Vulnerability Scanner Report

Project Name: Python-Based Vulnerability Scanner

Prepared By: Jatin Saini, Aryaman Pratap Singh , Vishesh Yadav , Ritik Balhara

Purpose: To identify security weaknesses in networks and web applications

Report created using chat gpt

1️ Overview

The Python-Based Vulnerability Scanner is designed to detect open ports, SQL injection vulnerabilities, and Cross-Site Scripting (XSS) vulnerabilities in a target system or web application. It is a lightweight security tool aimed at assisting cybersecurity professionals in assessing potential risks and misconfigurations.

2️ Features & Functionality

✅ Open Port Scanning

- Identifies open ports on a target machine

- Uses Python’s socket module for network communication

- Generates logs for detected open ports

✅ SQL Injection Detection

- Tests web application parameters for SQL injection vulnerabilities

- Uses malicious payloads to identify security flaws

- Logs vulnerable endpoints for analysis

✅ XSS Detection

- Scans for potential Cross-Site Scripting (XSS) attacks

- Detects JavaScript execution vulnerabilities in web inputs

- Stores results in a dedicated log file

✅ Automated Reporting

- Generates security logs in scan\_results.log

- Saves detailed findings in text files for analysis

3️ Methodology

- Port Scanning: Utilizes TCP socket connections to detect open ports on the target system.

- SQL Injection Testing: Appends payloads to website parameters and analyzes responses.

- XSS Detection: Injects JavaScript-based payloads into search/query fields to evaluate security flaws.

4️ Results & Findings

Target System: 192.168.1.100

Target Web Application: http://example.com/product

📌 Open Ports Detected:

- Port 80 - HTTP

- Port 443 - HTTPS

📌 SQL Injection Vulnerabilities Identified:

- /product?id= OR 1=1 -- (Possible database exposure)

📌 XSS Vulnerabilities Identified:

- /search?q=<script>alert(‘XSS’)</script>

5️ Recommendations

🔹 Port Security: Close unused ports or restrict access with firewall rules.

🔹 SQL Injection Prevention: Use prepared statements and parameterized queries.

🔹 XSS Mitigation: Implement input validation and content security policies (CSP).

6️ Conclusion

The Python-Based Vulnerability Scanner successfully identified security weaknesses in the target system and web application. The findings emphasize the need for stronger security measures, including firewall configurations, secure coding practices, and input sanitization to prevent cyber threats.