XSS Vulnerability Testing using DVWA

# 1. INTRODUCTION

Cross-Site Scripting (XSS) is a web site security vulnerability that enables an attacker to inject hostile scripts into a web site. The scripts are run by the browsers of the users and can result in session hijacking, data stealing, and other malicious activities.

The objective of this project is to analyze and illustrate different XSS vulnerability types through the use of the Damn Vulnerable Web Application (DVWA) tool in a secure, isolated environment.

# 2. EXISTING SOLUTIONS

There are a number of tools and techniques to fight XSS vulnerabilities:

- Web Application Firewalls (WAFs)

- Content Security Policies (CSP)

- Strong libraries such as React and Angular which escape outputs automatically

- Security testing tools such as Burp Suite, OWASP ZAP, and Acunetix

Despite all this, most web applications remain vulnerable because of poor validation and insecure code.

# 3. PROPOSED SOLUTIONS

The purpose of this project is to:

- Show Reflected, Stored, and DOM-Based XSS attacks

- Use real-world payloads in a test environment under control

- Provide effective mitigating measures

- Create a DVWA Dockerized testing lab securely

# 4. SYSTEM REQUIREMENTS AND TOOLS

fHardware Requirements:  
- CPU: Intel i3 or above  
- RAM: Minimum 4 GB (8 GB recommended)  
- Storage: At least 10 GB free space  
- Internet connection for Docker image download

Software Requirements:  
- OS: Windows 10/11, macOS, or any Linux distribution  
- Docker: Latest stable version  
- DVWA: Pulled via Docker  
- Browser: Chrome, Firefox (latest version)  
- Burp Suite: Community Edition  
- Text Editor: VS Code or Notepad++

# 5. EXECUTION AND RESULTS

A. Reflected XSS:  
- Payload injected into a URL query parameter  
- Script executed immediately upon page load

B. Stored XSS:  
- Malicious script submitted through a form (e.g., comment section)  
- Script stored in the database and executed on page revisit

C. DOM-Based XSS:  
- JavaScript payload executed by manipulating the DOM via URL fragment

Tools Used:  
- Burp Suite for intercepting and analyzing requests  
- Browser Developer Tools for testing DOM manipulation

Observations:  
- All three XSS types were successfully demonstrated  
- Screenshots captured for evidence  
- Safe and controlled payloads were used

# 6. CONCLUSION AND REFERENCES

The project was able to successfully demonstrate the vulnerability of web applications to XSS attacks. DVWA, a secure and controlled testing environment, provided students with the experience of working directly with the attack vectors and XSS protection mechanisms.

References:  
- OWASP XSS Guide: https://owasp.org/www-community/attacks/xss/  
- DVWA GitHub Repository: https://github.com/digininja/DVWA  
- PortSwigger XSS Tutorials: https://portswigger.net/web-security/cross-site-scripting  
- Burp Suite Docs: <https://portswigger.net/burp>