PROJECT:WEB APP VULNERABILITY SCANNER

Introduction

With the increasing reliance on web applications, **security vulnerabilities** have become a critical concern. Developers often focus on functionality and overlook security, leaving applications susceptible to exploitation. Traditional manual security testing is time-consuming and prone to human error, making automated vulnerability scanning essential.

The **Web Application Vulnerability Scanner** aims to simplify the detection of common vulnerabilities. It scans websites by:

- Crawling links and forms.
- Injecting malicious payloads.
- Analyzing responses to detect anomalies.

This scanner was designed for developers and security enthusiasts to perform quick security checks on web applications. While it is not a substitute for comprehensive penetration testing, it provides valuable insights during the early stages of development.

Tools Used

The project uses Python as the core programming language due to its extensive libraries and flexibility.

Libraries and Frameworks:

- requests – for sending HTTP requests.
- BeautifulSoup – for parsing HTML and detecting forms.
- re – for regular expression matching.
- urllib – for handling URLs.

Testing Platforms:

- OWASP Juice Shop

- HackThisSite
- bWAPP (Buggy Web App)

Steps Involved in Building the Project

1. **Requirement Analysis:** Identify target vulnerabilities (XSS, SQLi, CSRF) and understand OWASP Top 10 guidelines.
2. **Environment Setup:** Install Python and required libraries (requests, BeautifulSoup).
3. **Crawling & Form Detection:**
   Extract URLs and forms from the target web page.
   Identify input parameters for testing.
4. **Payload Injection:**
   Inject test payloads such as "<script>alert('XSS')</script>" for XSS and "' OR '1'='1" for SQLi.
5. **Response Analysis:**
   Analyze responses for signs of execution or error messages.
6. **Result Generation:**
   Display detected vulnerabilities with endpoint details.
7. **Testing:**
   Run the scanner on intentionally vulnerable websites and record results.

Conclusion

The **Web Application Vulnerability Scanner** is an effective tool for detecting basic vulnerabilities in web applications. It automates the tedious process of scanning for XSS, SQLi, and CSRF, helping developers secure their applications during development.

During testing, the tool successfully detected vulnerabilities in OWASP Juice Shop and bWAPP, while secure websites returned no issues, demonstrating its accuracy.

Limitations: It does not cover advanced vulnerabilities like SSRF or RCE and cannot handle JavaScript-heavy single-page applications without tools like Selenium.

Future Work: Enhancements include building a GUI dashboard, expanding payloads, and integrating advanced vulnerability detection.