SQL Injection Playground with Vulnerability Scanner

Introduction

This project demonstrates a realistic web application with common web vulnerabilities including SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). The goal is to provide a platform for testing and simulating attacks, along with a custom-built vulnerability scanner that identifies these issues.

Abstract

The project includes two major components:

1. A multi-page vulnerable web application that mimics real-world login, feedback, and search pages.
2. A custom Python-based vulnerability scanner that automatically detects SQLi, XSS, and CSRF by submitting payloads and analyzing responses.

It helps learners and testers gain hands-on experience with both exploiting and detecting these vulnerabilities, making it ideal for penetration testing practice and cybersecurity education.

SQL Injection Playground with Vulnerability Scanner

Tools Used

- Python (Flask) for backend server and scanner logic
- HTML/CSS/Bootstrap for web app UI
- JavaScript for client-side rendering and alerts
- BeautifulSoup & Requests for scanning and parsing forms
- FPDF for exporting scan results
- Kali Linux and browser dev tools for testing

Steps Involved in Building the Project

1. Designed a vulnerable web app with login, feedback, and search pages.
2. Introduced intentional vulnerabilities (SQLi, XSS, CSRF).
3. Developed a Python-based scanner that:
   - Crawls the target URL
   - Detects forms
   - Injects payloads
   - Analyzes responses
4. Created PDF reporting feature to export results.
5. Ensured only key vulnerabilities are reported.

Conclusion

This project combines attack simulation and detection in one environment. It enhances understanding of OWASP vulnerabilities and shows how automation can assist in penetration testing workflows. The export feature ensures scan results can be saved and shared.