Task 1

Perform a Vulnerability Assessment of a Sample Web Application

Vulnerability Assessment Report : -

Tool Used : Burp Suite

- **Burp Suite** is a tool used for testing the security of web applications.
- It works by intercepting and analyzing the communication between your browser and a website.
- This allows security testers to check if the website has any vulnerabilities, like **SQL injection** or **Cross-Site Scripting (XSS)**.
- Burp Suite helps you manually test, modify requests, and even run automated scans (in paid versions) to find potential security issues.

---

1. Introduction : -

- This report summarizes the findings from a vulnerability assessment conducted on the OWASP Juice Shop application using Burp Suite Community Edition. The goal was to identify common web application vulnerabilities, focusing on SQL Injection (SQLi) and Cross-Site Scripting (XSS). This assessment was carried out through manual testing using Burp Suite's Proxy, Repeater, and Intruder tools.

Goal:

- Perform a basic vulnerability assessment of **OWASP Juice Shop** using Burp Suite Community Edition, focusing on manual testing for vulnerabilities like **SQL Injection (SQLi)** and **Cross-Site Scripting (XSS).**

---

2. Methodology :-

The assessment followed the below steps:

1. **Proxy Setup**: Burp Suite was configured as a proxy to intercept traffic between the web browser and the OWASP Juice Shop application.
2. **Manual Testing**: Focused on common web vulnerabilities such as SQL Injection and XSS by manipulating HTTP requests and inspecting responses.
3. **Tools Used**:
    - **Burp Proxy**: Intercepting and modifying HTTP requests.
    - **Burp Repeater**: Sending modified requests to test for vulnerability responses.

- **Burp Intruder**: Automated testing of parameters to detect potential injection points (used for SQLi).

Manual testing was used due to the limitations of **Burp Suite Community Edition**, which lacks the automated scanning capabilities available in the Pro version.

---

3. Findings :-

3.1 Cross-Site Scripting (XSS)

Vulnerability Type: Reflected XSS
Location: Search input (parameter q)
Description:
During testing, the q parameter, which is used for search functionality, was found to be vulnerable to reflected XSS. The input is reflected back to the user without proper sanitization, allowing the execution of arbitrary JavaScript in the user's browser.

Steps :

1. Intercept the HTTP request to the search page using Burp Suite Proxy.
2. Modify the q parameter to inject the payload: <img src="x" onerror="alert('XSS')">
3. Send the modified request using Burp Suite's Repeater.
4. Observe the response – an alert box with the message **XSS** is triggered, confirming the presence of a reflected XSS vulnerability.

Impact :
An attacker could exploit this vulnerability to execute arbitrary JavaScript in the context of a user's session, potentially stealing cookies or performing actions on behalf of the user.

Recommendation :
Ensure that all user inputs are properly sanitized and validated on the server side. JavaScript should not be executed directly from user input fields. Use an HTML escaping library to prevent scripts from executing.

3.2 SQL Injection (SQLi)

Vulnerability Type: SQL Injection (SQLi)
Location: Login form (username and password parameters)
Description:
Manual testing revealed a potential SQL Injection vulnerability in the login form. When injecting typical SQLi payloads into the username and password fields, the application did not sanitize the input, and the server returned an error indicating that the input was not properly escaped.

Steps :

1. **Intercept the login request using Burp Suite Proxy.**
2. **Modify the username parameter to inject an SQL payload : ' OR 1=1 --**
3. **Modify the password parameter with any random value.**
4. **Send the request via Burp Suite's Repeater.**
5. **Observe the response: The login page should either bypass authentication or show a database error message, indicating that the SQL query was manipulated successfully.**

Impact:
An attacker could exploit this vulnerability to gain unauthorized access to the application or retrieve sensitive data from the database.

Recommendation:
Use prepared statements and parameterized queries to prevent SQL injection. Ensure that all user inputs are validated and sanitized before being used in SQL queries.

---

4. Conclusion : -

- The assessment of **OWASP Juice Shop** identified two critical vulnerabilities: **Reflected XSS** and **SQL Injection**.
- Both vulnerabilities can be exploited by an attacker to perform actions such as stealing sensitive information or bypassing authentication.
- While these vulnerabilities are common in web applications, they can be mitigated with proper input validation, parameterized queries, and appropriate security headers.

---

5. Screenshots : -

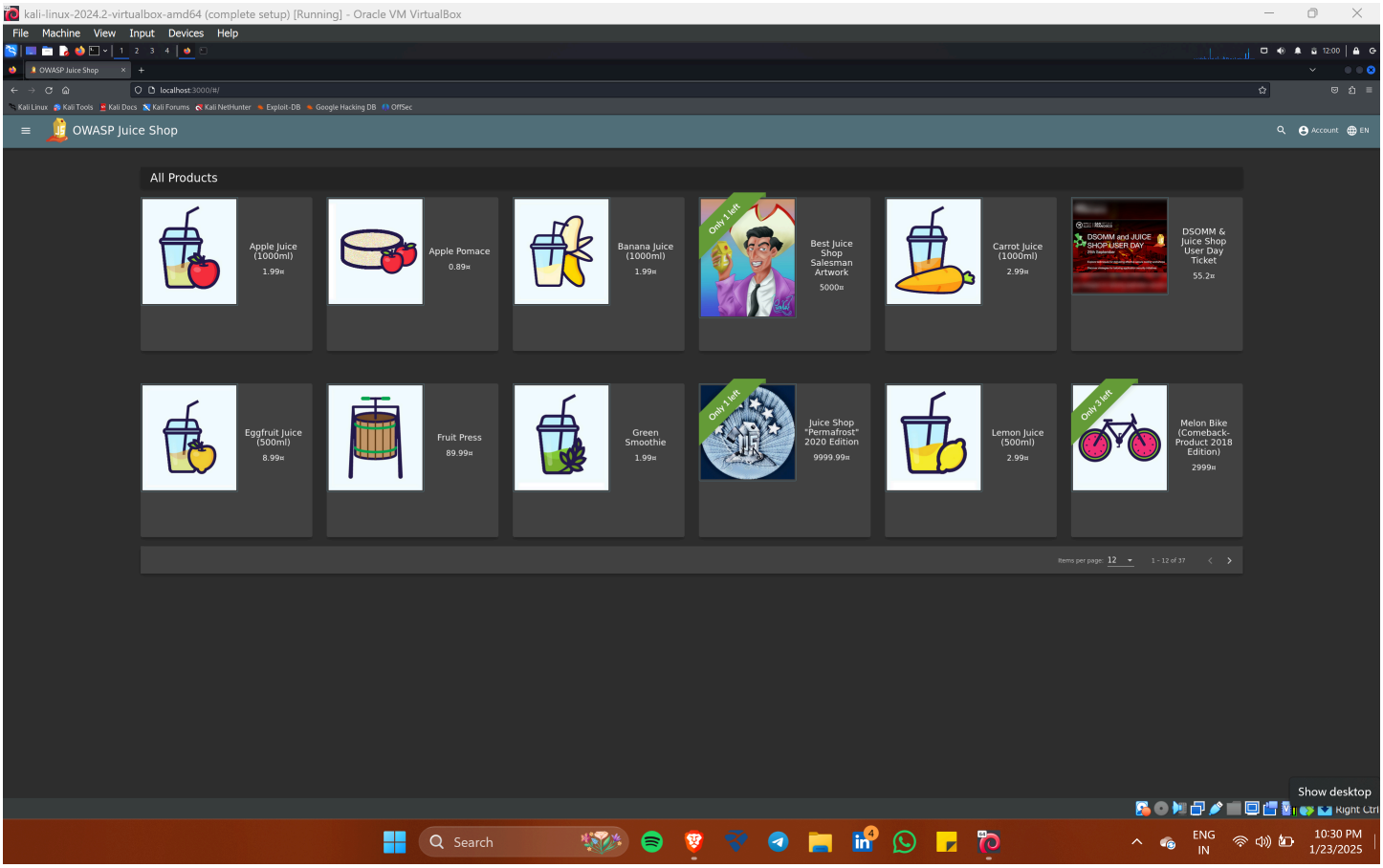

Figure - 1 : OWASP JUICE SHOP

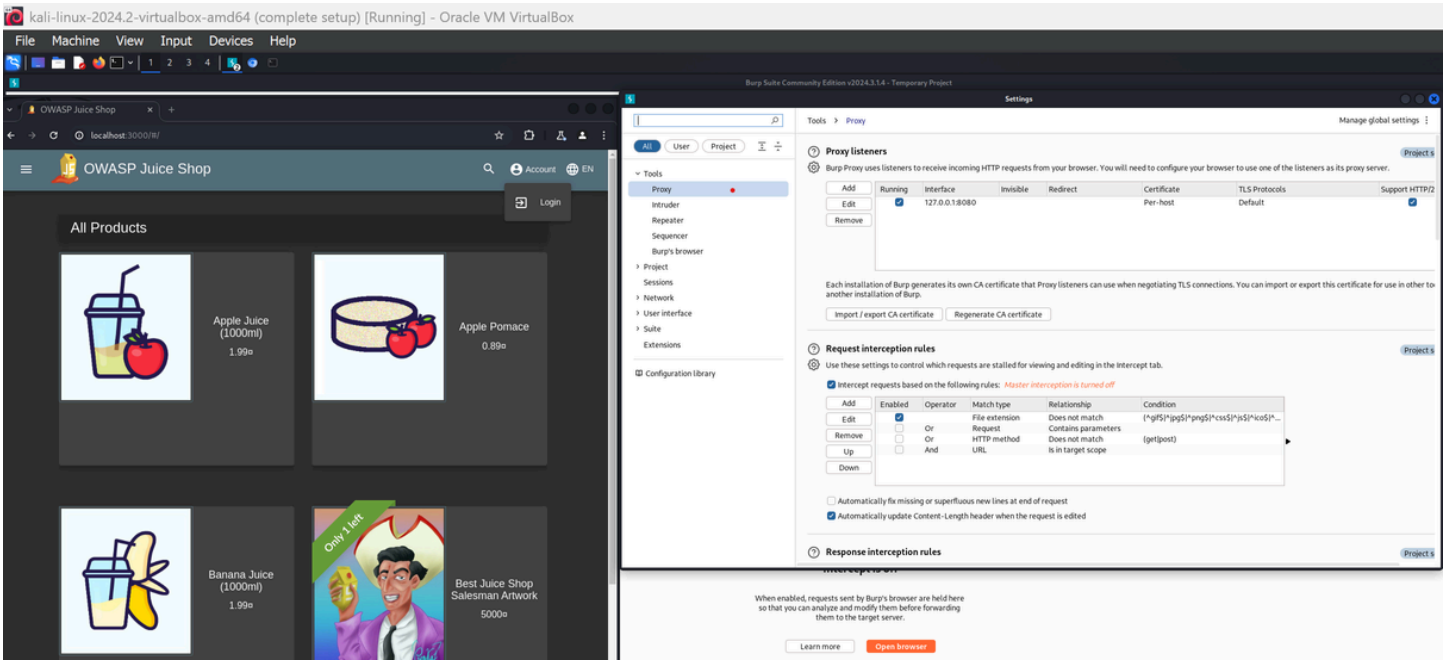

Figure - 2 : Burp Suite and Browser Proxy Setup

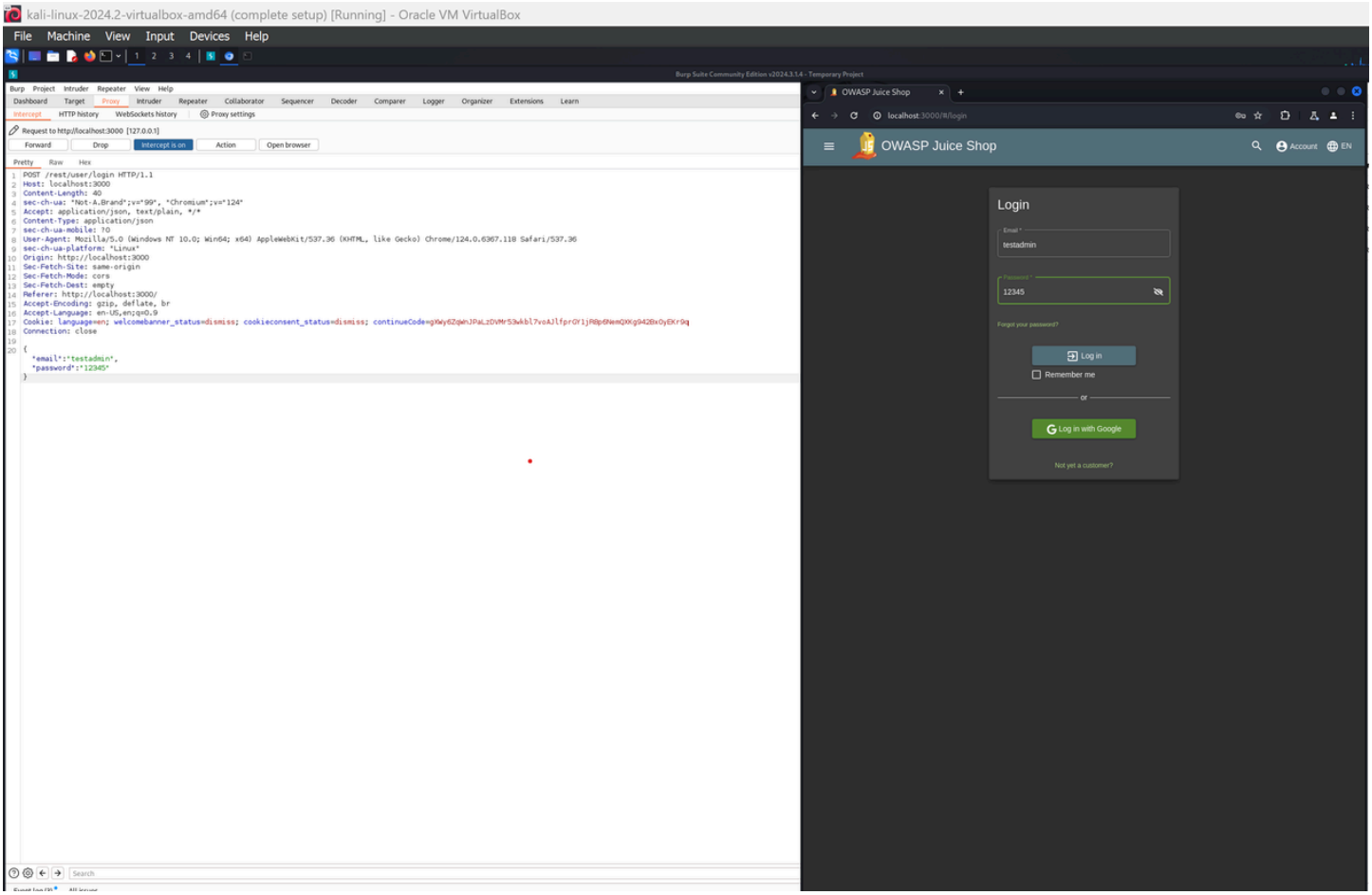

Figure - 3 : Intercepting Login Request

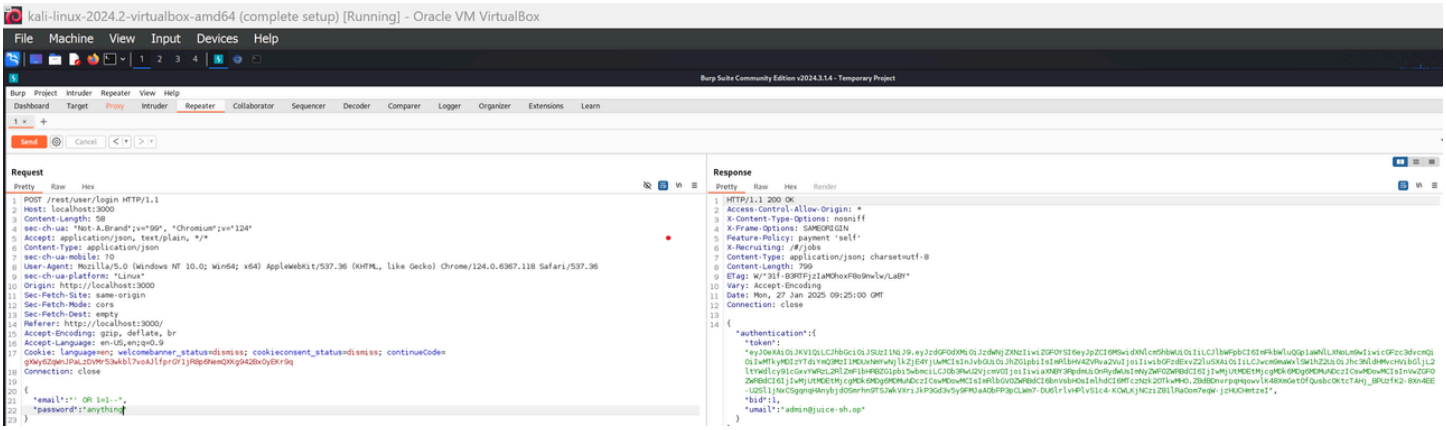

Figure - 4 : Manual Testing using Repeater - SQLi

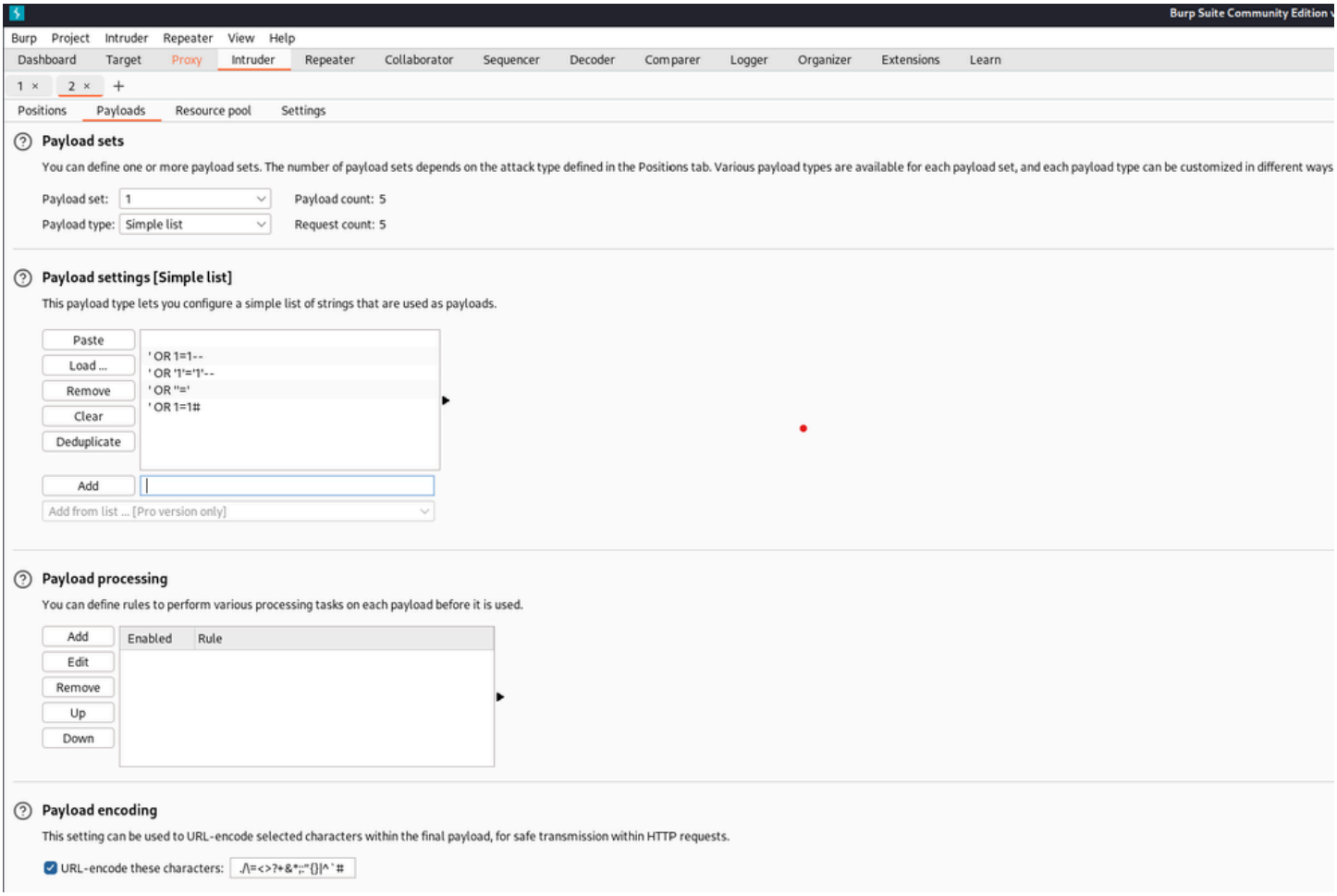

Figure - 5 : Payload Sub Tab

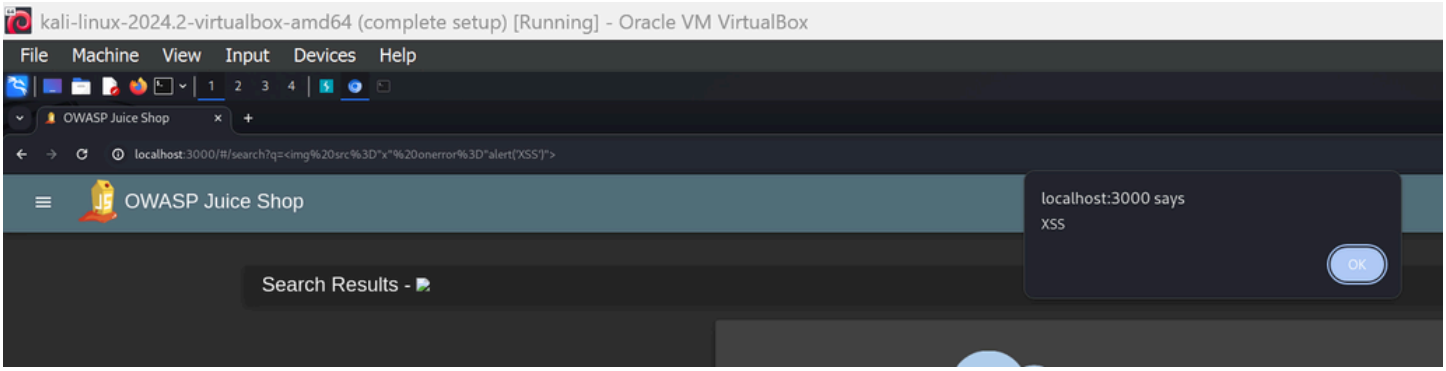

Figure - 6 : Reflected XSS Popup box