

**Web Application Pentest Report**

**On**

**E-Commerce Website**

**(OWASP Juice Shop)**

**Report By**: Anamay Mishra

**Report Date**:13-Febuary-2024

**Executive Summary**

This report hereby describes the proceedings and results of a Black Box Security Assessment conducted against an **E-Commerce Web Application (OWASP Juice Shop).** The report hereby lists the findings and corresponding best practice mitigation actions and recommendations**.**

**Disclaimer**

Note that this assessment may not disclose all vulnerabilities that are present on the web application within the scope of the engagement. This report is a summary of the findings based on **OWASP Top 10** Security Vulnerabilities. Any changes made to the environment during the period of testing may affect the results of the assessment.

**Scope**

|  |  |
| --- | --- |
| Application Name | OWASP Juice Shop |
| URL | localhost:3000 |

**Methodology**

The framework used for the pen testing is ***Penetration Testing Execution Standard (PTES) i.e. OWASP Testing Framework.*** It defines penetration testing as 7 phases. Particularly, PTES Technical Guidelines give hands-on suggestions on testing procedures, and recommendations for security testing tools.

The 7 phases are: -



**Assessment Findings**

It consists of OWASP top 10 common vulnerabilities which may lead to potential data loss.

|  |  |  |
| --- | --- | --- |
| **S. No** | **Vulnerability Name** | **Page No** |
| 1 | IDOR (Path Traversal) | 5 |
| 2 | SQLi to Bypass Authentication | 6 |
| 3 | Reflected XSS | 7 |

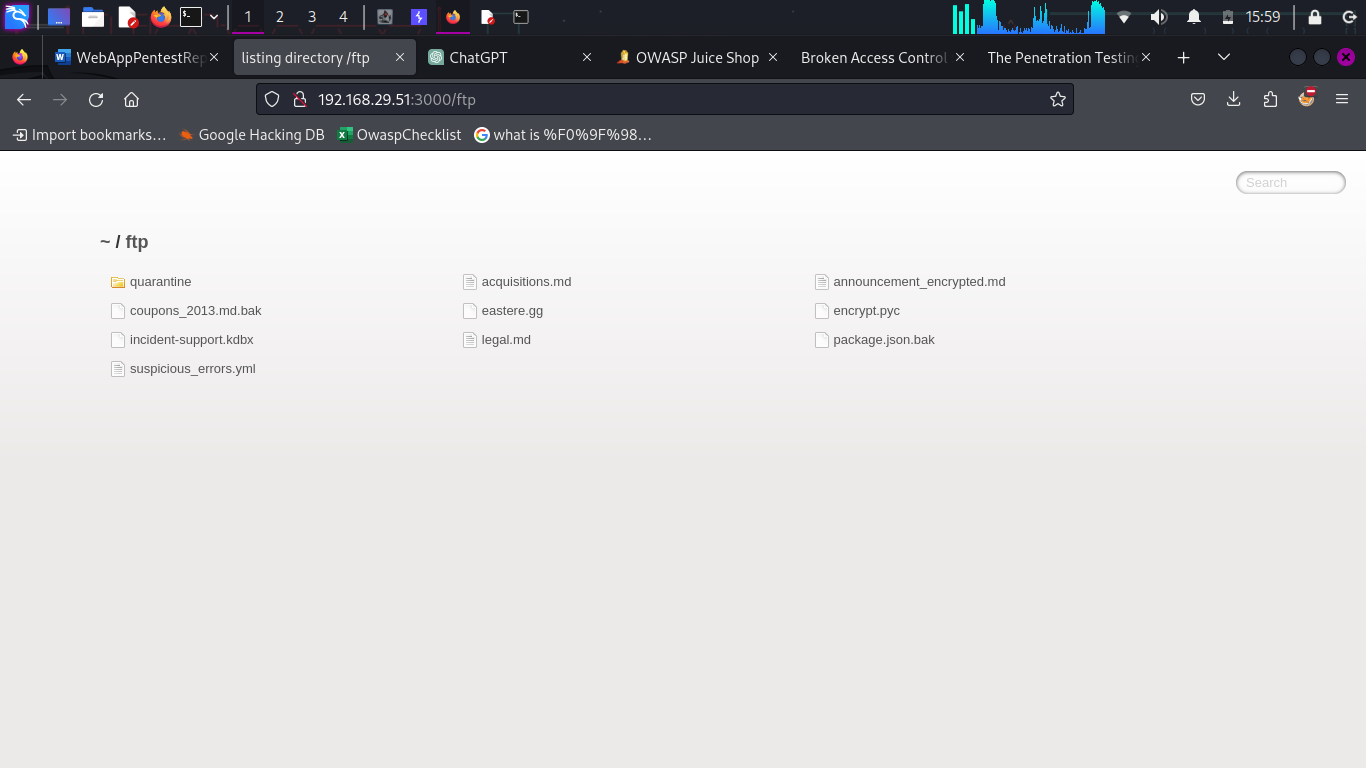
## **Vulnerability Findings**

1. **Path Traversal (Critical)**

|  |  |
| --- | --- |
| **Description:** | Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data. |
| **Impact:** | Critical |
| **System:** | Localhost:3000/ftp |
| **References:** | https://pwning.owasp-juice.shop/companion-guide/latest/part2/broken-access-control.html |

**Exploitation Proof of Concept**

We gathered that /ftp page being listed in the “robots.txt” file which disallowed /ftp to be listed by google. The localhost:3000/ftp does not have any access control which then leaked many confidential documents.



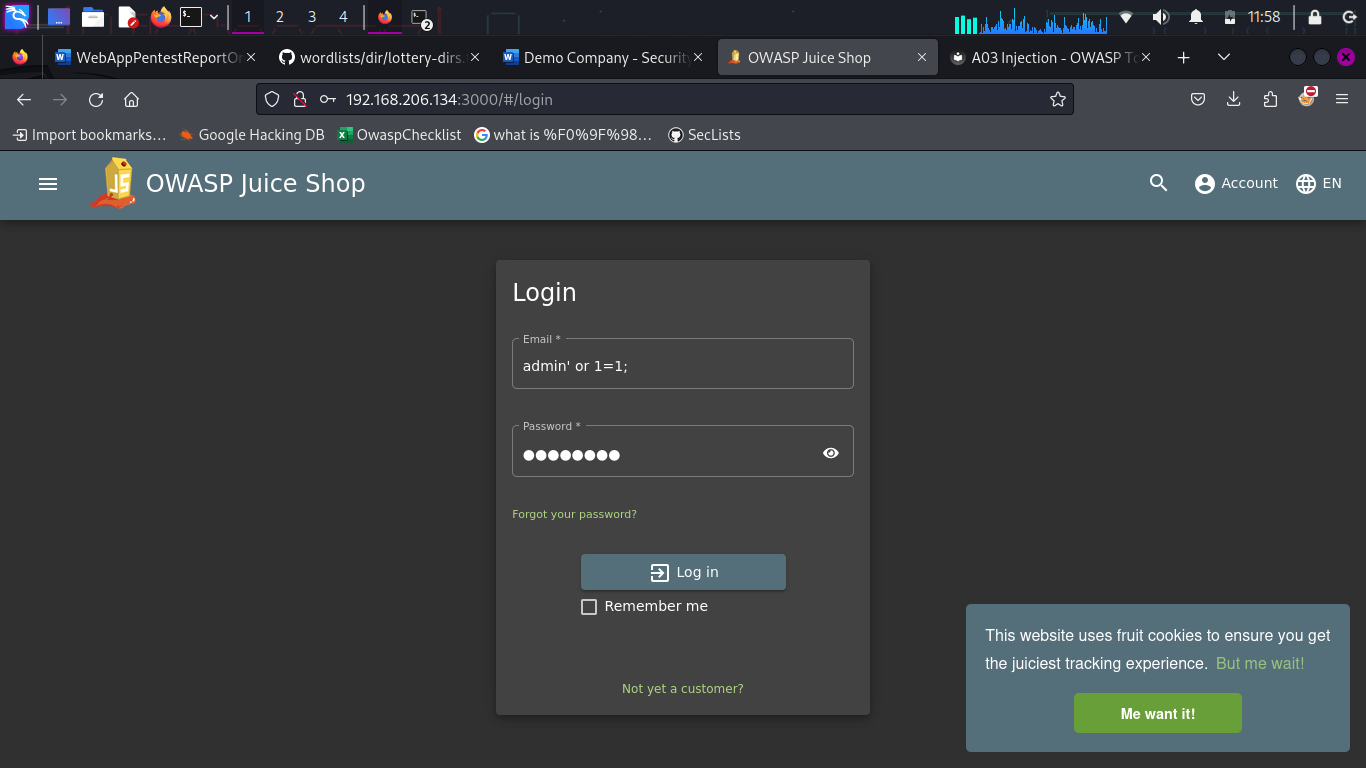
*Figure 1: List of confidential documents*

1. **SQL Injection to bypass authentication (High)**

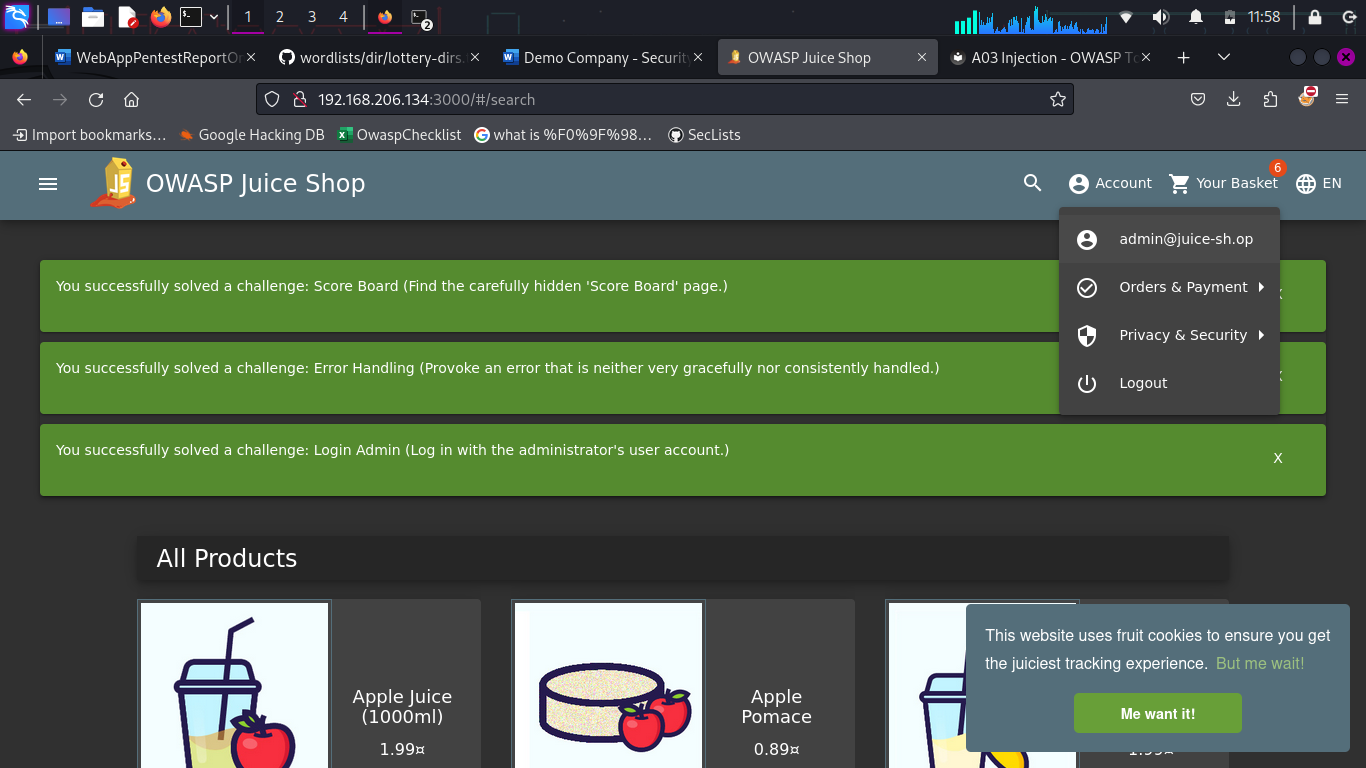
|  |  |
| --- | --- |
| **Description:** | SQL injection, also known as SQLI, is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed. |
| **Impact:** | High |
| **System:** | localhost:3000/#/login |
| **References:** | https://portswigger.net/support/using-sql-injection-to-bypass-authentication |

**Exploitation Proof of Concept**

We gathered that login page is vulnerable to SQL injection which bypasses the admin authentication. The payload “admin’ or 1=1;” as username and “password” as password is not handled/validated by the backend.



*Figure 1: SQL Injection attack*



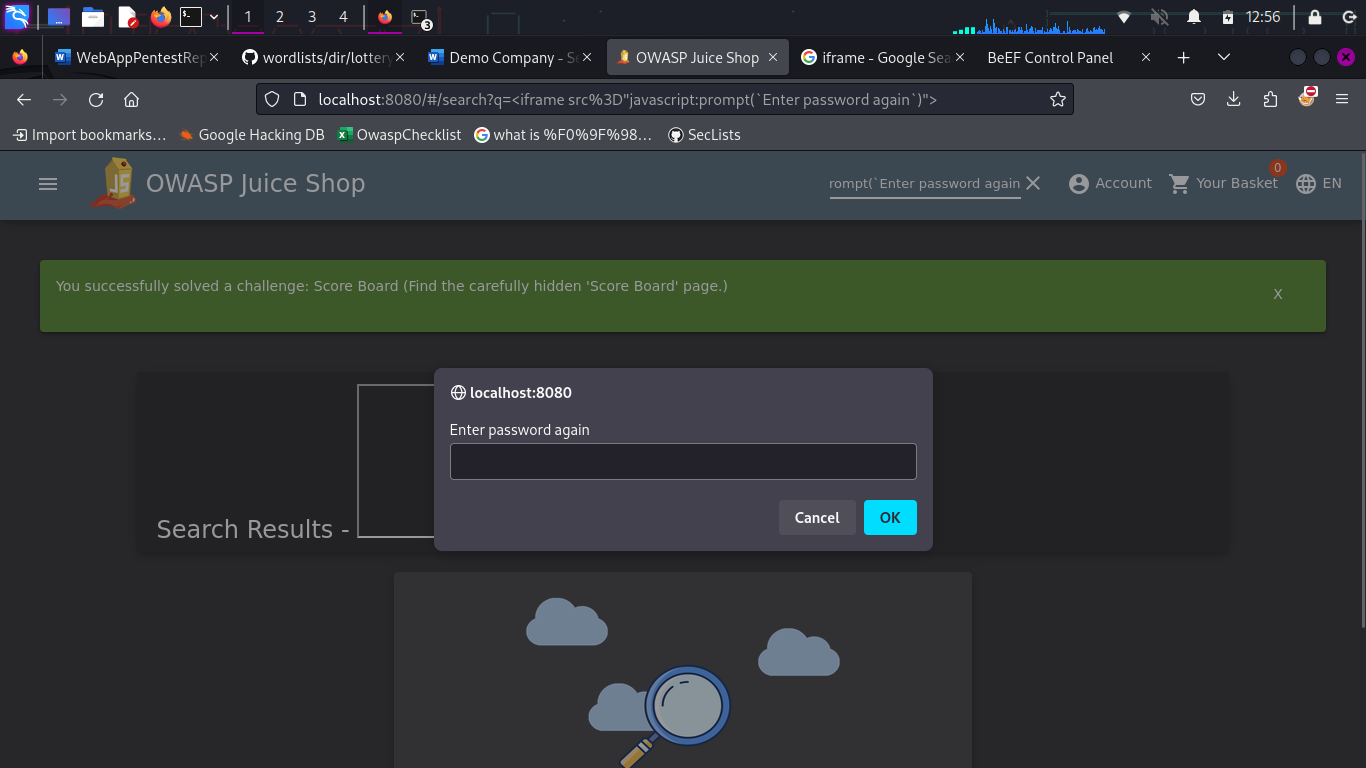
*Figure 2: Admin authentication bypassed.*

1. **Reflected Cross Site Scripting (Medium)**

|  |  |
| --- | --- |
| **Description:** | Reflected XSS attacks, also known as non-persistent attacks, occur when a malicious script is reflected off of a web application to the victim's browser |
| **Impact:** | Medium |
| **System:** | localhost:3000/#/search |
| **References:** | https://portswigger.net/web-security/cross-site-scripting/reflected |

**Exploitation Proof of Concept**

We gathered that the web application is vulnerable to a Reflected XSS attack which in turn enters malicious script on behalf of victim machine. The malicious code is inserted through search field i.e. “<iframe src="javascript:prompt(`Enter password again`)">”



*Figure 1: Reflected XSS attack*