WEB APPLICATION VULNERABILITY SCANNER

Introduction

In today's digital landscape, web applications have become essential to businesses, governments, and individuals, but this reliance has also increased their exposure to cyber threats. The proliferation of vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), insecure configurations, and unencrypted data transport places sensitive information at serious risk. To mitigate such threats, regular vulnerability assessment and automated scanning are recognized as foundational steps in modern cybersecurity. This project focuses on the development and evaluation of a custom-built web vulnerability scanner, designed to detect and report common security flaws aligned with the OWASP Top 10 guidelines.

Abstract

The objective of this project was to design and develop a Python-based web vulnerability scanner capable of identifying critical security weaknesses in web applications. Leveraging open-source libraries and custom logic, the scanner automates detection of SQL Injection, XSS, CSRF, missing security headers, and SSL/TLS issues. Features include real-time vulnerability detection, severity grading, and detailed multi-format reporting (JSON, CSV, HTML). The tool was tested against intentionally vulnerable demo sites to validate its effectiveness. This report details the motivations, tools used, development methodology, and key outcomes of the project, culminating in a functional tool useful for both educational and real-world security assessment.

Tools Used

- **Python 3.7+**: Chosen for its extensive ecosystem and simplicity for scripting and automation.
- **Requests**: For making HTTP requests to target web servers.
- **BeautifulSoup4**: For parsing and analyzing HTML forms and responses.
- **lxml**: Provides efficient web data parsing support.
- **Visual Studio Code**: Used as the code editor and development environment.
- **Command-line / PowerShell / Terminal**: For running and testing the scanner scripts.
- **Test Sites**: testphp.vulnweb.com, demo.testfire.net, and self-hosted DVWA for realistic validation.

Steps Involved in Building the Project

1. **Requirement Analysis**: Studied the OWASP Top 10 vulnerabilities and common threats affecting web applications.
2. **Design Architecture**: Defined modular structure for code—separate functions for scanning, reporting, and utilities.
3. **Development**:

- Created logic for automated detection of vulnerabilities such as SQL Injection and XSS using crafted payloads.
- Implemented HTTP requests and HTML parsing to simulate attacker behavior and test web forms and parameters.
- Added scanning for missing HTTP security headers and insecure SSL/TLS deployment.
- Designed code to categorize findings by severity (critical, high, medium, low).
- Automated report generation in JSON, CSV, and HTML formats, with UTF-8 compatibility for system compatibility.

4. **Testing and Validation**:
    - Ran the scanner against intentionally vulnerable public demo sites.
    - Debugged and resolved issues like Unicode encoding errors and dynamic CSV header handling.
    - Verified that reports generated successfully and were human-readable.
5. **Documentation**: Prepared README, methodology notes, usage instructions and professional output reports for submission.

Conclusion

This project demonstrates a hands-on approach to cybersecurity tool development by delivering a reliable web vulnerability scanner tailored to educational and practical needs. The tool enables systematic detection of major web risks, promotes security awareness, and provides clear, actionable reporting to guide remediation. During tests on demo websites, it was effective at identifying real vulnerabilities and generating structured reports. Future enhancements could include adding authenticated scanning, support for more vulnerability types, or integrating with CI/CD pipelines to automate continuous security testing. The successful completion of this scanner not only fulfills academic requirements but also lays a solid groundwork for further research and advanced tool development in the field of web application security.