**Software Requirements Specification**

For

Minor Project

Vulnerability Scanner

October 2024

Prepared by

|  |  |  |
| --- | --- | --- |
| **Name** | **SAP ID** | **Specialization** |
| Akash Aswal | 500126734 | BTECH CSE CSF B1(H) |
| Shashank Bharti | 500106026 | BTECH CSE CSF B5 |

School Of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Table of Contents

|  |  |  |
| --- | --- | --- |
| **Topic** | | **Page No** |
| Table of Content | |  |
| Revision History | |  |
| 1 | Introduction |  |
|  | 1.1 Purpose of the Project |  |
|  | 1.2 Target Beneficiary |  |
|  | 1.3 Project Scope |  |
|  | 1.4 References |  |
| 2 | Project Description |  |
|  | 2.1 Reference Algorithm |  |
|  | 2.2 Data/ Data structure |  |
|  | 2.3 SWOT Analysis |  |
|  | 2.4 Project Features |  |
|  | 2.5 User Classes and Characteristics |  |
|  | 2.6 Design and Implementation Constraints |  |
|  | 2.7 Design diagrams |  |
|  | 2.8 Assumption and Dependencies |  |
| 3 | System Requirements |  |
|  | 3.1 User Interface |  |
|  | 3.2 Software Interface |  |
|  | 3.3 Database Interface |  |
|  | 3.4 Protocols |  |
| 4 | Non-functional Requirements |  |
|  | 4.1 Performance requirements |  |
|  | 4.2 Security requirements |  |
|  | 4.3 Software Quality Attributes |  |
| 5 | Other Requirements |  |
| Appendix A: Glossary | |  |
| Appendix B: Analysis Model | |  |
| Appendix C: Issues List | |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Change** | **Reason for Changes** | **Mentor Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 1 | INTRODUCTION | |
|  | 1.1 Purpose of the Project | The purpose of this project is to develop a Vulnerability Scanner Tool that identifies, assesses, and prioritizes vulnerabilities such as SQL injection, XSS, and open ports. |
|  | 1.2 Target Beneficiary | students who are interested in cybersecurity |
|  | 1.3 Project Scope | The Vulnerability Scanner Tool is designed for use in cybersecurity environments, targeting website. It is applicable in organizations of all sizes, including educational institutions and small businesses |
|  | 1.4 References | OWASP Top 10 |
| 2 | PROJECT DESCRIPTION | |
|  | 2.1 Reference Algorithm | State the reference algorithm for the project and identify the required data structure (**Mandatory for Minor1**) Or/Add design algorithm justifying the methodology of the project |
|  | 2.2 Characteristic of Data | Present with the characteristic of the dataset used for the project. Provide the primary and secondary source of the data, along with sampling techniques. Explain the statistical method used for data processing (**if any**). |
|  | 2.3 SWOT Analysis | Present with a justification to support your project. |
|  | 2.4 Project Features | Summarize the major features the product contains or the significant functions that it performs or lets the user perform. (Level 2 USE Case diagram) |
|  | 2.5 User Classes and Characteristics | Identify the various user classes that you anticipate will use this product. |
|  | 2.6 Design and Implementation Constraints | Present hardware boundary conditions (timing requirements, memory requirements); interfaces to other applications; specific technologies, and tools to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards. |
|  | 2.7 Design diagrams | Present all the required Diagram (USE –Case, Class Diagram, Activity, Sequence, Data Flow diagram and State Diagram. (Major project should include Collaboration and Deployment Diagram too) |
|  | 2.8 Assumption and Dependencies | List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. Also identify any dependencies the project has on external factors. |
| 3 | SYSTEM REQUIREMENTS | |
|  | 3.1 User Interface | Define the software components for which a user interface is needed. |
|  | 3.2 Software Interface | Describe the connections between modules. Describe the services needed and the nature of communications. Describe detailed application programming interface protocols. |
|  | 3.3 Database Interface | Explain the Database management system used |
|  | 3.4 Protocols | Describe the requirements associated with any protocol deployed in the project. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms |
| 4 | NON-FUNCTIONAL REQUIREMENTS | |
|  | 4.1 Performance requirements | If there are performance requirements for the product under various circumstances, state them. Specify the timing relationships for real time systems. State performance requirements for individual functional requirements or features |
|  | 4.2 Security requirements | Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define authentication, verification and validation of the system. Refer to any external policies or regulations containing security issues that affect the product. |
|  | 4.3 Software Quality Attributes | Explain: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. |
| 5 | Other Requirements | Define any other requirements not covered elsewhere in the SRS. |
| Appendix A: Glossary | | Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. |
| Appendix B: Analysis Model | | Pertinent analysis models used for this project |
| Appendix C: Issues List | | This is a dynamic list of the open requirements issues. |

1. **Introduction**
   1. **Purpose of the Project**

The purpose of this project is to develop a Vulnerability Scanner Tool that identifies,

assesses, and prioritizes vulnerabilities such as SQL injection, XSS, open ports,

vulnerability in headers, and malware detection.

Key objectives of this project include:

* + - Supporting customizable scanning options for different types of vulnerabilities.
    - A learning tool for students and professionals interested in cybersecurity.
  1. **Target Beneficiary**
* **Students** with an interest in cybersecurity.
* **Small businesses and educational institutions** looking to secure their web environments.

* 1. **Project Scope**

The Vulnerability Scanner Tool is designed for use in cybersecurity environments, targeting website. It is applicable in organizations of all sizes, including educational institutions and small businesses.

1. **Project Description**
   1. **Reference Algorithm**

The tool leverages common vulnerability scanning algorithms that systematically test for:

* SQL Injection
* XSS
* Malware Detection
* Open Ports
* Insecure Headers
  1. **Data / Data Structure**

The tool will use structured data such as:

* IP addresses and URLs for target websites.
* HTTP requests/responses for vulnerability identification.
  1. **SWOT Analysis**
* Strengths: Ability to automate detection, scalability, integration with other security tools.
* Weaknesses: Potential for false positives/negatives, resource-intensive scanning processes.
* Opportunities Continuous improvement due to evolving threats
* Threats: Attackers exploiting the slightest security loopholes or limitations in detection algorithms.

* 1. **Project Features**
* Real-time scanning for vulnerabilities.
* Customizable scan options for different vulnerabilities**.** 
  1. **User Classes and Characteristics**

Identify different user groups:

* Administrators who will configure the scans.
* Security professionals/students who will analyze the results.
* Developers working on website security improvements.
  1. **Design and Implementation Constraints**
* Hardware requirements (e.g., minimum system memory).
* Specific language requirements (Python, C++, etc.).
* The tool will be developed using **Python** and may include modules such as **flask.**
* Frontend: HTML/CSS/JS
* Backend: Flask (Python)
  1. **Design Diagrams**

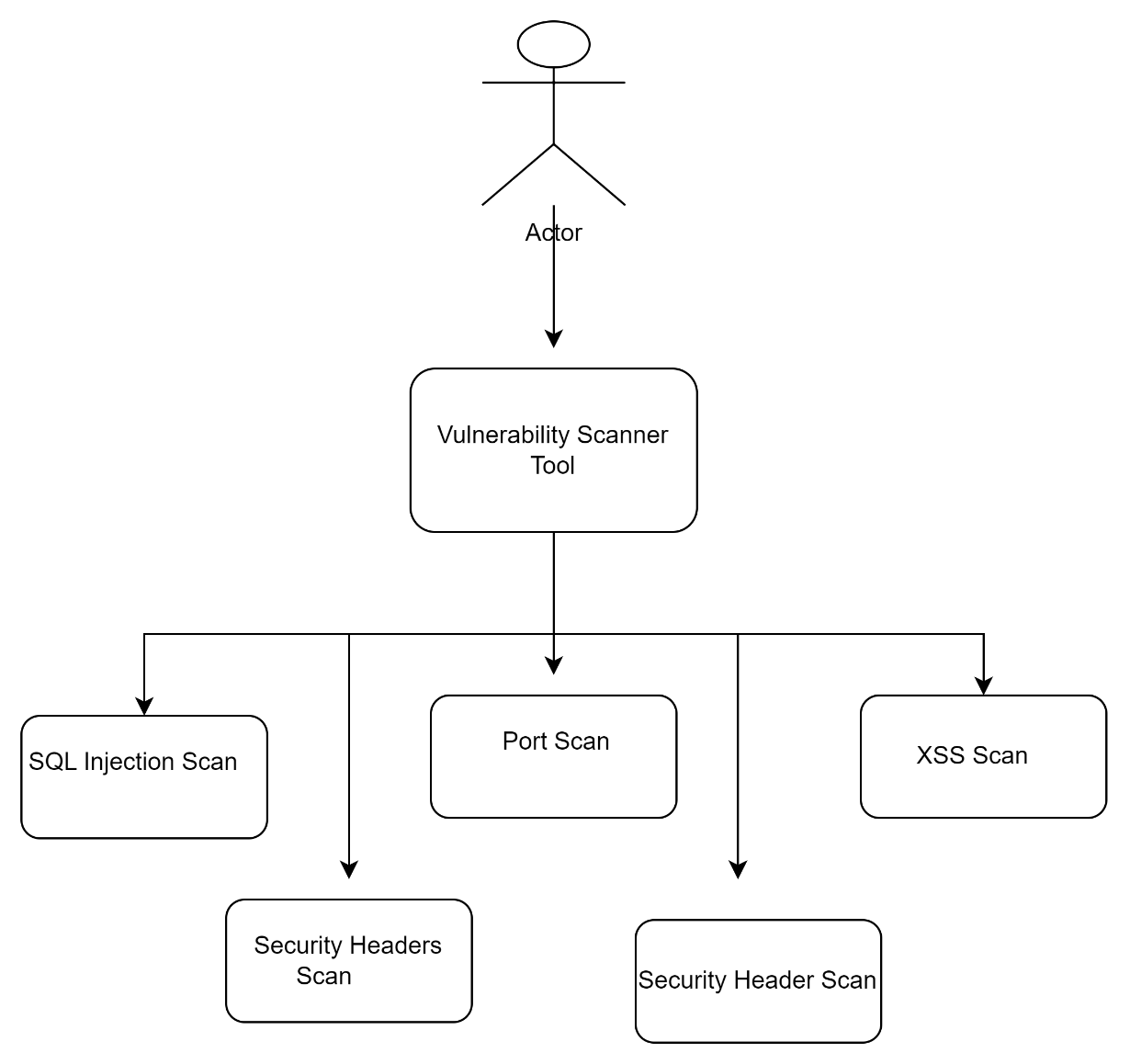


Fig1.1 Use case Diagram

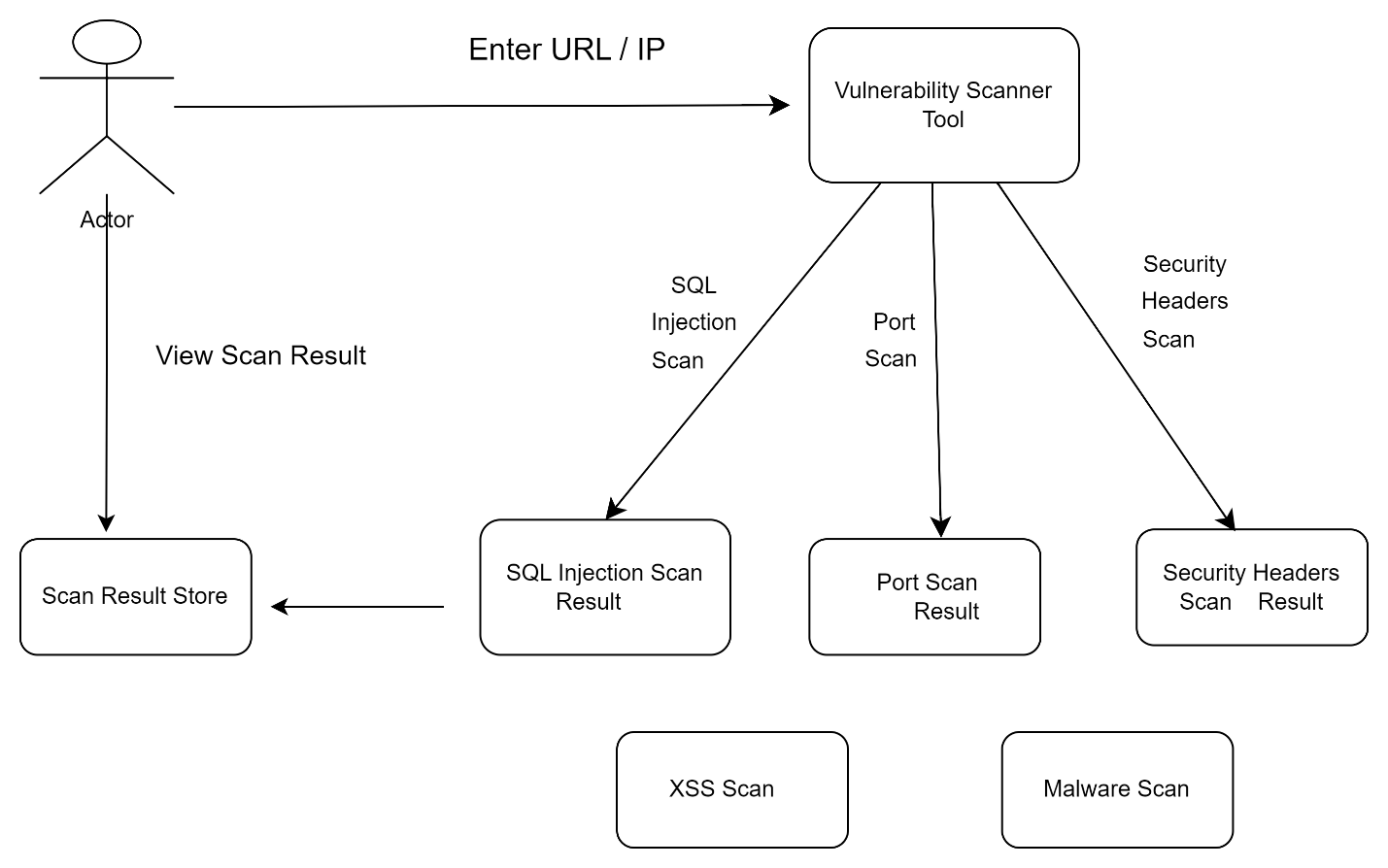


Fig1.2 Data Flow Diagram (DFD)

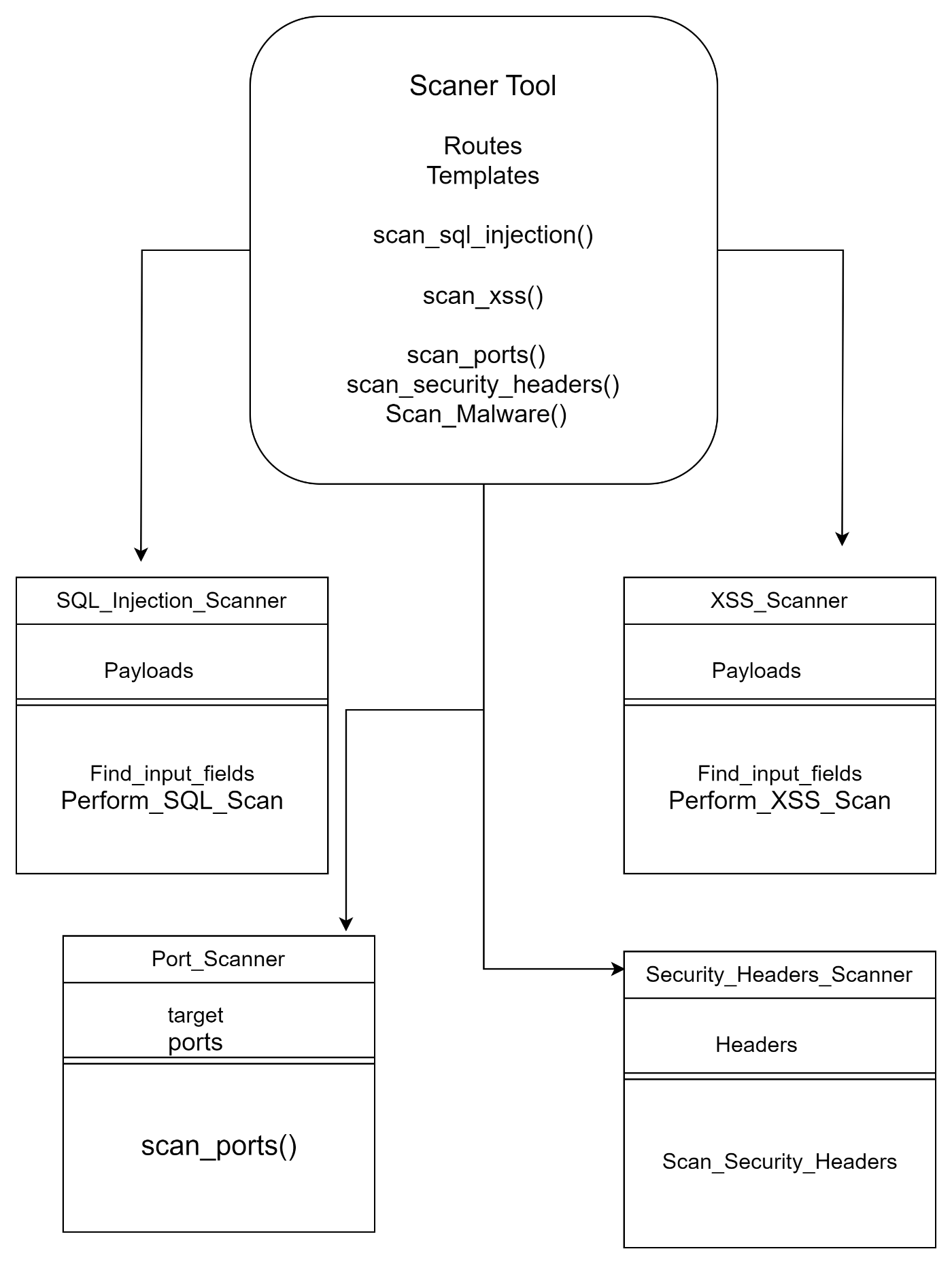


Fig:1.3 Class Diagram

* 1. **Assumptions and Dependencies**
* The project assumes access to websites for testing vulnerability scans.
* The tool depends on the availability of the OWASP Top 10 database for vulnerability classifications.

1. **System requirements:**
   1. **User Interface**

The tool will feature a Graphical User Interface (GUI) that allows users to:

* Configure scan parameters.
* View scan results in real-time.
  1. **Database Interface**

The tool may interact with a local database for storing historical scan results and configurations.

* 1. **Protocols**

Uses protocols like **HTTP** and **HTTPS** for communication, and security protocols for data transmission

1. **Non-functional Requirements**
   1. **Performance Requirements**

The tool must be able to perform vulnerability scans on single websites simultaneously, ensuring that each scan is completed within **reasonable time limits** (e.g., 10-15 minutes for small websites).

* 1. **Security Requirements**
* The scan process is authenticated and authorized.
* No sensitive data is leaked during scanning.
* Users can securely export and store reports.
  1. **Software Quality Attributes**
* Usability: Simple interface for beginners.
* Maintainability: Easily updatable with new vulnerabilities.
* Reliability: Minimal false positives/negatives.

**References**

* OWASP (Open Web Application Security Project)