Software Requirements Specification (SRS) for ZAP (OWASP Zed Attack Proxy)

1. Introduction

1.1 **Purpose** This document defines the software requirements for using OWASP ZAP (Zed Attack Proxy) as a web application security testing tool.

1.2 **Scope** The SRS applies to ZAP's functionalities for scanning and testing web applications for vulnerabilities, enabling security analysts to identify and remediate security risks.

1.3 **Definitions, Acronyms, and Abbreviations**

- **ZAP**: OWASP Zed Attack Proxy, a web application security scanner.
- **HTTP**: HyperText Transfer Protocol.
- **HTTPS**: HyperText Transfer Protocol Secure.
- **API**: Application Programming Interface.

2. Overall Description

2.1 **Product Perspective** ZAP is an open-source security tool that functions as a proxy between the user and the web application, enabling detailed analysis of HTTP(S) requests and responses.

2.2 **Product Functions**

- Automated and manual scanning for security vulnerabilities.
- Interception and modification of web traffic.
- Generation of detailed reports on vulnerabilities detected.
- Integration with CI/CD pipelines for continuous security testing.

2.3 **User Classes and Characteristics**

- **Security Analysts**: Professionals with expertise in web application security who use ZAP for penetration testing.
- **Developers**: Software developers who need to ensure their applications are secure from vulnerabilities.
- **Quality Assurance Engineers**: QA personnel integrating security testing into the software testing process.

2.4 **Operating Environment** ZAP operates in various environments, including:

- Windows
- Linux
- macOS

3. Specific Requirements

3.1 **Functional Requirements**

- **FR1**: The system shall allow users to configure proxy settings for capturing web traffic.

- **FR2**: The system shall support automated scans of web applications for known vulnerabilities.
- **FR3**: The system shall allow users to perform manual tests using a GUI or command-line interface.
- **FR4**: The system shall enable users to create and execute custom scripts for advanced testing.
- **FR5**: The system shall provide detailed reporting of vulnerabilities, including remediation guidance.
- **FR6**: The system shall support integration with CI/CD tools (e.g., Jenkins, GitLab CI).

3.2 **Non-Functional Requirements**

- **NFR1**: The system shall execute vulnerability scans efficiently, with minimal impact on application performance.
- **NFR2**: The system shall provide an intuitive user interface for both novice and experienced users.
- **NFR3**: The system shall maintain an accuracy rate of at least 95% for vulnerability detection.
- **NFR4**: The system shall be extensible to allow for third-party plugins and scripts.

4. System Features

4.1 **User Interface**

- GUI that supports drag-and-drop functionality for managing scans and configurations.
- Command-line interface for automated and scripted operations.

4.2 **Reporting**

- Ability to export reports in multiple formats (e.g., HTML, XML, Markdown).
- Summary dashboards displaying vulnerability trends and risk levels.

5. External Interface Requirements

5.1 **Hardware Interfaces**

- ZAP can be run on standard server or desktop hardware with network interface capabilities.

5.2 **Software Interfaces**

- Integration with web browsers for manual testing.
- APIs for integrating with other security tools and frameworks.

6. Performance Requirements

- The system should handle large-scale web applications and perform scans in a reasonable timeframe, typically under 30 minutes for standard tests.

7. Security Requirements

- The system should ensure that all scanning activities are conducted ethically and within the legal constraints of the target application.
- User authentication and access controls should be implemented to protect sensitive testing configurations.

8. Documentation

- Comprehensive user manuals covering installation, configuration, and usage scenarios.
- Technical documentation for developers contributing to ZAP's codebase or integrating with it.