

TASK 1: Web Application Security Testing Report

Intern Name: B Haritheertha

Test Application: OWASP Juice Shop

Tools Used: Burp Suite, Chrome Developer Tools

Date: 16 July 2025

Introduction

This report documents the vulnerability assessment performed on the OWASP Juice Shop application as part of the Cybersecurity Internship Task 1 with Future Interns. The goal was to identify and document common web application security issues using ethical hacking tools and OWASP Top 10 standards.

Vulnerability 1: SQL Injection in Login Form

The login form accepts SQL statements, allowing bypass of authentication.

Payload Used: ' OR 1=1 -

impact: Allows unauthorized access to admin accounts.

OWASP Category: A01:2021 – Broken Access Control

Fix Recommendation: Use parameterized queries or ORM frameworks to prevent SQL injection.

Vulnerability 2: Stored/Reflected XSS

The search input reflects unsanitized JavaScript, triggering alerts.

Payload Used: <iframe src="javascript:alert('xss')">

Impact: Can execute malicious scripts on users' browsers.

OWASP Category: A03:2021 – Injection

Fix Recommendation: Use output encoding and input sanitization

The screenshot shows a browser window for 'OWASP Juice Shop' with a search results page. A JavaScript alert dialog box is overlaid, containing the text '=javascript:alert(%60%0a%60%)'. The developer tools are open, showing the DOM structure of the page, which includes various Angular components like 'mat-sidenav' and 'mat-drawer'.

This screenshot shows a browser window with a tab titled 'juice-shop.herokuapp.com/#/search?q=<iframe%20src%3D%27javascript:alert%27%0a%27%27)'. The content area displays a search results page with a large blacked-out search input field. The developer tools show the DOM structure, including an 'iframe' element with the src attribute set to 'javascript:alert'.

Vulnerability 3: Sensitive File Disclosure (FTP)

A .md file was accessible via FTP endpoint.

Impact: Revealed internal planning and confidential acquisition plans.

OWASP Category: A06:2021 – Security Misconfiguration

Fix Recommendation: Remove unused routes, deny access to internal documentation, and use access controls.

The screenshot shows the Burp Suite interface with a list of captured requests. One request is highlighted, showing a detailed view of the 'Request' and 'Response' sections. The 'Request' section shows a GET request to '/api/Challenges/name/Score20...'. The 'Response' section shows the raw response content, which includes sensitive information such as 'password: "self"', 'Feature-Policy: payment "self;"', and 'Last-Modified: Tue, 18 Jul 2025 12:47:16 GMT'.

Vulnerability 4: Insecure Session Token Exposure

The authentication token is stored in a cookie without the HttpOnly flag, making it accessible through JavaScript in the browser. This is visible in the Application → Cookies tab of developer tools.

Impact: If an attacker is able to inject JavaScript (e.g., via XSS), they can steal the session token and impersonate the user.

Observed:

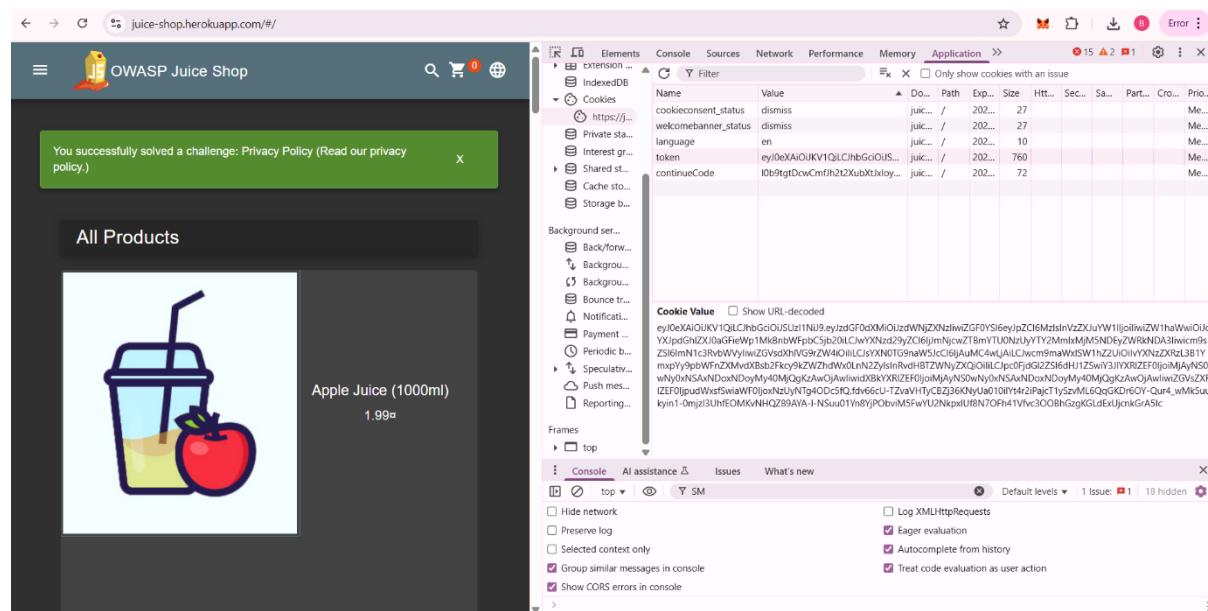
- ◆ token cookie is present.
- ◆ HttpOnly flag is not set.
- ◆ Secure flag is also not visible (check on HTTPS).

OWASP Category:

A07:2021 – Identification and Authentication Failures

Fix Recommendations:

- Set the HttpOnly flag to prevent JavaScript access.
- Use the Secure flag to allow the cookie only over HTTPS.
- Regenerate tokens on login/logout and enforce short token lifespans.
- Implement XSS protection to prevent token leakage.



Recommendations

- Use input validation and output encoding across all forms.
- Remove outdated redirect links and apply strict allowlisting.
- Prevent access to unnecessary files and routes.
- Always apply the **Principle of Least Privilege**.
- Regularly scan and test the application with automated tools.

Conclusion

This assessment demonstrated how common web vulnerabilities can be identified using simple tools and techniques. Addressing these vulnerabilities ensures a more secure web application and builds trust with users.