**Product Requirements Document (PRD)**

**Project Name:** Container Image Vulnerability Scanner  
 **Author:** R.Aditya Raghava  
 **Date:** 02-03-2025

# **1. Introduction**

The **Container Image Vulnerability Scanner** is a security tool that scans container images for vulnerabilities. It helps **developers, DevOps teams, and security engineers** quickly identify security risks in containerized applications. The tool provides a **dashboard** displaying detected vulnerabilities, severity levels, and recommended fixes.

## Purpose

* Improve container security by detecting known vulnerabilities.
* Help users prioritize and remediate security issues efficiently.
* Provide an easy-to-use UI for managing scan results.

### 1.2 Scope

The tool will scan **Docker and Kubernetes container images** and provide a report on security vulnerabilities. It will support **manual and automated scanning** as part of a CI/CD pipeline.

#### **2. Problem Statement**

##### 2.1 The Problem

* Many container images contain security vulnerabilities that expose applications to cyber threats.
* Developers and security teams struggle to identify and fix these vulnerabilities quickly.
* There is no simple, centralized dashboard to manage vulnerability reports.

###### 2.2 The Solution

* A tool that scans container images and detects known security vulnerabilities.
* A dashboard that displays vulnerabilities categorized by severity (Critical, High, Medium, Low).
* Filtering and searching options to help users focus on critical issues.
* Fix recommendations to guide developers in resolving vulnerabilities

**3. Target Users**

The primary users of this tool include:

* Developers: Need to verify that container images are free of security issues before deployment.
* DevOps Teams: Need to integrate the tool into CI/CD pipelines for automated security scanning.
* Security Engineers: Need to monitor security vulnerabilities across containerized applications.

**4. Key Features & Functional Requirements**

| **Feature** | **Description** |
| --- | --- |
| **Container Image Scanning** | Scans container images for known vulnerabilities. |
| **Dashboard for Vulnerability Management** | Displays a **list of scanned images** and security findings. |
| **Severity Categorization** | Groups vulnerabilities into **Critical, High, Medium, Low** categories. |
| **Filtering & Searching** | Allows users to **filter by severity** and search for specific images. |
| **Fix Recommendations** | Provides **security patches and updates** for detected vulnerabilities. |
| **CI/CD Integration (Future Feature)** | Enables **automated security checks** in CI/CD pipelines. |

**5. User Flow & Wireframes**

## 5.1 User Flow

1. User uploads a container image for scanning.
2. System scans the image and generates a vulnerability report.
3. User views the report on the dashboard (severity levels, affected components).
4. User applies security patches based on fix recommendations.
5. User rescans the image to verify fixes.

## 5.2 Wireframes (Next Step)

To visualize this process, we will create wireframes (UI sketches) of:

* Dashboard Page – List of scanned images + vulnerability summary.
* Detailed View Page – Specific vulnerabilities for a selected image.
* Filter/Search UI – Options to filter vulnerabilities by severity.

## **6. Non-Functional Requirements**

* Performance: The tool should complete scans in under 30 seconds for most images.
* Scalability: Must support scanning thousands of container images without delays.
* Security: Scan results must be encrypted and protected against unauthorized access.

## **7. Deliverables for Assignment Submission**

* PRD Document
* Wireframes