**Information Security COMP 421**

**Spring 2025 Section B**

**Assignment 2**

**Hafsah Shahbaz – 251684784**

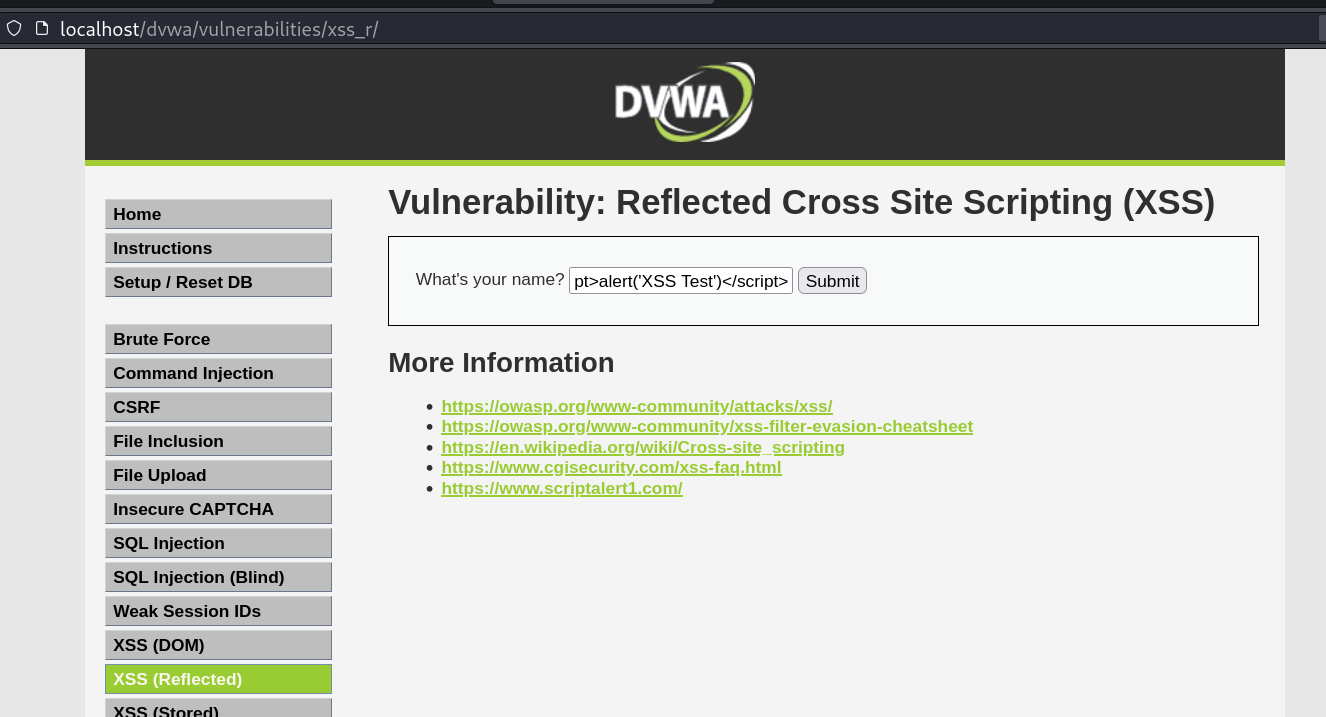
**Task 1 (Reflected XSS Attack):**

Observation:

The input <script>alert('XSS Test')</script> triggered a browser pop-up after submission. This confirms that the server reflected the input back into the page without sanitization. The JavaScript executed instantly, showing that the site is vulnerable to reflected XSS.

Lessons Learned:

* Reflected XSS works by injecting malicious scripts that bounce off the server and execute in the browser.
* It can be used in phishing links or manipulated URLs to trick users into executing unwanted scripts.
* Even a simple form input, when mishandled, can become an entry point for attackers.

****

**A black rectangle with a white border

AI-generated content may be incorrect.**

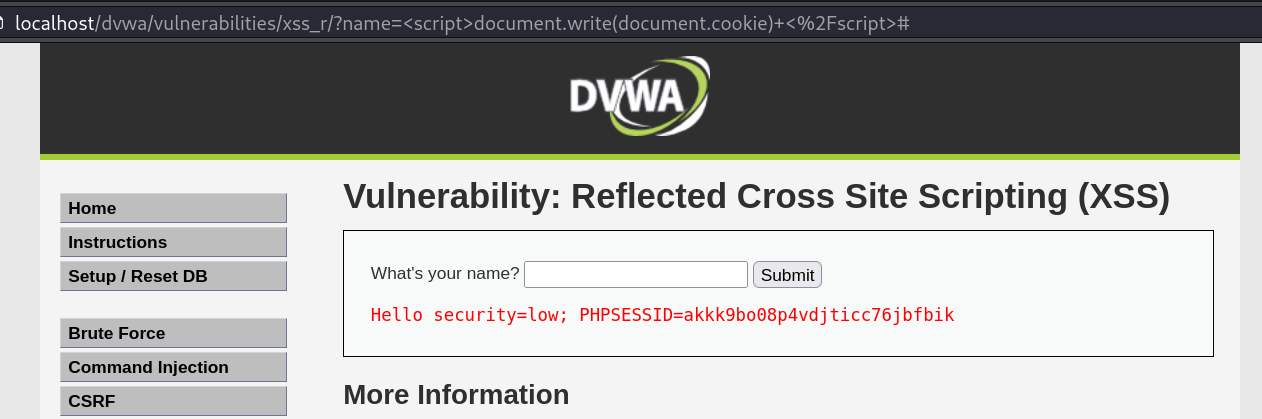
**Tasks 2 (Steal Cookie via XSS):**

Observation:

Using <script>document.write(document.cookie)</script>, the browser printed the current session cookie to the page. This proves an attacker could steal authentication tokens if this script were embedded in a malicious payload.

Lessons Learned:

* Cookies are vulnerable if not protected properly.
* Attackers can use XSS to hijack sessions by stealing session cookies.
* To protect cookies, developers must use HttpOnly and Secure flags so they can’t be accessed via JavaScript.

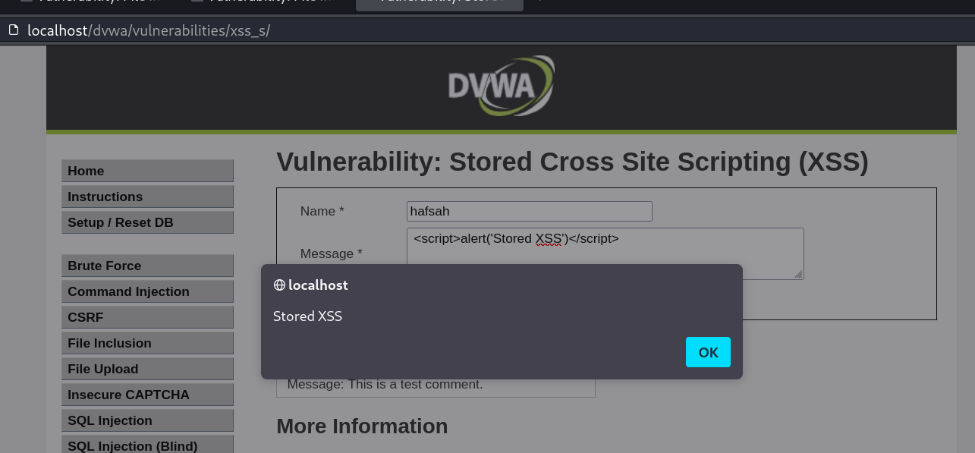
****

**Tasks 3 (Stored XSS Attack):**

Observation:

After submitting a comment with <script>alert('Stored XSS')</script>, the alert popped up every time the page loaded — even after a refresh. This confirms the script was stored in the backend and executed on page render, affecting all users.

Lessons Learned:

* Stored XSS is persistent and impacts every user who views the affected page.
* It’s one of the most dangerous types of XSS since it lives in the system until manually removed.
* ****Proper input validation and output encoding are essential to prevent this — especially in forums, comment sections, or any place user content is saved and displayed.