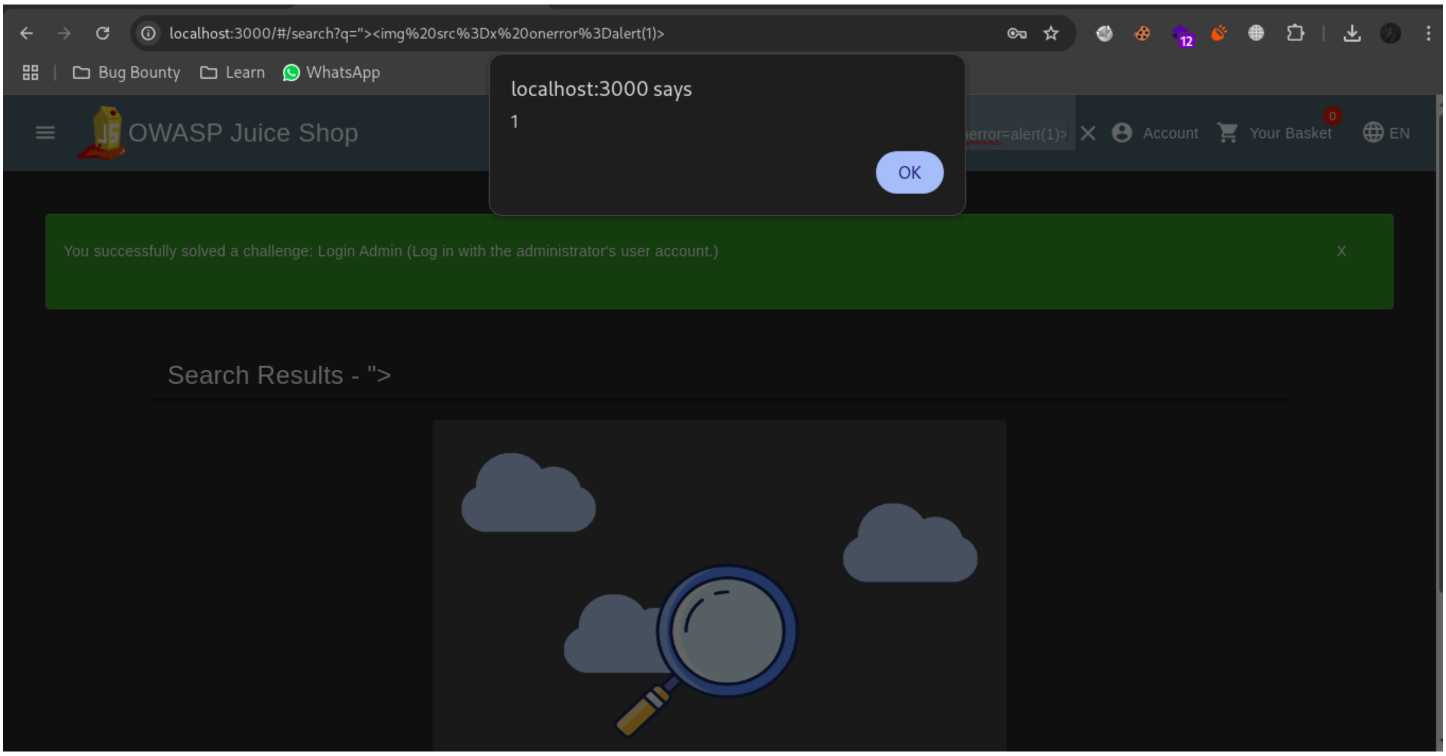
Pages Tested:

- Signup Page
- Login Page
- Profile Page

2. Vulnerability Assessment:

A. Cross-Site Scripting (XSS):

- Tested Area: Search functionality
- Payload Used: "><img src=x onerror=alert(1)>
- Result: The payload was successfully executed, triggering an alert, indicating that the application is vulnerable to reflected XSS attacks.



localhost:3000/#/search?q="><img%20src%3Dx%20onerror%3Dalert(1)>

localhost:3000 says

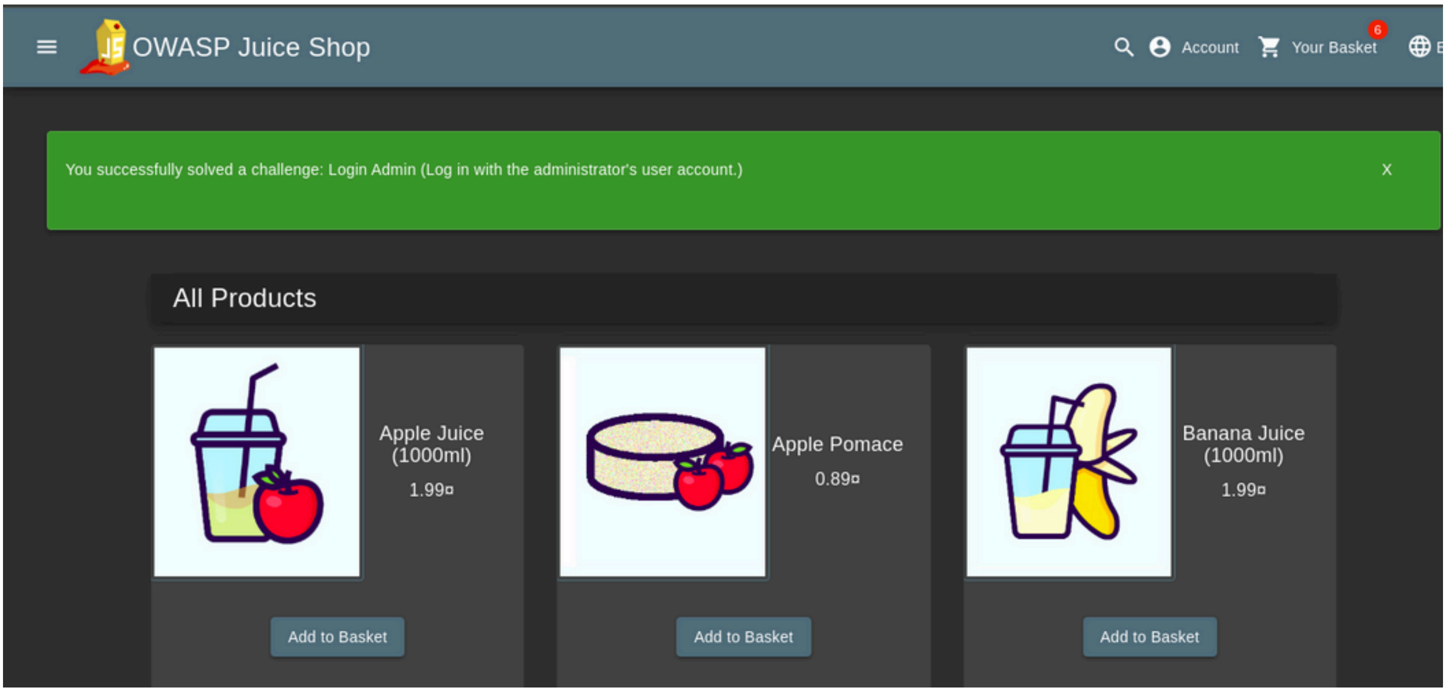
1

You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

Search Results - ">

B. SQL Injection:

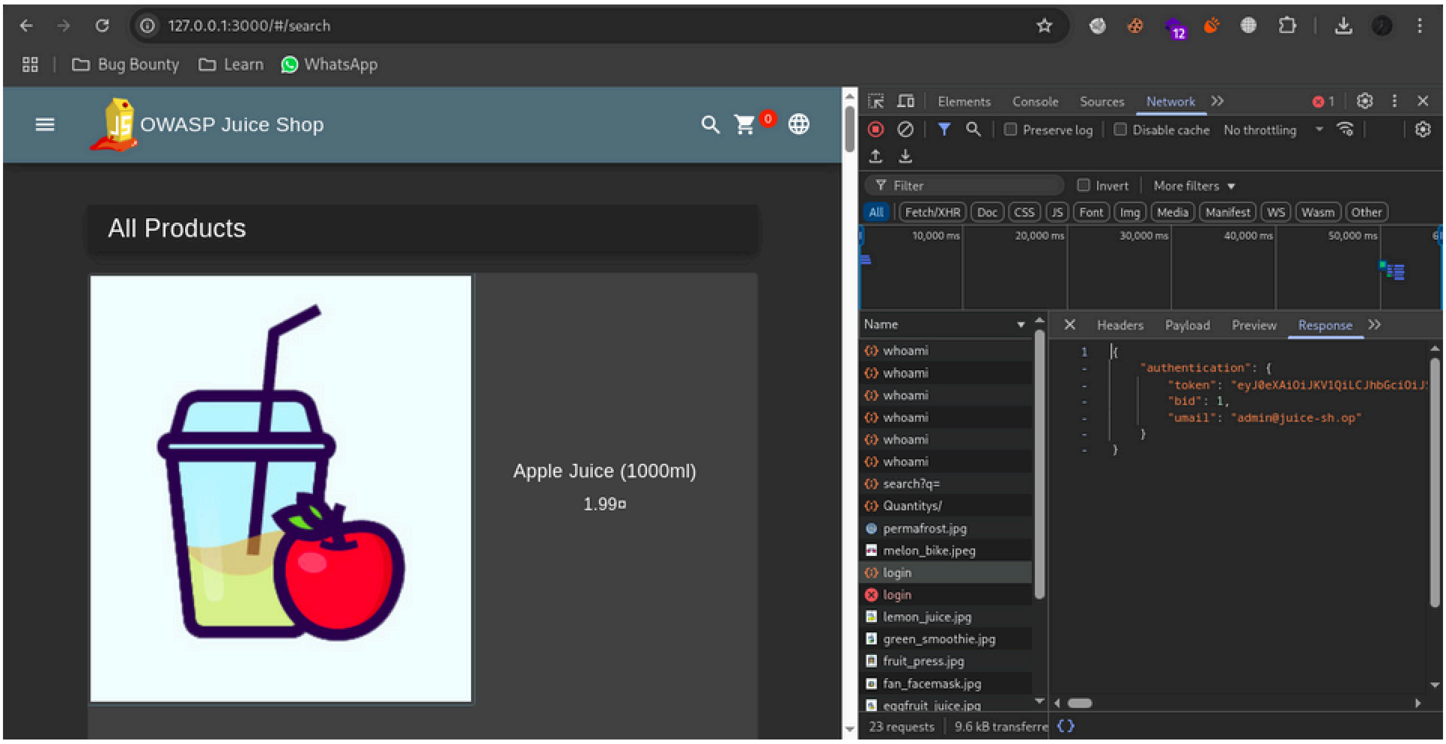
- **Tested Area: Login functionality**
- **Payload Used: ' OR 1=1--**
- **Result: Successfully logged in as the admin user, bypassing authentication via SQL Injection.**



You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

C. Weak Password Storage:

- **Tested Area: Login functionality**
- **Issue Found: Username and password were transmitted in plain text without encryption (visible in the network tab during login).**
- **Impact: Sensitive credentials are exposed, risking interception and unauthorized access.**



```
whoami
whoami
whoami
whoami
whoami
whoami
search?q=
Quantitys/
permafrost.jpg
melon_bike.jpeg
login
login
lemon_juice.jpg
green_smoothie.jpg
fruit_press.jpg
fan_facemask.jpg
eggfruit_juice.jpg
```

```
{
    "authentication": {
        "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJ
        "bid": 1,
        "umail": "admin@juice-sh.op"
    }
}
```

3. Vulnerabilities Found:

- **Cross-Site Scripting (XSS): Reflected XSS vulnerability in the search functionality.**
- **SQL Injection: Authentication bypass using SQL Injection in the login page.**
- **Weak Password Storage: User credentials are transmitted without encryption.**

4. Areas of Improvement:

- **Cross-Site Scripting (XSS): Implement input validation and output encoding to prevent XSS attacks.**
- **SQL Injection: Use parameterized queries or prepared statements to prevent SQL Injection.**
- **Weak Password Storage: Implement HTTPS to secure the transmission of sensitive data and consider encrypting passwords using algorithms such as bcrypt.**