Product Requirement Document

Container Image Vulnerability Scanner

Author: Himanshu Raheja

Summary

Container Image Vulnerability Scanner is a feature designed to scan container images, applications deployed on containers, and their dependencies for vulnerabilities and provide users with detailed reports on security issues. This scanner would help users point out the critical vulnerabilities in their container images quickly, leading to a shortening of remediation time and, hence making sure that the applications and its dependencies are safe.

Problem Definition

According to Datadog's 2023 report, the adoption of serverless containers continues to increase, with 46% of container organizations now running serverless containers, up from 31% two years ago. As of DockerCon 2023, Docker Hub contains approximately 15 million repositories. This adoption demonstrates that the organizations are taking critical dependency on Docker containers to run their business-critical applications and hosting thousands of images and containers.

At this scale, organizations are facing challenges in keeping track of and identifying vulnerabilities in their containerized environments. To add to these complexities, the application running on these containers may have dependencies with vulnerabilities.

Organizations are looking for automated solutions to help them proactively identify and mitigate risks in their environment.

Proposed Solution

We propose developing the Container Image Vulnerability Scanner to fill gap for better security in containerized environments. Because of the tremendous growth of containerization, protecting the container images from vulnerabilities has posed a great challenge. This tool will provide an all-around solution that can identify, classify, and remediate vulnerabilities in container images so that the overall security posture of the organizations gets improved as they roll out containerized applications

The main problems we are tackling are:

- Vulnerability Detection: Identify vulnerabilities in container images that attackers might exploit.
- Severity Assessment: Classify vulnerabilities by severity to prioritize which ones to fix first.
- Efficiency and Automation: Automate the scanning process to save time and reduce manual efforts.
- Proactive Risk Management: Provide insights and suggestions to prevent security breaches before they happen.

The solution would include:

1. **Comprehensive Scanning:** Implement a tool scanning all container images in the repository for known vulnerabilities. Use vulnerability databases such as CVE and NVD to ensure the complete detection of them.
2. **Severity Classification:** It classifies vulnerabilities based on their severity, which includes critical, high, medium, and low severity. This makes it easier for users to know their priority areas.
3. **Dashboard:** Create a summarizing dashboard for all scanned images, it should be able to point out the images with critical and high vulnerabilities. Add functionality of filter/ search to navigate with ease.
4. **Detailed Reports:** Generates detailed reports for every container image that provides all detected vulnerabilities along with description and remediation instructions.
5. **Real-Time Notifications:** Have a notification system whereby when critical or high-severity vulnerabilities are found, one should be notified immediately so they can respond.
6. **Remediation Guidance:** Offer actionable recommendations for fixing detected vulnerabilities. You can offer links to patches or updates available.
7. **Integration and Automation:** Show integration of scanner with existing CI/CD pipelines wherein scanning happens automatically during the development process and thus gets vulnerabilities caught early.
8. **Security and Compliance:** The solution shall meet the level of security and compliance for the industries involved in order to give users that peace of mind.

Success Metrics

- **User Adoption**: Track active users and usage frequency.
- **Detection and Resolution**: Measure vulnerabilities found and fixed, especially critical ones.
- **User Satisfaction**: Collect feedback and NPS scores.
- **Performance**: Monitor scanning speed and system reliability.
- **Compliance**: Ensure adherence to security standards.
- **Business Impact**: Assess cost savings and risk reduction.

User Stories

- As a user, I can see an overview of 1000s of container images from my repository and their vulnerability status by severity (critical, high).
- As a user, I can filter container images by severity to prioritize which ones to fix first.
- As a user, I can view detailed reports on each container image to understand the vulnerability and its remediation.

Functional Requirements

Priority	Requirement Type	Requirements
P0	User Interface	**Dashboard:** A list of container images and their statuses such as each image and number of vulnerable or high vulnerability level is displayed visually.
P0	User Interface	**Image Details Page:** Inventory of all the containers images with linked summaries of specific vulnerabilities in each, its risk level, brief description, and recommendations.
P0	User Interface	**Filters and Search Bar:** Filters for users to select more images based on his/her prescribed requirements.
P0	Platform	Interoperability with current container repositories. The scanner should be able to connect to existing repositories for scanning.
P0	Platform	Scanner should be able to connect to vulnerability databases, such as CVE and NVD, for listed problems.
P0	Platform	Allow user to schedule the scanning job or execute on-demand.

Non-Functional Requirements

Priority	Requirement Type	Requirements
P0	User Interface	User should be able to support listing over 10000 container images
P0	Platform	Scanner should be able to scan over 10000 contain images

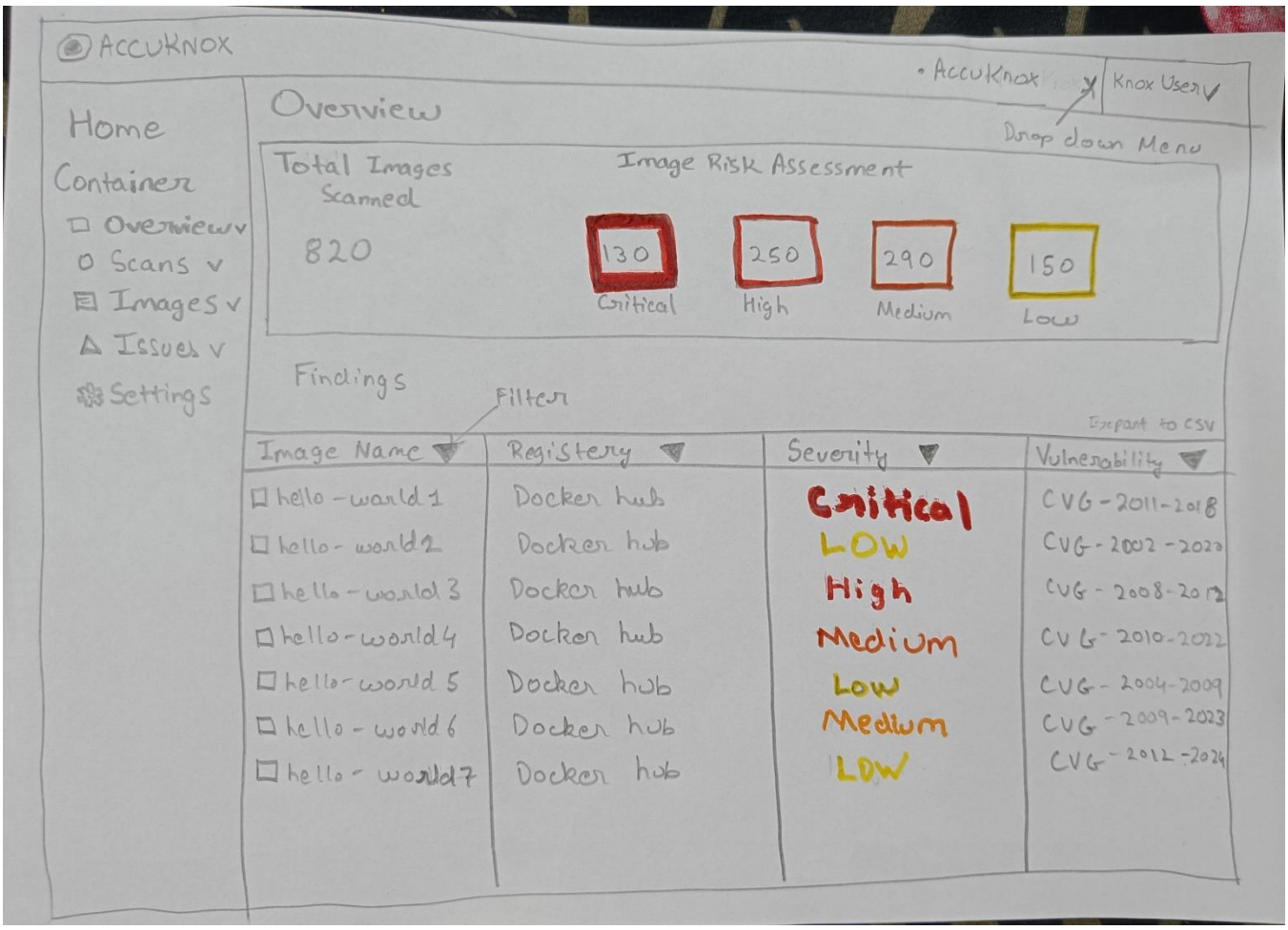
Low-fidelity Wireframes

1. **Overview page**

- As a user, I can see an overview of 1000s of container images from my repository and their vulnerability status by severity (critical, high).
- As a user, I can filter container images by severity to prioritize which ones to fix first.

User journey

1. Click on overview under “Containers” from the left menu.
2. The user will see a high – level overview of the container images.
3. Then the user can view Detailed list with image names and vulnerability counts.
4. User navigates to the dashboard.
5. User chooses to filter the list by severity, by critical/high/medium/low.
6. The list refreshes to display the filtered output.
7. The user identifies the images on the list that are at the high-risk end and should thus have priority.
8. The user can click on any image for a more detailed report.



2. Container image page

- As a user, I can view detailed reports on each container image to understand the vulnerability and its remediation.

User journey

1. The user is on the dashboard.
2. The user clicks on the image of the listed container for more details.
3. The screen provides a detailed report concerning the selected image.
4. By clicking on the table, the user sees the vulnerability ID, the severity, and the remediation step.
5. The user goes through the report to know what problems have arisen and how to correct them.