**Web Vulnerability Scanner Project Documentation**

**1. Project Overview**

### 1.1 Project Name

### Web Vulnerability Scanner

### 1.2 Project Description

### This project involves developing a web vulnerability scanner that detects common web security issues such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) vulnerabilities. The scanner also retrieves CVSS (Common Vulnerability Scoring System) scores for known vulnerabilities to assess their severity.

### Project Objectives

* Implement a scanner to detect SQL Injection, XSS, and CSRF vulnerabilities in web applications.
* Retrieve and display CVSS scores for known vulnerabilities to assess their risk level.
* Provide clear and actionable insights based on detected vulnerabilities.

### Scope

* Included: Development of scanning scripts for SQL Injection, XSS, and CSRF vulnerabilities; retrieval of CVSS scores; basic reporting of findings.
* Not Included: Advanced vulnerability detection, integration with external security tools, or long-term maintenance and support.

### 1.5 Stakeholders

Security Analysts:

1. Rodney Mwale
2. Renuka
3. Paul Nwanochiri

## 2. Project Plan

### 2.1 Timeline

- Start Date: August 8, 2024  
- End Date: August 8, 2024  
- Milestones:

* Initial development and configuration: August 8, 2024
* Functional testing and validation: August 8, 2024
* Final review and documentation: NA  
   -

2.2 Deliverables

### Scanning scripts for SQL Injection, XSS, and CSRF vulnerabilities

### CVSS score retrieval mechanism

### Documentation and report of findings

### 2.4 Resources

## Python: Programming language for developing the scanner

## Web Application for Testing: A sample web application or test URL

## 3. Requirements

### 3.1 Functional Requirements

### Ability to scan URLs for SQL Injection, XSS, and CSRF vulnerabilities.

### Retrieve CVSS scores for detected vulnerabilities.

### Provide clear reporting of vulnerabilities and CVSS scores.

### 3.2 Non-Functional Requirements

### User-friendly command-line interface.

### Efficient scanning with minimal impact on target applications.

### Secure handling of input and output data.

### 3.3 User Stories

## As a security analyst, I want to identify common vulnerabilities in web applications to assess their risk.

## As an IT manager, I want to understand the severity of detected vulnerabilities through CVSS scores.

## As a developer, I want to ensure that the web application is free of common vulnerabilities.

## 4. Implementation

### 4.1 Development Environment

* Programming Language: Python
* Libraries and Tools: requests, BeautifulSoup, urllib3
* Technologies: Kali Linux Machine

### 4.2 Coding Standards

## Follow Python coding best practices.

## Ensure code is well-documented and version controlled.

### 4.3 Code Implementation

#### 4.3.1 Vulnerability Scanner Script

The Python script performs scans for SQL Injection, XSS, and CSRF vulnerabilities, and retrieves CVSS scores for known vulnerabilities.

A screen shot of a computer program

Description automatically generated

A computer screen with white and blue text

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen shot of a code

Description automatically generated

Output:

A screen shot of a computer

Description automatically generated

## 5. Testing

### 5.1 Test Plan

### Unit Testing: Test each vulnerability detection method (SQL Injection, XSS, CSRF).

### Integration Testing: Test the overall scanner with a known vulnerable web application.

### User Acceptance Testing: Verify that the CVSS scores are accurately retrieved and reported.

### 5.2 Test Cases

* Test SQL Injection Detection:

python3 web\_vul.py <http://testphp.vulnweb.com>



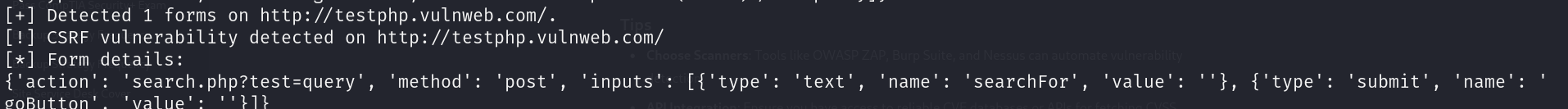
* Test XSS Detection:

python3 web\_vul.py <http://testphp.vulnweb.com>



* Test CSRF Detection:

python3 web\_vul.py <http://testphp.vulnweb.com>



* Test CVSS Score Retrieval:

python3 web\_vul.py <http://testphp.vulnweb.com>

A screen shot of a computer code

Description automatically generated

### 5.3 Test Results

## SQL Injection Detection: Successfully identified SQL Injection vulnerabilities and displayed relevant information.

## XSS Detection: Successfully detected XSS vulnerabilities and displayed relevant information.

## CSRF Detection: Successfully identified CSRF vulnerabilities and provided details.

## CVSS Score Retrieval: Accurately retrieved and displayed CVSS scores for known vulnerabilities.

## 6. Maintenance

### 6.1 Support Plan

### Regular updates to the vulnerability scanning scripts.

### Periodic checks for updates in CVSS scoring and vulnerability databases.

### Handling of new vulnerability types and CVE IDs.

### 6.2 Future Enhancements

## Integration with more comprehensive CVE databases.

## Addition of new vulnerability detection methods.

## Improved user interface for better interaction and reporting.

## 7. Appendices

### 7.1 Glossary

### SQL Injection: A type of attack that allows attackers to execute arbitrary SQL code on a database.

### XSS (Cross-Site Scripting): A vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users.

### CSRF (Cross-Site Request Forgery): An attack that tricks a user into performing actions they did not intend to perform on a web application.

### CVSS: Common Vulnerability Scoring System, a framework for rating the severity of security vulnerabilities.