Vulnerability Report

Vulnerability Information

- **Vulnerability Name:** Reflected and Stored Cross-Site Scripting (XSS) in DVWA
- **Location:**
  - Reflected XSS: `http://localhost/dvwa/vulnerabilities/xss_r/`
  - Stored XSS: `http://localhost/dvwa/vulnerabilities/xss_s/`
- **Severity:** High

Vulnerability Description

Reflected XSS

- **Summary:**
  The reflected XSS vulnerability exists in the input field on the Reflected XSS page of DVWA, where user input is reflected into the HTML response without proper sanitization, allowing for JavaScript execution in the victim's browser.
- **Steps to Reproduce:**
  1. Go to: `http://localhost/dvwa/vulnerabilities/xss_r/`
  2. Enter the following payload in the “Name” input field:
     `<script>alert('Reflected XSS')</script>`
  3. Click the "Submit" button.
  4. The alert box will be executed, confirming the XSS vulnerability.

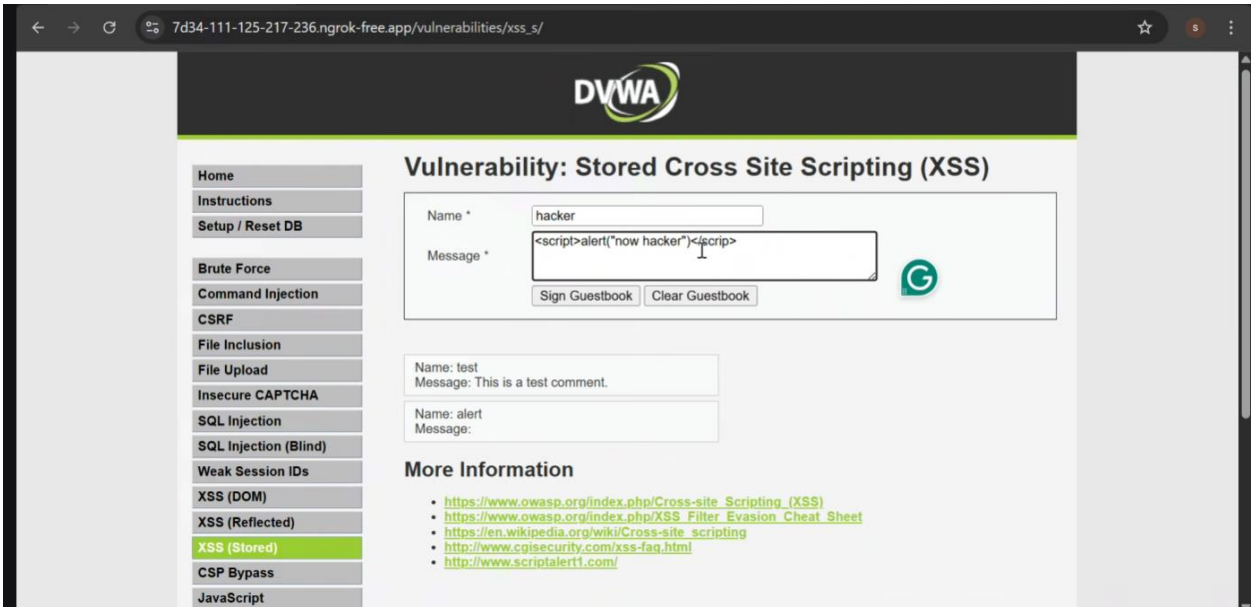
Stored XSS

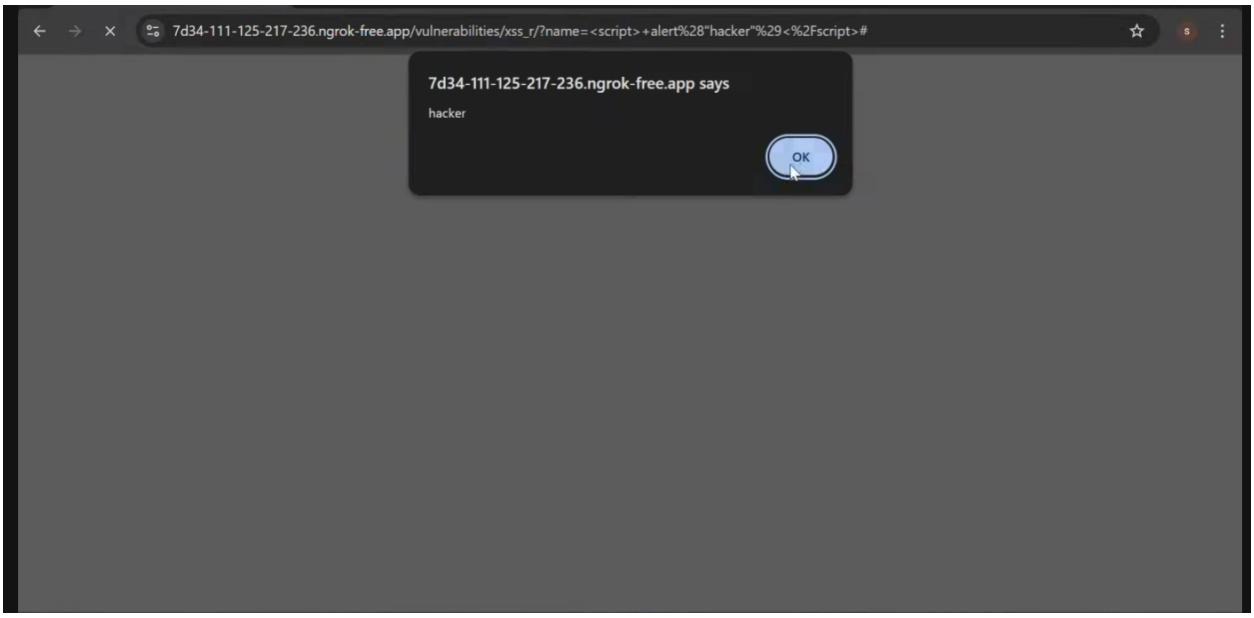
- **Summary:**
  The Stored XSS vulnerability exists in the message board (Guestbook) section of DVWA. When a user submits malicious JavaScript code, it is stored in the backend and executed when other users view the page.
- **Steps to Reproduce:**
  1. Go to: `http://localhost/dvwa/vulnerabilities/xss_s/`
  2. Enter any name in the “Name” field.
  3. Enter the following payload in the “Message” field:
     `<script>alert('Stored XSS')</script>`
  4. Click "Sign Guestbook".
  5. When the page reloads, the script is executed automatically.

Proof of Concept (PoC)

Reflected XSS

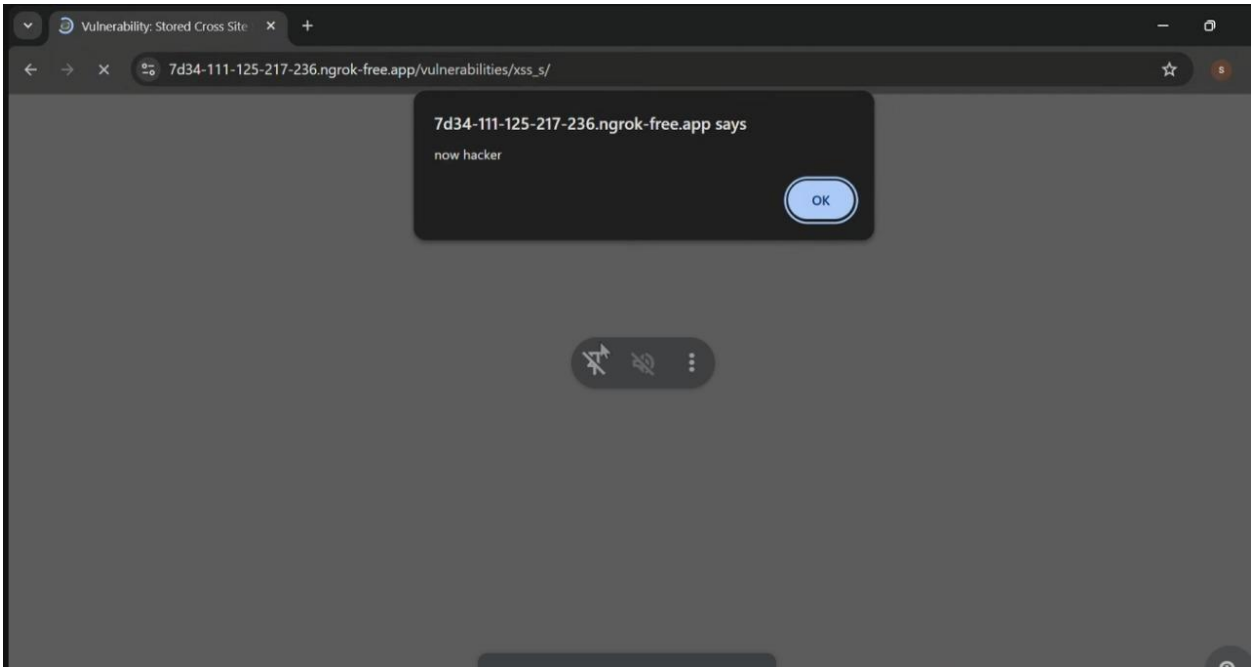
- **Description:**
  The payload `<script>alert('Reflected XSS')</script>` is injected into the name parameter and is directly reflected back in the HTML without encoding, leading to script execution.
- **PoC URL:**
  `http://localhost/dvwa/vulnerabilities/xss_r/?name=<script>alert('Reflected XSS')</script>`





Stored XSS

- **Description:**
  The `<script>alert('Stored XSS')</script>` payload is submitted via the Guestbook form and stored in the application. When any user views the Guestbook, the script executes.
- **Persistent Trigger:** Anyone viewing the Guestbook page sees the alert box.



C:\Users\PC\Videos\XSS Attack.mp4

Recommendations

Mitigation

- Use proper **output encoding** when displaying user-supplied data.
- Implement input validation to **reject or sanitize** HTML/JavaScript input.
- Use security libraries or frameworks that **automatically escape output**.

Best Practices

- Implement a strong **Content Security Policy (CSP)** to reduce the impact of XSS.
- Encode output based on context: HTML, JavaScript, URL, etc.
- Regularly scan the application using automated tools (like OWASP ZAP).
- Educate developers on **secure coding practices** for handling user input.

Team Member Information

- **Reported by:** Nagarjun Helambkar, Amey Parab, Pruthviraj Kanitkar
- **Reviewed by:** Tech Syndicates

This report highlights two major XSS vulnerabilities in DVWA, which demonstrate the risk of improper input handling. If exploited in a real-world scenario, such vulnerabilities could lead to session hijacking, defacement, or theft of sensitive data. Proper input validation, output encoding, and adherence to security best practices are essential to mitigate these risks.
