Project Report: Web Application Vulnerability Scanner

Introduction:
The Web Application Vulnerability Scanner project focuses on automating the detection of common web security issues in applications. It helps identify and mitigate vulnerabilities before they can be exploited by attackers, enhancing the overall security posture of a website or web system.

Abstract:
This project is designed to detect vulnerabilities listed under the OWASP Top 10 such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). The system uses Python libraries like Requests and BeautifulSoup for web crawling, while regex-based analysis identifies vulnerable inputs. A Flask-based interface enables users to manage scans, view detected issues, and generate detailed reports with severity ratings and evidence.

Tools Used:
- Python
- Requests, BeautifulSoup, Regex
- Flask Framework
- OWASP Top 10 Checklist

Steps Involved in Building the Project:
1. Studied common web vulnerabilities and created a list of attack payloads.
2. Implemented web crawling using Requests and BeautifulSoup to detect forms and URLs.
3. Injected malicious payloads for XSS, SQLi, and CSRF to test application security.
4. Analyzed HTTP responses to detect reflected or exploitable vulnerabilities.
5. Developed a Flask-based user interface to initiate scans and display reports.
6. Logged each vulnerability with its type, severity, and proof of concept.

Conclusion:
The Web Application Vulnerability Scanner effectively automates vulnerability testing, saving time and reducing manual effort. It helps developers and testers proactively identify weaknesses in web applications and take corrective action before exploitation occurs.