PRD Title: Image Vulnerability Scanner

**Author:** Harsha S. Sharma

**Team:** []

**Created On:** 18-03-25, Tuesday.

|  |  |  |  |
| --- | --- | --- | --- |
| Product Manager/Analyst | Harsha S. Sharma | | |
| Team 1 |  | | |
| Team 2 |  | | |
| Team 3 |  | | |
| Team 4 |  | | |
| Approvers/Sign-Off |  | | |
| Business Domain |  | | |
| Business Unit |  | | |
| Version | V1 | | |
| Last Updated | Name  Harsha Sharma | Reasons/changes done  Document update | Date |

PM Epic: Insert link to Project Management tool epic (optional)

Status of PRD: Stakeholder Review

Contents

[Overview 3](#_Toc193185243)

[Problem 3](#_Toc193185244)

[Objectives 3](#_Toc193185245)

[Constraints 3](#_Toc193185246)

[Persona 3](#_Toc193185247)

[Use Stories + Cases 3](#_Toc193185248)

[Key Features 4](#_Toc193185249)

[Low-Fidelity Wireframes 5](#_Toc193185250)

[Development Action Items: 6](#_Toc193185251)

[Technical Considerations 7](#_Toc193185252)

[Non-Functional Considerations 7](#_Toc193185253)

[Success Metrics 7](#_Toc193185254)

[GTM Approach 7](#_Toc193185255)

[Release Criteria Open Issues 7](#_Toc193185256)

[Q&A 8](#_Toc193185257)

[Feature Timeline and Phasing 8](#_Toc193185258)

[Glossary 8](#_Toc193185259)

# Overview

The product serves as a vulnerability detector for container images and ranks/grade them based on their criticality/security threat level.

# Problem

Problem statement: To have a solution that will automate the process scanning vulnerabilities in container images. Since,

* + Users have thousands of container images in their repository.
  + These images may contain vulnerabilities, which could range from low to critical severity.
  + Users need a way to identify vulnerable images and prioritize fixes for critical or high-severity issues.

# Objectives

To build a product that helps users identify vulnerabilities in container images and prioritize fixes based on severity.

# Constraints

Roadblocks, realities, and dependencies.

1. Data approval process.
2. Data Checks
3. Data Leakage

# Persona

The target personas for this product.

|  |  |
| --- | --- |
| **Key Persona** | DevOps engineers |
| **Persona 2** | Security teams |
| **Persona 3** | Developers |
| **Persona 4** | IT Manager |

# Use Stories + Cases

Instances where various personas will use the product.

* + As a user, I want to see a list of all container images with vulnerabilities.
  + As a user, I want to filter images by severity level to focus on critical issues.
  + As a user, I want detailed information about each vulnerability to understand the risk.
  + As a user, I want recommendations for fixing vulnerabilities.
  + **Use case Example:**
    - DevOps Engineer: Responsible for maintaining container images and ensuring security compliance.
    - Security Team: Focused on identifying and mitigating risks in the containerized environment.
    - IT Manager: Needs an overview of the security posture of all container images.

# Key Features

The features to be developed are as follows.

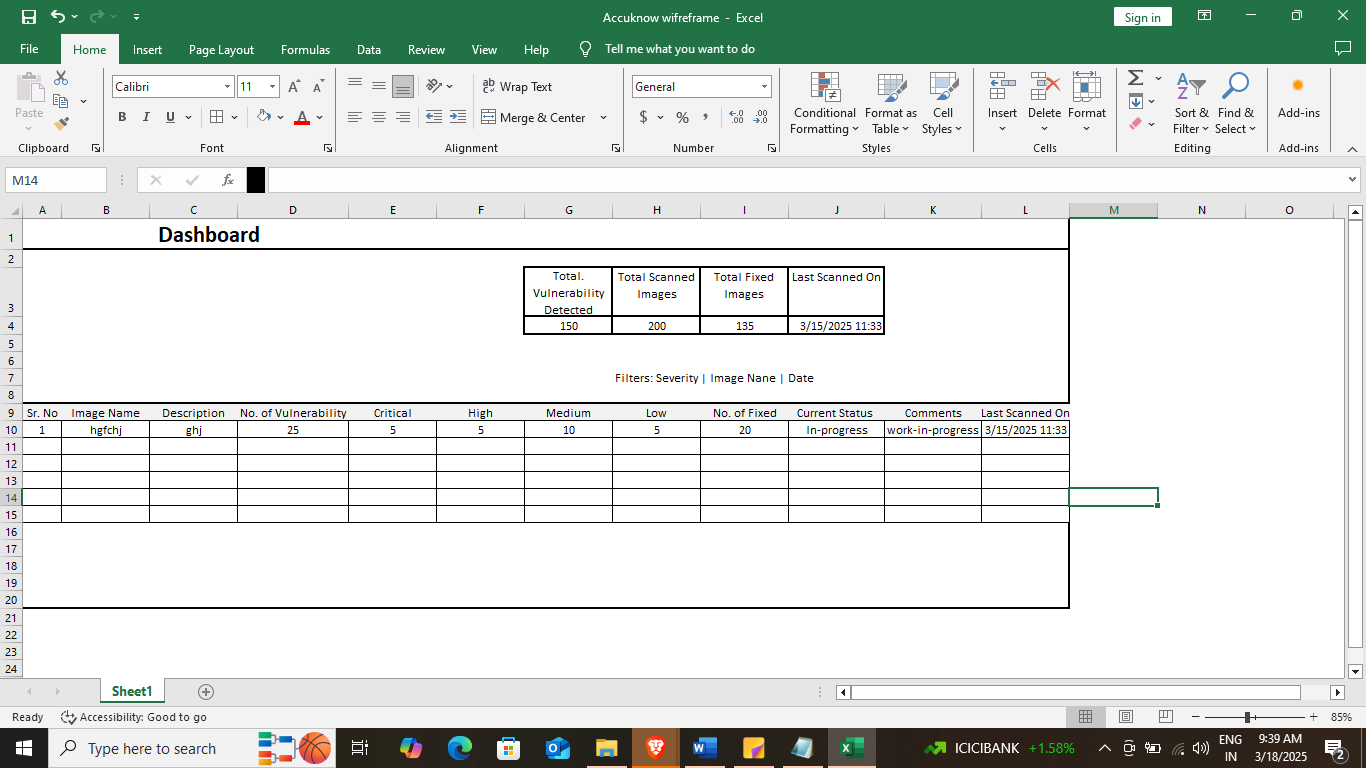
* + Image Scanning:
    - Automatically scan container images for known vulnerabilities.
    - Support bulk scanning for repositories with thousands of images.
  + Vulnerability Scanning + Severity Classification:
    - Use databases like CVE (Common Vulnerabilities and Exposures) for vulnerability identification.
    - Categorize vulnerabilities into Critical, High, Medium, and Low.
  + Dashboard:
    - Display a summary of scanned images with filters for severity, image name, repository and date.
  + Detailed Reports:
    - Provide a drill-down view for each image showing all vulnerabilities and their details (like description, ID, possible resolution).
  + Fix Recommendations:

Suggest remediation steps for each vulnerability (e.g., update dependencies, patch versions).

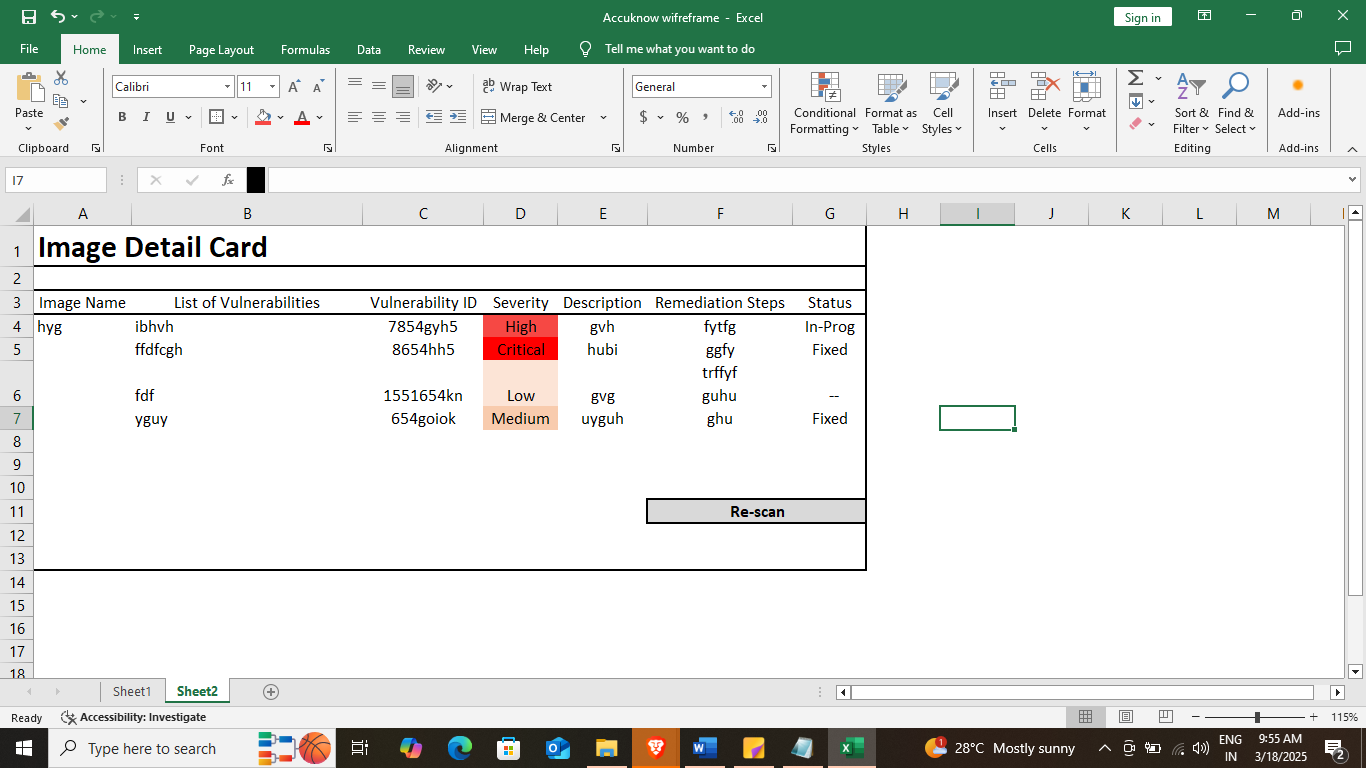
* + Alerts and Notifications:
    - Notify users when critical vulnerabilities are detected.
  + Integration:
    - Integrate with CI/CD pipelines to automate the scan and block deployments of vulnerable images.
    - Export report/finding as json, csv or pdf.

# Low-Fidelity Wireframes

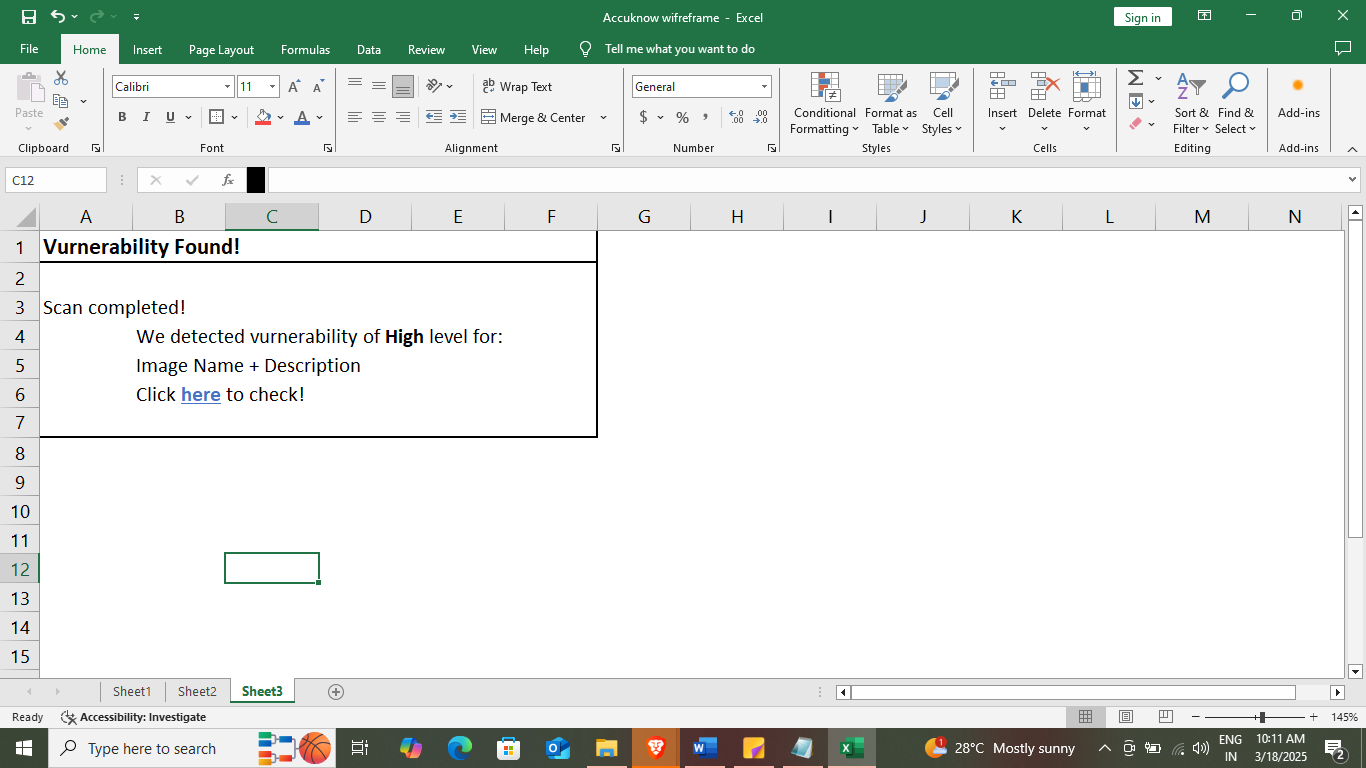
1. Dashboard:
   * A table displaying:
     + Image Name
     + Number of Vulnerabilities (by severity: Critical, High, Medium, Low)
     + Last Scanned Date
     + Status (Scanned/Fixed)
   * Filters for Severity, Image Name, and Date.
   * Sample Screen 1:
     + Hyperlink (Image Name) for image card to be provided for re-direction.



1. Image Details Page:
   * Image Name
   * List of Vulnerabilities:
     + Vulnerability ID
     + Severity
     + Description
     + Remediation Steps
   * Button to "Re-scan" the image.
   * Sample Screen 2:



1. Notifications Panel:
   * Alerts for newly detected critical vulnerabilities.
   * Links to affected images.
   * Sample screen 3:



# Development Action Items:

* Backend:
  + Build a vulnerability scanning engine using tools.
  + Store scan results in a database (e.g., SQL).
* Frontend:
  + Develop a React/Angular/vue.js-based dashboard for visualization.
  + Implement filtering, sorting, and pagination for the dashboard.
  + Export report functionality
* CI/CD Integration:
  + Create APIs to integrate with GitHub Actions, or GitLab CI/CD.
* Testing:
  + Write unit tests for backend logic and UI components.
  + Perform load testing to ensure scalability.

## Technical Considerations

* Business Impact
* Use vulnerability databases like NVD (National Vulnerability Database) or commercial tools.
* Ensure scalability to handle large repositories with thousands of images.

## Non-Functional Considerations

* Scalability: Support scanning thousands of images efficiently.
* Security: Ensure sensitive data (e.g., image contents) is not exposed during scanning.
* Usability: Simple and intuitive interface for non-technical users.

## Success Metrics

Success Metrics that indicate we are achieving our internal goals for the project.

* Percentage of images scanned regularly.
* Reduction in critical/high vulnerabilities over time.
* User satisfaction score for the product.

## GTM Approach

* **Getting all the stakeholders on board, making them understand the advantages of using this application for their business.**

## Release Criteria Open Issues

* Internal Development team testing.
* Product Team testing.
* Business user testing.
* **Additional pointers (for the release of every update/patch)**:
  + The Accuracy Improvement Plan is to be finalized and shared with stakeholders.

## Q&A

Common questions about the product along with the answers we have decided.

|  |  |  |
| --- | --- | --- |
| Asked by | Question | Answer |
|  |  |  |
|  |  |  |
|  |  |  |

## Feature Timeline and Phasing

|  |  |  |
| --- | --- | --- |
| Feature | Status | Dates |
| Dashboard | In Review |  |
| Image Detail Page | In Review |  |
| Alert/Notification | In Review |  |
| Export Report + preview | In Review |  |

## Glossary

|  |  |
| --- | --- |
| Terms | Definition |
|  |  |
|  |  |