Web Application Vulnerability Scanner – Project Report

Introduction

Web applications are widely used for online banking, shopping, education, and communication. However, they are also primary targets for attackers who exploit vulnerabilities such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). These vulnerabilities are part of the OWASP Top 10, which is an industry standard for web security risks.
The objective of this project is to design and implement a basic web application vulnerability scanner that can automatically crawl a target application, identify input points such as forms, and test them for common vulnerabilities. The project aims to help learners understand the vulnerability detection process and strengthen their practical knowledge of cybersecurity.

Abstract

The Web Application Vulnerability Scanner is a Python-based tool with a Flask web interface. It crawls through a target website, detects forms and inputs, and performs automated security tests. Specifically, the scanner checks for:

- Cross-Site Scripting (XSS): Injecting malicious scripts into input fields.
- SQL Injection (SQLi): Sending crafted SQL payloads to detect database errors.
- Cross-Site Request Forgery (CSRF): Identifying missing CSRF protection tokens in forms.

The tool logs results into an SQLite database and displays them in a user-friendly web interface, making it easy to review and analyze detected vulnerabilities.

Tools Used

- Python 3 – Core programming language.
- Flask – Web framework for building the user interface.
- Requests – To send HTTP requests to the target.
- BeautifulSoup (bs4) – For parsing HTML and extracting links/forms.
- SQLite – Lightweight database for storing scan results.

Steps Involved in Building the Project

- Crawler Development: Implemented a crawler to extract links and forms from target pages.
- Vulnerability Testing: Tested forms for XSS, SQLi, and CSRF vulnerabilities.
- Database Logging: Stored vulnerability details in SQLite with severity levels.
- Flask Web Interface: Built UI to input targets, display results, and view scan history.

Conclusion

This project provided practical knowledge of how vulnerability scanners work and helped reinforce concepts of web application security. The tool successfully detected XSS, SQLi,

and CSRF vulnerabilities on deliberately insecure test sites such as Acunetix's testphp.vulnweb.com.
While the scanner is basic, it demonstrates the complete cycle of crawling, payload injection, vulnerability detection, and reporting.

Future Enhancements

- Add detection for other vulnerabilities such as File Upload flaws, Open Redirects, and Command Injection.
- Implement multi-threading for faster scans.
- Provide an option to export scan results as PDF/CSV reports.
- Add authentication support to scan login-protected areas of web apps.

Author: Mohammad Farhan Hussain
Date: September 2025