Product Requirements Document (PRD)

Title: Container Image Vulnerability Dashboard

1. Overview

This product enables users to scan container images stored in their repositories for vulnerabilities and view actionable insights, helping security teams to identify, prioritize, and remediate risks efficiently.

2. Background

Container images are a common way to package applications and their dependencies. These images may contain known vulnerabilities that pose a security risk. Users, especially in enterprise environments, often manage thousands of container images and need a quick, actionable overview of their security posture.

3. Goals

- Detect and list vulnerabilities in container images.
- Show severity levels (e.g., Critical, High, Medium, Low).
- Allow users to filter and sort based on severity, image name, and tags.
- Enable users to prioritize and fix critical and high vulnerabilities.
- Support bulk management for large repositories.

4. Users

- DevOps Engineers
- Security Engineers
- Platform Engineers
- SREs

5. User Stories

5.1 Vulnerability Overview

As a user, I want a dashboard that shows me which container images have vulnerabilities and how severe they are so that I can quickly assess risk.

5.2 Filtering and Sorting

As a user, I want to filter and sort container images by severity, date, name, or number of vulnerabilities so that I can focus on the most critical items.

5.3 Drill-down View

As a user, I want to click into a container image to see detailed vulnerability information (CVE ID, severity, package name, fix version, etc.).

5.4 Bulk Action Support

As a user, I want to take action (e.g., trigger a rescan, mark as false positive) on multiple images to streamline my workflow.

6. Features

6.1. Dashboard View

- List of images with metadata (name, last scanned, base image, tag).
- Vulnerability summary per image (Critical, High, Medium, Low, Total).
- Sort and filter capabilities by severity, date scanned, etc.
- Quick action icons (e.g., scan now, export, view details).

6.2. Image Detail View

- Image metadata and summary.
- List of vulnerabilities with:
  - CVE ID
  - Severity
  - Affected package
  - Fix available (yes/no)
  - Link to CVE documentation
- Suggested remediation steps.

6.3. Notifications & Reports

- Email/SMS/Slack/MS Teams alerts for critical findings.
- CSV/PDF export options.
- Webhook/API integration for pipeline alerts.

7. Non-Functional Requirements

- Fast response time (under 2s for dashboard rendering).
- Scalable to handle 10k+ images.
- Secure API access with authentication and role-based permissions.
- Regular database updates from CVE sources (e.g., NVD).

8. Metrics for Success

- % of users who fix critical vulnerabilities within 24 hours of detection.
- Average time to remediation from scan.
- Number of scans initiated per week.
- Reduction in unresolved critical vulnerabilities over time.

9. Technical Considerations

- Use CVE databases (e.g., NVD, vendor-specific) for vulnerability source.
- Schedule periodic scans using cron or event-driven triggers.
- Store scan results in a searchable DB (e.g., Elasticsearch, PostgreSQL).
- Support webhook notifications for new critical vulnerabilities.

10. Wireframes (Low-Fidelity)

1. Dashboard Overview

- Top filter bar (search, severity filter, fixable toggle)
- Table/List of images
  - Image Name
  - Last Scanned
  - Total Vulnerabilities
  - Critical / High / Medium / Low counts
  - Fixable [Y/N]
  - [View Details] button

2. Image Detail View

- Image Name & metadata (tag, digest, size)
- Table of CVEs:
  - CVE ID
  - Severity
  - Package
  - Installed Version
  - Fixed Version (if any)
  - Description
  - Status (Fixable / Not Fixable)

3. Scan Trigger Page

- Manual input for image name (e.g., nginx:1.21)
- Option to bulk upload list of images
- Scan Now button
- Optional settings: Include dev dependencies, Ignore unfixed vulns
- https://www.figma.com/design/sin32FGOVw7xfnkBuIHpmu/Untitled?node-id=0-1&p=f&t=UotMswPIHU9L9Ip-0

11. Development Action Items (Optional Bonus)

Frontend

- Build dashboard component with table, filters, and search.
- Image detail page with vulnerability breakdown and navigation.
- Notification preference UI.

Backend

- API for fetching container images and vulnerability metadata.
- CVE database sync service (e.g., sync with NVD).
- Image scanning integration with tools like Trivy or Clair.
- Role-based access control (RBAC).

DevOps / Infra

- Schedule regular scans and re-scans of images.
- Container registry integration (e.g., Docker Hub, ECR).
- Logging and monitoring (e.g., Prometheus + Grafana).