Elevate Labs (Cyber-Security) – Internship

Project-Title: *Quantum-Tech : Web-Application Vulnerability (Scanner)*

➢ Introduction:

In today's digital era, web applications play a crucial role in delivering services, processing transactions, and managing sensitive information. However, their wide exposure to the internet makes them frequent targets for cyber-attacks. Vulnerabilities such as **Cross-Site Scripting (XSS), SQL Injection (SQLi), Local File Inclusion (LFI), Server-Side Request Forgery (SSRF), and Server-Side Template Injection (SSTI)** are among the most exploited by attackers to compromise security.

This project, **Web Application Vulnerability Scanner**, has been developed to automate the process of detecting these vulnerabilities. The scanner helps penetration testers and security professionals by reducing manual effort, increasing efficiency, and ensuring comprehensive coverage during web application assessments.

➢ Abstract:

The **Web Application Vulnerability Scanner** is a Python-driven project that integrates custom scripts with widely used open-source security tools. It automates the detection of vulnerabilities by:

- Extracting URL parameters from target web applications.
- Injecting crafted payloads into those parameters.
- Analysing HTTP responses to confirm the presence of vulnerabilities.

The scanner supports multiple vulnerability classes:

- **XSS** – Detects reflected input by injecting JavaScript-based payloads.
- **SQLi** – Uses **sqlmap** to identify injection flaws and enumerate databases.
- **LFI** – Tests for path traversal by attempting to access system files.
- **SSRF** – Confirms server-side requests to external services.
- **SSTI** – Detects template engines executing injected expressions.

By combining automation and payload-based verification, the scanner provides reliable results that can be leveraged for securing applications before deployment.

➢ Tools & Technologies Used:

The scanner leverages both **custom Python modules** and **external open-source tools** to maximize efficiency and accuracy:

- **Programming Language:** Python 3
- **Security Tools:**
    - **GF (Gf-Patterns):** Extracts parameters based on vulnerability signatures.

    - **httpx:** Lightweight HTTP client for response inspection.
    - **sqlmap:** Industry-standard SQLi detection and exploitation tool.
    - **kxss & Gxss:** Identifies reflected parameters vulnerable to XSS.
    - **qsreplace:** Replaces query parameters with payloads for quick testing.
- **Payload Repositories:** Custom .txt files containing crafted payloads for each vulnerability type (XSS, LFI, SSTI, SSRF).
- **Optional Integrations:** Interactsh or Burp Collaborator for SSRF detection.

➢ Methodology & Step Involved:

1. **Target Enumeration & Input Collection**
    - User provides a domain/URL list.
    - GF tool extracts parameters (e.g., id=, file=, redirect=).
2. **Module Execution**
    - **XSS Scanner:** Uses Gxss + Kxss to find reflected parameters, injects payloads, and checks reflections in HTTP responses.
    - **SQLi Scanner:** Runs sqlmap in two stages – (i) quick batch testing on multiple URLs, (ii) detailed exploitation on confirmed vulnerable endpoints.
    - **LFI Scanner:** Replaces parameter values with directory traversal payloads (../../etc/passwd) and validates through known markers (root:x:). Includes null-byte and wrapper techniques.
    - **SSRF Scanner:** Injects external callback URLs to detect server-side requests.
    - **SSTI Scanner:** Injects arithmetic/logical template payloads ({{7*7}}, ${7*7}) and validates execution in the response.
3. **Automation & Logging**
    - Each module writes its results to a dedicated output file (xss_output.txt, sqli_output.txt, etc.).
    - Successful detections are tagged as [VULNERABLE] for easy review.

➢ Conclusion:

The **Web Application Vulnerability Scanner** demonstrates the importance and practicality of **automation in penetration testing**. Instead of manually testing every parameter, the scanner streamlines vulnerability detection by integrating open-source tools with intelligent payload injection techniques.

This project not only improves the efficiency of web security assessments but also highlights how open-source tools can be combined with custom scripting to create powerful security utilities.

Future enhancements could include:

- Support for authenticated scanning (handling login sessions).
- Integration with a reporting dashboard for visualization.
- Adding detection for advanced vulnerabilities (e.g., RCE, CSRF).

By proactively identifying security flaws, this scanner empowers developers and organizations to build safer web applications.