Building a Web-Based Platform for Demonstrating Reflected and Stored XSS Attacks with Mitigations

Project Report submitted in partial fulfilment of the requirements of

Mini Project

During the course

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Name: Arjun
Roll no. 2315000414

Name: Sourav Yadav
Roll no. 2315002197

Name: Malkeet Singh
Roll no. 2315001301

Name: Mithi Kumari
Roll no. 2315001363

Under

Dr. Arun Singh Yadav

Department of Computer Engineering & Applications Institute of Engineering & Technology GLA University Mathura- 281406, INDIA 2025

Project Synopsis

Building a Web-Based Platform for Demonstrating Reflected and Stored XSS Attacks with Mitigations

1. Introduction

Cross-Site Scripting (XSS) is a critical security vulnerability in web applications that allows attackers to inject malicious scripts into a website.
These scripts can be used to steal sensitive information, hijack sessions, or perform unauthorized actions on behalf of a user.
This project aims to build a secure, web-based platform that demonstrates both Reflected XSS (non-persistent) and Stored XSS (persistent) attacks in a safe, controlled environment.
The platform will also showcase effective mitigation techniques to prevent such attacks, providing a valuable learning resource for students, developers, and cybersecurity professionals.

2. Problem Statement

Although XSS vulnerabilities are well-documented, many developers lack hands-on exposure to identifying and mitigating them.
Testing such vulnerabilities on live systems is risky, which creates a gap in practical training.
There is a lack of dedicated, safe, and interactive platforms where learners can see how XSS attacks work and explore real-world prevention strategies without causing harm to actual systems.

3. Proposed Solution

The proposed solution is to develop an interactive web application that contains intentionally vulnerable pages for demonstrating Reflected and Stored XSS attacks.
The platform will guide users through the process of executing these attacks, showing the results in real time.

Following the attack demonstration, users will be able to apply security measures such as:
- Input validation and sanitization
- Output encoding
- Implementation of Content Security Policy (CSP)

The system will be fully self-contained, running in a local or sandboxed environment to ensure safety.
The architecture will consist of a frontend interface for user interaction, a backend server for processing requests, and a database for simulating stored attack payloads.

4. Key Functionality

- Reflected XSS Demonstration: Simulate URL-based script injection.
- Stored XSS Demonstration: Store malicious payloads in a database and execute them on page load.
- Attack Walkthrough: Step-by-step explanation of each attack scenario.

- Mitigation Module: Implement and visualize various security fixes.
- Safe Learning Environment: Isolated, offline execution to prevent misuse.

5. Innovative Functionality

- Dual View Mode: Switch between "Attacker View" and "Defender View" to understand both perspectives.
- Live Payload Execution Preview: Visualize how a payload executes in real time.
- Interactive Security Testing: Apply different security measures and instantly see their effectiveness.
- Educational Narratives: Built-in explanations for each step of the attack and defense process.

6. Timeline for Development (Two-Month Plan)

Week 1-2: Requirement analysis, environment setup, basic frontend & backend structure.
Week 3-4: Develop Reflected XSS attack simulation module.
Week 5: Develop Stored XSS attack simulation module with database integration.
Week 6: Implement mitigation modules (input validation, output encoding, CSP).
Week 7: Testing, debugging, and refining the UI/UX.
Week 8: Documentation and final project presentation preparation.

7. Tools and Technology Required

- Frontend: HTML, CSS, JavaScript
- Backend: PHP / Node.js / Python Flask
- Database: MySQL / MongoDB
- Development Tools: Visual Studio Code, XAMPP/WAMP or Node.js runtime
- Security Frameworks: OWASP guidelines, HTML sanitizers
- Testing Tools: Browser developer tools, Postman

8. Conclusion

This project will create a safe, practical, and interactive learning environment for understanding and mitigating XSS vulnerabilities.
By simulating both Reflected and Stored XSS attacks and providing hands-on experience with mitigation techniques, it will bridge the gap between theoretical knowledge and real-world security practices.
The platform's educational value will benefit developers, students, and cybersecurity professionals, ultimately contributing to the development of more secure web applications.