Security Assessment Report

Vulnerability Type: Stored Cross-Site Scripting (XSS)
Target: PortSwigger Academy Lab – *Stored XSS into HTML context*
Assessor: Abdelalim Saada
Tools Used: Firefox Developer Tools, , PortSwigger Academy

1. Vulnerability Description

Stored XSS occurs when user input is stored on the server (e.g., in a database) and later included in pages sent to other users **without proper sanitization**. Unlike Reflected XSS, the malicious payload is permanently stored and triggered every time the vulnerable page is viewed.

2. Steps to Reproduce

1. **Enter the Lab**:
    - PortSwigger Academy lab: *Stored XSS into HTML context*
2. **Locate vulnerable input**:
    - Open one of the blog **articles**
    - Found a comment input box at the bottom of the article
3. **Test harmless HTML**:
    - Input:
    - `<h1>Test Header</h1>`
    - Rendered in larger font → HTML **not sanitized**
4. **Inject malicious JavaScript payload**:
    - Input:
    - `<script>alert('Stored XSS')</script>`
    - Alert popped up every time the article was revisited
5. **Captured Screenshots**:
    - Comment submission
    - Page reloaded with XSS triggered
    - Popup shown from stored script

3. Root Cause

- The application **stores user-submitted content** (comments) without sanitizing or encoding it.
- Later, the content is rendered inside the HTML context of a blog post **without escaping special characters**, enabling script execution.

4. Risk Assessment

Category	Details
Impact	Critical – Stored XSS affects all users viewing the content.
Likelihood	High – Easy to exploit, no input filtering applied

Category	Details
OWASP	A03:2021 – Injection (XSS)

5. Mitigation Recommendations

1. **Sanitize User Input**
   - Remove potentially dangerous input, such as `<script>` tags.
2. **Context-Aware Output Encoding**
   - Always encode user-generated content before rendering in HTML.
   - Use frameworks/libraries that escape output automatically.
3. **Content Security Policy (CSP)**
   - Enforce a restrictive CSP to block inline scripts:
   - `Content-Security-Policy: script-src 'self';`
4. **HTML Escaping Libraries**
   - Use libraries like DOMPurify or built-in templating sanitizers.

6. OWASP Mapping

OWASP Top 10	Vulnerability Type	Found
A03:2021	Injection (Stored XSS)	☑ Yes

.