Task 1 — Web Application Security Testing (OWASP Juice Shop)

Intern: Daniel Isaac E | Project: Future Interns — Cyber Security Internship | Date: 2025-11-10

Executive summary

During a web-app security test of a local OWASP Juice Shop instance I identified stored user content being accepted by the product reviews endpoint. A crafted review containing a JavaScript payload was accepted by the server (HTTP 201 / {"status":"success"}) and stored. The payload appears in the product reviews area, demonstrating persistence of untrusted input. While the current client displays the content as literal/escaped text in this instance, storing raw HTML/JS in the application creates a realistic risk that another rendering path or future change could execute the content (Stored XSS).

Affected asset

Application: OWASP Juice Shop (local). Affected endpoint: reviews API (HTTP PUT to product reviews). Affected UI: product details / reviews display.

Proof of concept (PoC) — short & repeatable

1) Authenticate to the application as a normal user.
2) Submit a review to a product using the reviews API with a payload that contains script content (e.g. alert('XSS-POC')).
3) Server responds with success (201 Created).
4) Reload the product page and observe the submitted review present in the reviews list — the content is persisted by the application.

Impact

Type: Stored Cross-Site Scripting (CWE-79). Risk summary: Stored XSS can allow attackers to run arbitrary JavaScript in victims' browsers if stored content is later rendered without proper encoding. Estimated severity: Medium.

Recommended remediation (practical & prioritized)

1) Sanitize input server-side.
2) Use context-sensitive output encoding.
3) Enforce a strong Content Security Policy (CSP).
4) Harden session handling.
5) Add automated tests.
6) Audit rendering paths.

Evidence & deliverables included

All supporting evidence gathered during testing (screenshots showing the UI and captured HTTP requests/responses) is included in the Evidence folder provided with this submission.

Closing notes

This is a lab/local finding and a useful learning PoC. After remediation, re-test the flow to verify that injected content is not executed.