**Cybersecurity Internship Report (Week 1)**

**Security Evaluation & Testing Report**

**Prepared By:**   
**Mayra Ahmer**  
Intern ID: DHC3529  
Email: [mayrahamer184@gmail.com](mailto:mayrahamer184@gmail.com)

**Table of Contents**

1. Overview
   * Purpose of the Assessment
2. System Configuration
   * Required Tools Installation
   * Deploying OWASP Juice Shop
   * Verifying Installation
3. Tools Utilized
4. Security Testing Approach
   * Automated Scanning (OWASP ZAP)
   * Manual Vulnerability Testing
5. Findings & Observations
6. Conclusion
7. References

**1. Overview**

This report presents the security assessment of the OWASP Juice Shop application, a deliberately vulnerable platform used for penetration testing and security research. The objective was to uncover security flaws by combining automated scanning with manual penetration testing, conducted within a controlled Kali Linux environment.

**Purpose of the Assessment:**

* **System Configuration:** Install and deploy the OWASP Juice Shop application on a local machine. This setup includes tools such as Git, Node.js, and npm.
* **Automated Security Scan:** Use OWASP ZAP to detect vulnerabilities like missing security headers and Cross-Site Scripting (XSS).
* **Manual Security Testing:** Simulate attacks using developer tools to identify flaws in input validation and authentication mechanisms.
* **Findings Documentation:** Record vulnerabilities, analyze risks, and suggest countermeasures.

**2. System Configuration**

Before conducting security assessments, it was essential to establish a testing environment using Kali Linux and the necessary tools.

**A. Required Tools Installation**

To prepare for testing, the following dependencies were installed:

* **Git** - To clone the Juice Shop repository.
* **Node.js & npm** - Required for running Juice Shop.

Installation Commands:

sudo apt update

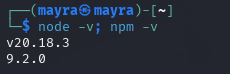
sudo apt install git nodejs npm –y  
  
  




To verify successful installation:

node -v # Check Node.js version

npm -v # Check npm version



**B. Deploying OWASP Juice Shop**

Juice Shop was downloaded and executed as follows:

git clone https://github.com/juice-shop/juice-shop.git

cd juice-shop

npm install

npm start





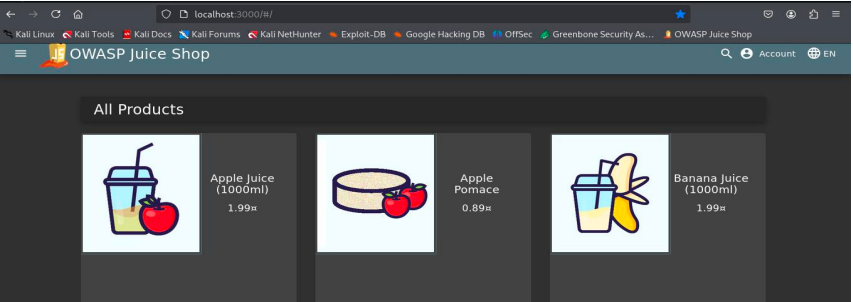




**C. Verifying Installation**

The setup was confirmed using:

* **Terminal Output:** Ensured no errors while launching Juice Shop.
* **Browser Verification:** Opened http://localhost:3000 to confirm access.



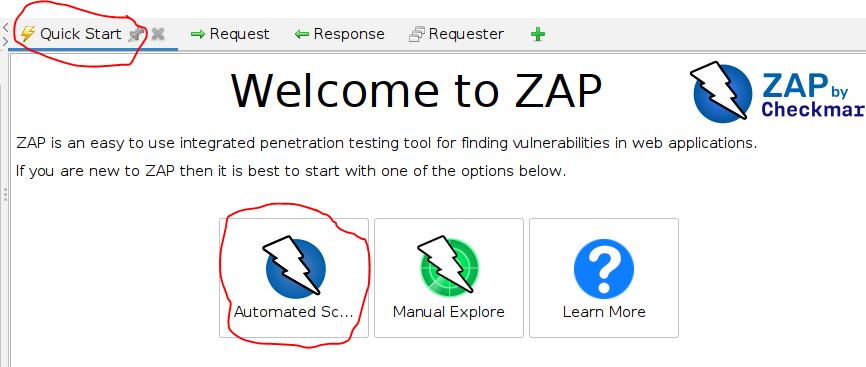
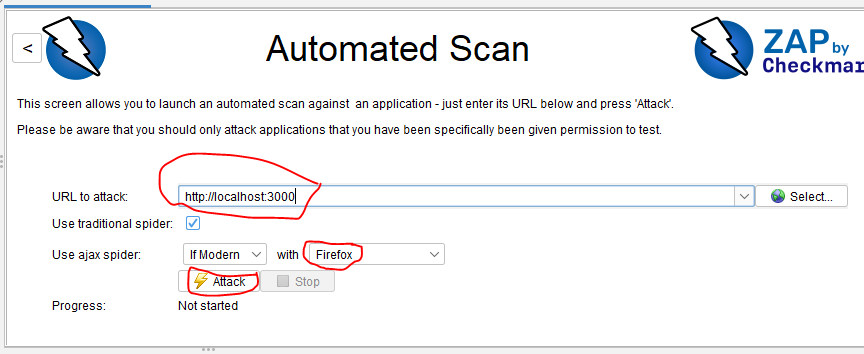
**3. Tools Utilized**

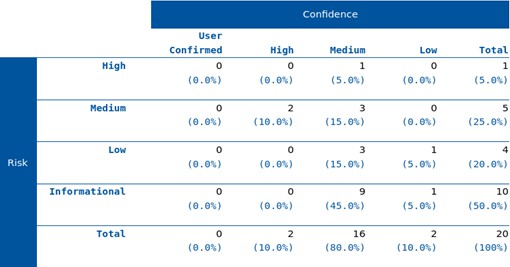
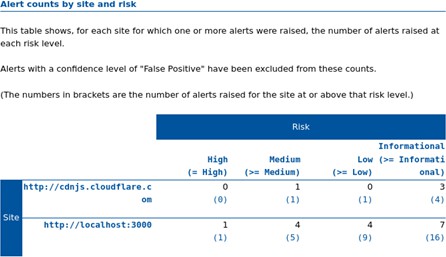
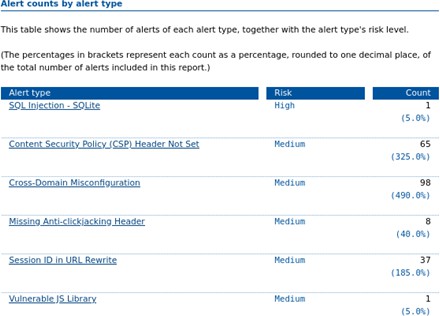
* **OWASP ZAP:** Conducted automated vulnerability scans.
* **Browser Developer Tools:** Used for manual testing (XSS & SQL Injection tests).

**4. Security Testing Approach**

**A. Automated Scanning with OWASP ZAP**

* The target URL (http://localhost:3000) was set.
* A full scan was executed to detect security weaknesses.
* Vulnerabilities such as XSS and missing headers were logged.

Result:  
  


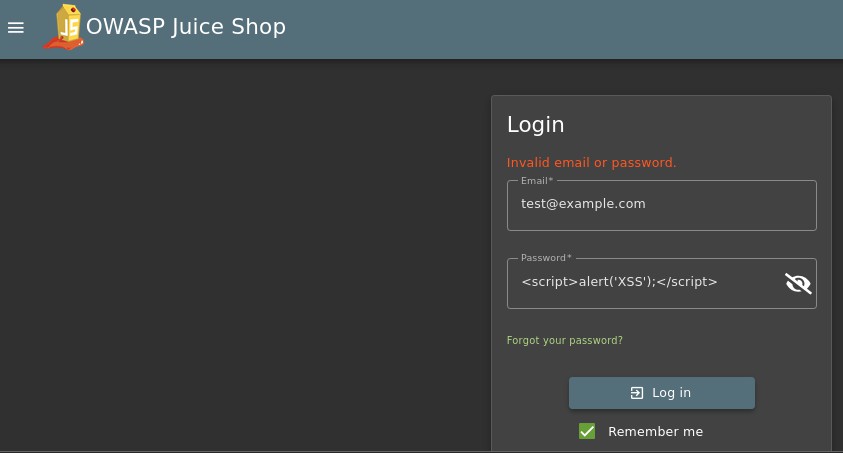
**B. Manual Vulnerability Testing**

**I. XSS Testing**

Test cases:

* **Email Field Injection:** test@example.com
* **Password Field Injection:** <script>alert('XSS');</script>

**Expected Outcome:** If vulnerable, an alert box should appear.  
**Actual Outcome:** No alert was triggered, indicating proper input handling

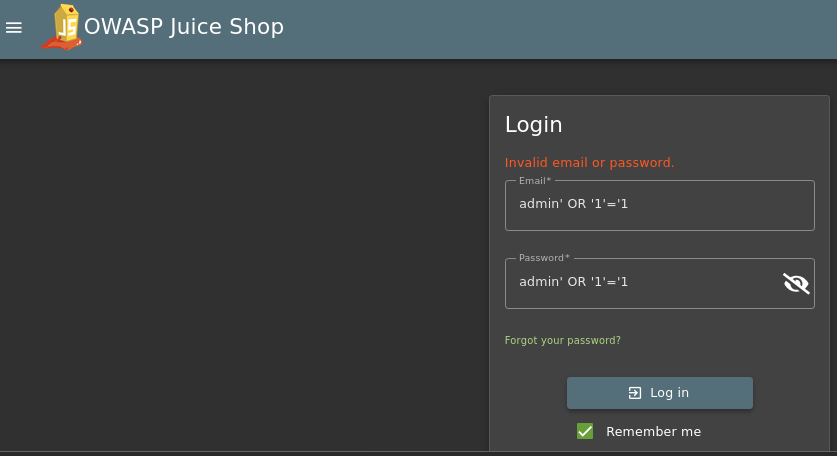
****

**II. SQL Injection Testing**

Test case:

* **Login Bypass Attempt:** admin' OR '1'='1 in both email and password fields.

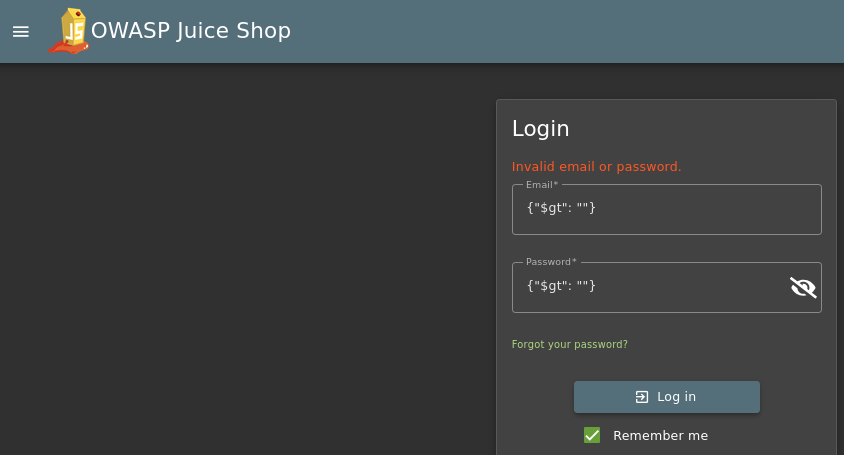
**Expected Outcome:** If vulnerable, authentication should bypass.   
**Actual Outcome:** The login attempt failed, indicating security against SQL Injection.



**III. NoSQL Injection Testing**

Test case:

* **NoSQL Injection Payload:** { "$gt": "" }

**Expected Outcome:** Unauthorized access.   
**Actual Outcome:** Login attempt rejected, indicating NoSQL injection protection.

**5. Findings & Observations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Type** | **Payload Used** | **Observed Result** | **Conclusion** |
| SQL Injection | admin' OR '1'='1 | Login failed | Secure against SQL |
| NoSQL Injection | { "$gt": "" } | Login failed | Secure against NoSQL |
| XSS Injection | <script>alert('XSS');</script> | No alert box | Secure against XSS |

**6. Conclusion**

The security assessment of the OWASP Juice Shop login form revealed no vulnerabilities for SQL Injection, NoSQL Injection, or basic XSS attacks. These results suggest that proper security measures are in place. However, further assessments using more advanced attack vectors across different application components are recommended.

**7. References**

* [OWASP ZAP Documentation](https://www.zaproxy.org/)
* [OWASP Juice Shop GitHub Repository](https://github.com/juice-shop/juice-shop)