Project Security

Scanning Tool

*Requirements Report*

Ghassan Awwad, Rabih Ayoub, & Mohamad Mostafa Hareb

G

**Abstract**—This document serves as a proposal for the project that will be submitted for the Software Engineering course. The project will contain a web server and a database containing static code security scans, credentials scans, and third-party component scans of all projects within a company. There would be options to either get the project from a remote repository or scan all files in a directory. The project will be dockerized and will be available to be deployed locally on any machine or VM

**Index Terms**— Static code security scanning, Credentials scans, Third-party component scans, Webapp, SAST

Adapted from the IEEE Computer Society template

—————————— ◆ ——————————

**Contents**

[1 Introduction 1](#_Toc130119676)

[2 Overall Description 2](#_Toc130119677)

[3 Specific Requirements 2](#_Toc130119678)

[4 References 2](#_Toc130119679)

# 1 Introduction

## 1.1 Purpose

This document specifies the requirements for the Project Security Scanning Tool, a web-based platform designed to automate the process of detecting and analyzing potential security threats in software code. The tool will cater to system administrators, project managers, and developers, providing role-specific functionalities to effectively manage software security.

## 1.2 Scope

The scope of this document is to define the functional and non-functional requirements of the Project Security Scanning Tool. It describes the intended features, the system's interactions with users, and the constraints under which the system will operate.

## 1.3 Definitions, Acronyms, and Abbreviations

PM: Project Manager

CVSS: Common Vulnerability Scoring System

CVE: Common Vulnerabilities and Exposures

CLI: Command Line Interface

## 1.4 Overview

I

N today's rapidly evolving digital landscape, software security has become a major concern for companies of all sizes. Security threats can arise from vulnerabilities in software code, credentials, and third-party components, and it's essential to detect and address these issues as early as possible. However, the task of manually scanning all software code for potential security threats is time-consuming, error-prone, and simply not feasible for larger projects. This is where a security scanning tool comes in - a web-based platform designed to automate the process of detecting and analyzing potential security threats in software code.

# 2 Overall Description

## 2.1 Product Perspective

The Project Security Scanning Tool is an all-encompassing, web-based platform designed to simplify and automate the process of identifying and analyzing security threats in software projects. It consists of a database containing project details, alerts, and mitigation information, as well as a website serving as the user interface for system administrators, project managers, and developers. This centralized, user-friendly platform aims to provide an efficient and effective solution for managing software security across various user roles.

By supporting a wide range of programming languages and frameworks, the tool offers a versatile solution for diverse software projects. Leveraging advanced algorithms and industry-standard security practices, the platform significantly reduces the time and effort required to maintain secure software projects.

The Project Security Scanning Tool seamlessly integrates with existing development workflows and tools, ensuring a consistent approach to software security across an organization. Its modular architecture allows for easy customization and extension, accommodating specific project requirements and security policies.

## 2.2 Product Functions

The main functions of the tool are:

1. Automated Security Scanning: Streamline the detection and analysis of potential security threats in software code using advanced algorithms and industry-standard practices.
2. Role-Based Access Control: Provide tailored features and access levels for System Administrators, Project Managers, and Developers, ensuring efficient management of software security across different roles. This includes managing the tool, users, projects, and viewing scan results.
3. Integration with Development Workflows: Seamlessly integrate with existing tools and support various programming languages and frameworks for versatile and consistent security scanning across diverse projects.
4. Centralized Dashboard: Offer a comprehensive, user-friendly interface to manage and monitor projects, scan results, user activity, and project statistics. This enables requesting and viewing scan results across user roles.
5. Issue Assignment and Tracking: Enable efficient assignment and tracking of security issues, facilitating prompt resolution and improved software security. This encompasses assigning security issues to developers, as well as addressing and resolving these issues within the platform.

## 2.3 User Characteristics

System Administrators are typically experienced IT professionals with strong technical expertise in managing software systems, networks, and security. They require a comprehensive understanding of the tool's configuration and management aspects. Project Managers possess domain knowledge in software development and project management, focusing on coordinating team efforts and ensuring timely and secure delivery of software projects. Developers have varying levels of programming expertise and familiarity with different languages and frameworks. Their primary goal is to write secure code and address any identified security issues promptly. All user types value an intuitive, user-friendly interface that facilitates efficient and effective management of software security.

## Constraints

The Project Security Scanning Tool has the following constraints:

1. Compatibility: The tool must be compatible with existing web services, Command Line Interface (CLI) binaries, and libraries, ensuring seamless integration with the current technology stack and development workflows.
2. Compliance: The tool must adhere to applicable security and data privacy regulations, ensuring that it maintains the highest standards of data protection and privacy for users and organizations.
3. Platform Independence: The tool must be platform-independent, allowing it to be used across various operating systems and environments without requiring specific hardware or software configurations.
4. Web Browser Accessibility: The tool must be accessible through modern web browsers, ensuring that users can easily access and utilize the platform without the need for additional software installations or modifications.

## 2.5 Assumptions and Dependencies

1. The target clients have access to a reliable internet connection.
2. Third-party libraries and services used in the tool will remain available and supported.

# 3 Specific Requirements

## 3.1 Functional Requirements

### System Administrators (SA):

* 1. Manage the security scanning tool and its configuration settings.
  2. Add and remove projects for scanning.
  3. Add and remove users, including project managers and developers.
  4. View all projects, results of scans, and user activity logs.
  5. Oversee the overall health and security of the software projects within the organization.

### 3.1.2 Project Managers (PM):

* 1. Add and remove developers to specific projects.
  2. Add project source code location.
  3. Request scans of their projects.
  4. View the results of scans for their projects.
  5. Assign specific security issues to individual developers.
  6. View which developer is responsible for a certain security issue.
  7. View Project Statistics.
  8. Coordinate and manage software security efforts within their projects.

### 3.1.3 Developers:

* 1. Request scans of the latest commits to the PM.
  2. View the results of scans for their own projects.
  3. Receive notifications when security issues are found in their code.
  4. Address security issues found in their own code.
  5. Collaborate with project managers and other developers to ensure the software is secure and free from vulnerabilities.

## 3.2 Non-functional Requirements

### 3.2.1 Usability

The tool's user interface should be intuitive and easy to navigate, allowing users to quickly access role-specific functionalities.

### 3.2.2 Performance

The tool should process scans and deliver results within a reasonable time frame, depending on the size and complexity of the scanned project.

### 3.2.3 Scalability

The tool should support a growing number of projects and users without significant performance degradation.

### 3.2.4 Security

The tool should ensure the security of user data, project information, and scan results.

# 4 References