Product Requirements Document: Container Image Vulnerability Scanner

1. Introduction

- **1.1 Purpose**
  This document outlines the requirements for a product that scans container images for vulnerabilities and presents the findings to users.
- **1.2 Scope**
  The product will enable users to understand which container images have vulnerabilities, the severity of those vulnerabilities, and which images need to be fixed. It will address the needs of users managing potentially thousands of images.

2. User Stories

- As a user, I want to be able to scan my container images for vulnerabilities so that I can assess their security posture.
- As a user, I need to see a list of vulnerabilities found in my container images so that I can understand the security risks.
- As a user, I need to know the severity level of each vulnerability (e.g., critical, high, medium, low) so that I can prioritize remediation efforts.
- As a user, I need to be able to identify which container images have critical or high vulnerabilities so that I can focus on fixing the most urgent issues.
- As a user, I need to be able to search and filter the vulnerability findings across a large number of images so that I can efficiently manage my repository.

3. Functional Requirements

- **3.1 Scanning**
  - The system shall allow users to scan container images from various repositories.
  - The system shall provide options to initiate on-demand scans.
  - The system shall provide the ability to schedule automated scans.
- **3.2 Vulnerability Data**
  - The system shall display a list of vulnerabilities found in each scanned image.
  - The system shall show the severity level for each vulnerability.
  - The system shall provide detailed information about each vulnerability, including its description, affected components, and potential remediation steps.
- **3.3 Image Management**
  - The system shall allow users to view a list of their container images.
  - The system shall allow users to filter images based on vulnerability status.

    - The system shall allow users to search for images by name or other criteria.
- **3.4 Reporting and Notifications**
    - The system shall generate reports summarizing the vulnerability scan results.
    - The system shall provide notifications about new vulnerabilities, especially critical or high severity ones.

4. Non-Functional Requirements

- **4.1 Performance**
    - The system shall be able to scan images and display results in a reasonable timeframe.
    - The system shall be scalable to handle a large number of images.
- **4.2 Usability**
    - The user interface shall be intuitive and easy to navigate.
    - The information shall be presented in a clear and concise manner.
- **4.3 Reliability**
    - The system shall provide accurate and consistent vulnerability scan results.
- **4.4 Security**
    - The system shall protect user data and scan results from unauthorized access.

Low-Fidelity Wireframes

Here are some low-fidelity wireframes to illustrate the user interface for the container image vulnerability scanner:

1. Image List View

- A table displaying a list of container images.
- Columns include: Image Name, Repository, Last Scan Date, Scan Status, Highest Vulnerability Severity, and Number of Vulnerabilities.
- Search bar to filter images by name.
- Filters to show images with specific vulnerability severities (e.g., Critical, High).
- Button to initiate a new scan.

2. Vulnerability Details View

- Displays details for a selected container image.
- Sections for:
    - Image Information: Name, Repository, Scan Date, Scan Status.
    - Vulnerability Summary: Overview of the number of vulnerabilities by severity.
    - Vulnerability List:
        - Table with columns: Vulnerability ID, Severity, Description, Affected

Component, Remediation Advice.

- Back button to return to the Image List View.

3. Scan Results Report

- A summary view of the scan results.
- Information on total images scanned, total vulnerabilities found, and breakdown by severity.
- Options to download the report in various formats (e.g., PDF, CSV).

These wireframes provide a basic structure for the user interface, focusing on the key functional requirements of the product.