## Exploratory Test Report - Owasp Juice App

**Introduction:**

This exploratory test was conducted on Juice Shop, a vulnerable web application designed for security testing purposes. The goal was to identify potential vulnerabilities and areas for further focused testing.

**Test Date & Time:** Sunday, December 29, 2024 at 9:23 PM PST

**Tester:** Aathiraja

**Test Environment:**

* Browser: Chrome 131.0.6
* Operating System: macOS Sequoia 15.1.1

**Test Scope:**

* Publicly accessible functionalities of the Juice Shop web application.
* Focus on identifying potential security vulnerabilities and usability issues.

**Methodology:**

The exploratory testing followed a free-flowing approach, navigating through different sections of the application and trying various functionalities. The following techniques were used:

* **Equivalence Partitioning:** Testing different valid and invalid inputs for fields like search queries, product quantities, and user registration.
* **Boundary Value Analysis:** Testing input values at the edges of expected ranges (e.g., entering very short or very long usernames during registration).
* **State Transition Testing:** Exploring different application states based on user actions (e.g., logged in vs. logged out, shopping cart empty vs. full).
* **Error Guessing:** Attempting actions that might trigger errors or unexpected behavior.

**Test Findings:**

**Security**

* **Potential SQL Injection:** The search functionality might be vulnerable to SQL injection attacks. By entering special characters or code in the search query field, it could be possible to manipulate the database queries executed by the application. (**Further testing results with SQL injection payloads attached with security test report**)
* **Cross-Site Scripting (XSS):** Text fields like product reviews or user bios might be vulnerable to XSS attacks. Entering malicious scripts in these fields could allow attackers to inject code that executes in the user's browser, potentially stealing data or hijacking sessions.
* **Broken Authentication:** The login and registration process investigated for weaknesses. Techniques like brute-force attacks or password spraying are possible against weak password policies.
* **Session Management:** The application's session management practices need evaluation. Techniques like session hijacking or session fixation might be possible if weaknesses exist. (**Requires testing with session management tools**)

**Usability**

* **Limited Search Functionality:** The search functionality seems basic and might not provide enough filtering options for a large product catalog.
* **Inconsistent Error Messages:** Some error messages are unclear or not user-friendly, making it difficult for users to understand what went wrong.
* **Missing Product Information:** Some product descriptions might be missing important details like ingredients or nutritional information.

**Further Exploration:**

* Test the functionality of the shopping cart and checkout process.
* Explore user account management features and profile settings.
* Investigate the use of cookies and other tracking mechanisms.
* Conduct performance testing to assess the application's responsiveness under load.

**Disclaimer:**

This exploratory test is a starting point for further security testing. Reports of the findings are attached as bug reports and security test reports. The identified vulnerabilities are potential and require further confirmation and exploitation attempts using appropriate tools and techniques.

**Recommendations:**

* Implement proper input validation and sanitization to prevent SQL injection and XSS attacks.
* Enforce strong password policies and consider two-factor authentication for improved login security.
* Review session management practices to mitigate risks of session hijacking or fixation.
* Improve search functionality and provide more comprehensive product information.
* Enhance error messages to be clear, actionable, and user-friendly.