Task 2: Web Application Security Basics

Objective

To identify basic web application vulnerabilities such as Cross-Site Scripting (XSS) using a deliberately vulnerable web application in a controlled environment.

Test Environment

- **Application:** Damn Vulnerable Web Application (DVWA)
- **URL:** http://127.0.0.1/dvwa
- **Operating System:** Kali Linux
- **Security Level:** Low
- **Authorization:** Testing performed on a local demo application for educational purposes only

Vulnerability Tested: Reflected Cross-Site Scripting (XSS)

Description

Reflected XSS occurs when user-supplied input is immediately returned by the web application without proper input validation or output encoding, allowing malicious scripts to execute in the user's browser.

Tool Used

- Web Browser (Firefox)
- DVWA (Vulnerable test application)

Payload Used

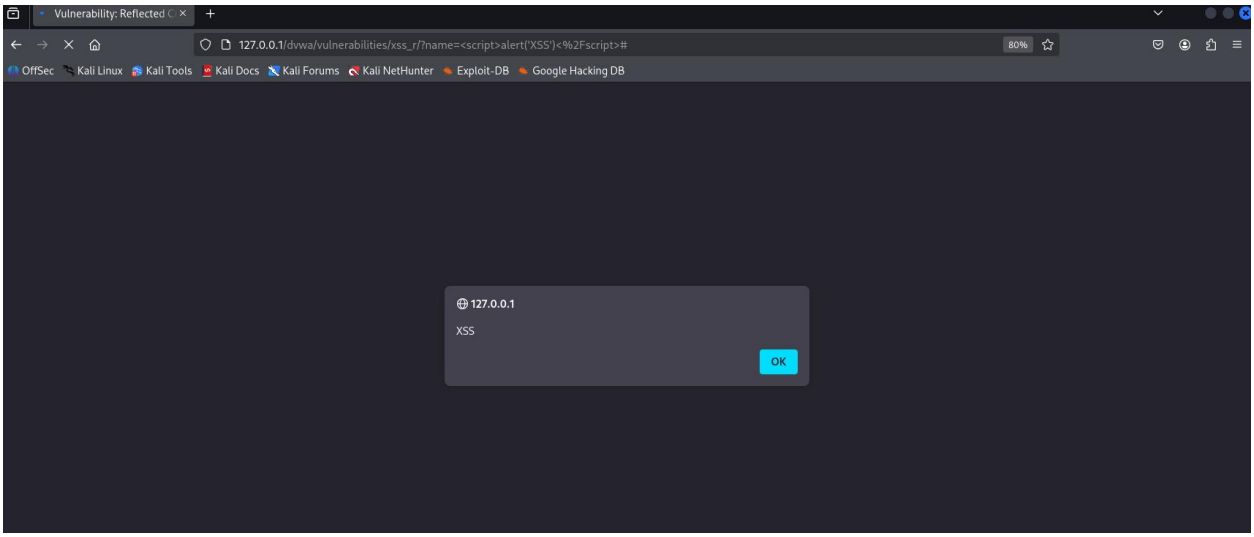
```
<script>alert('XSS')</script>
```

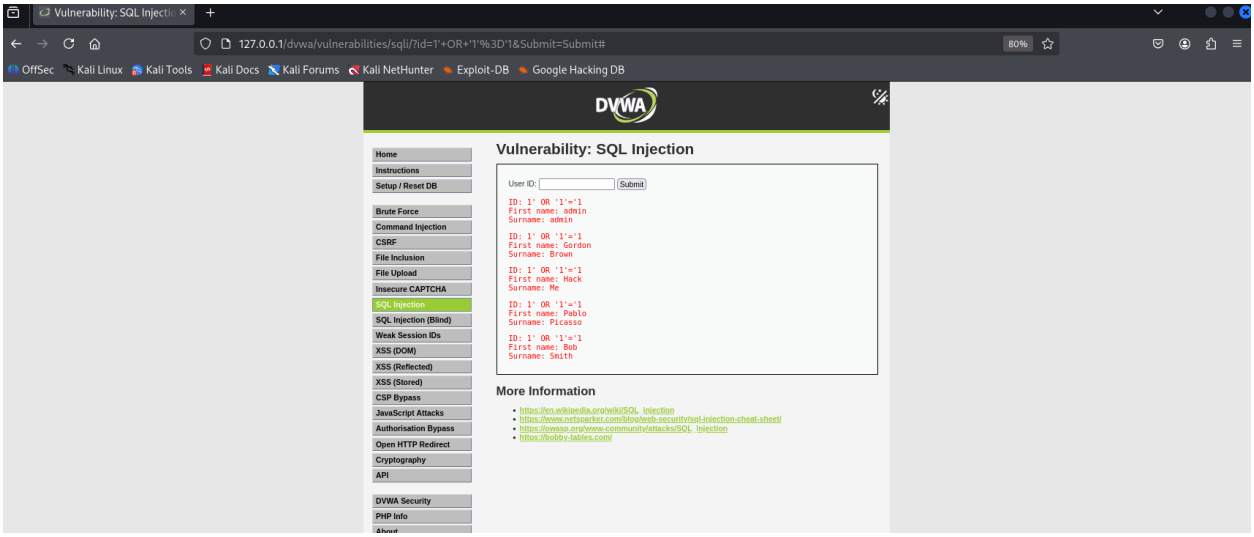
Steps Performed

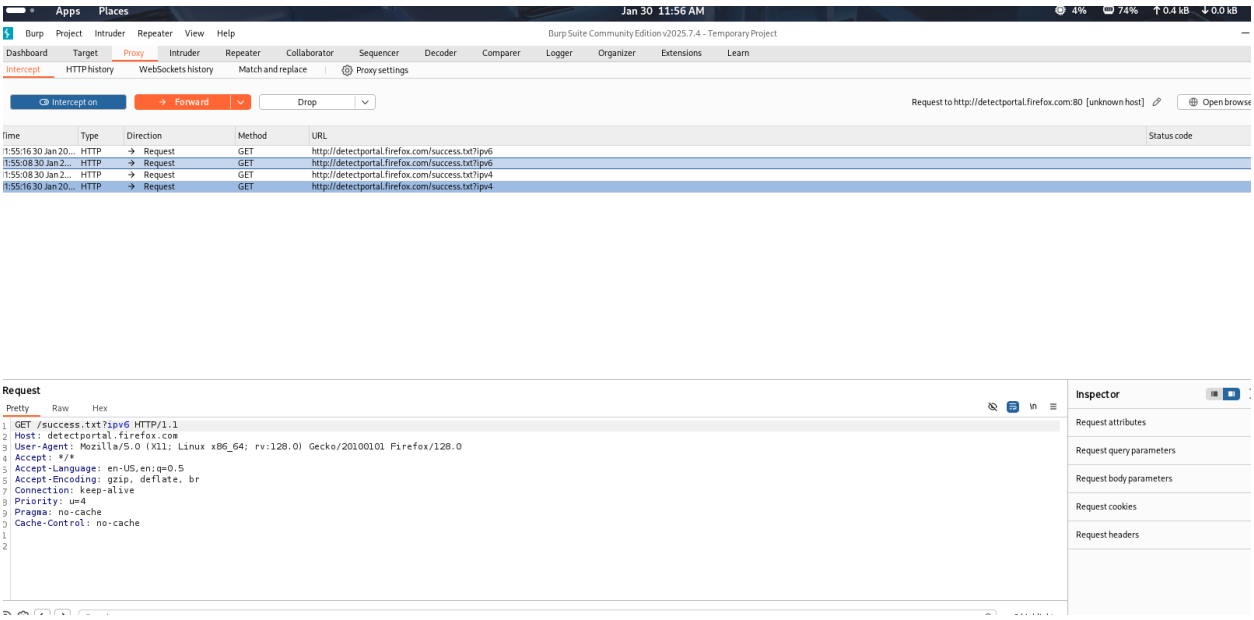
1. Logged into DVWA using valid credentials.
2. Set DVWA security level to **Low**.
3. Navigated to **XSS (Reflected)** module.
4. Entered the XSS payload into the input field.
5. Submitted the request.

Result / Observation

- A JavaScript alert box was displayed in the browser.
- This confirms that the injected script executed successfully.
- The application is vulnerable to reflected Cross-Site Scripting.







Captured on Burpsuite

Impact / Security Risk

- Attackers can execute malicious scripts in a victim's browser.
- Sensitive data such as session cookies may be stolen.
- Can lead to session hijacking and phishing attacks.

Mitigation Recommendations

- Validate and sanitize all user inputs.
- Encode output before rendering it in the browser.
- Implement Content Security Policy (CSP).
- Use secure frameworks that automatically escape user input.

Conclusion

The DVWA application was found to be vulnerable to reflected XSS due to improper input validation. This highlights the importance of secure coding practices and regular security testing in web applications.