Project Report: Web Application Vulnerability Scanner

Abstract

This project focuses on building a Python-based Web Application Vulnerability Scanner. The tool is designed to automatically test websites for common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and missing security headers. By combining crawling, vulnerability testing, and reporting, the scanner provides an easy-to-understand summary of potential risks. The purpose is to help organizations and developers identify weaknesses before attackers exploit them.

Introduction

In today's digital era, web applications are frequent targets for cyber-attacks. Two of the most common and dangerous vulnerabilities are SQL Injection and Cross-Site Scripting. These issues often arise when user input is not properly validated or sanitized. The project was undertaken to understand how automated security tools work, and to create a simplified scanner that mimics the process used by professional tools like Burp Suite or OWASP ZAP.

Tools Used

- Python : Core programming language
- Requests: For sending HTTP requests
- BeautifulSoup4: For parsing HTML and extracting links/forms
- Colorama: For colorful console outputs
- Validators: For URL validation

Steps Involved in Building the Project

1. Environment Setup: Installed Python and required libraries.
2. Crawling: Developed a crawler to navigate the target website and extract links and forms.
3. SQL Injection Testing: Implemented payloads to check if forms or URL parameters are vulnerable.
4. XSS Testing: Injected JavaScript snippets to detect if inputs are reflected back unsafely.
5. Security Header Check: Verified if recommended HTTP headers like Content-Security-Policy are present.
6. Reporting: Summarized results in both console output and an HTML report for clarity.

Conclusion

The Web Application Vulnerability Scanner project gave practical insights into how automated security tools identify risks in web applications. It reinforced the importance of secure coding practices and demonstrated how Python can be leveraged for cybersecurity automation. While the scanner is a simplified version of enterprise tools, it successfully highlights common vulnerabilities and provides a strong foundation for understanding web application security testing.