Document: Security Review of Trivy for USSF Project

Overview of Trivy

Trivy is a comprehensive and versatile open-source vulnerability and security scanner developed by Aqua Security. It is capable of scanning various resources such as container images, filesystems, and Git repositories to detect:

- **Misconfigurations**: Identifying risky or improper configurations in container images and infrastructure.
- **Secrets**: Detecting exposed sensitive information, like API keys, credentials, and tokens that might have been accidentally included.

This document will review Trivy's functionality, particularly its secret scanning, database management features, and data handling practices as applied in the USSF project.

Configuration Scanning Capabilities

Container images have configurations that can be examined using common tools like `docker inspect` and `docker history`. These configurations store critical information regarding the image's environment, setup, and credentials. Trivy enhances this by offering:

- **Misconfiguration Detection**: Trivy scans container images to find misconfigurations that could lead to security risks.
- **Secret Scanning**: It also scans configuration files for secrets like API keys, tokens, passwords, and other credentials. This is essential for avoiding accidental exposure of sensitive data.

Misconfigurations

- Trivy flags potential misconfigurations, allowing developers to address these issues before deploying the image.

Secret Scanning Process

Trivy's secret scanning feature works by converting the image's configuration into a JSON file and then scanning it for patterns that match sensitive data, such as:

- **Environment Variables**: Likely to contain credentials.
- **Built-in Rules**: Trivy leverages an extensive set of rules that look for common types of secrets such as:
    - AWS access keys

    - GCP service accounts
    - GitHub/GitLab personal access tokens
    - Slack access tokens
    - Etc.

These rules are defined and maintained in the Trivy repository. Trivy ensures that the secret data remains within the system being scanned and does not store or expose the actual secrets. It only reports their potential presence, allowing for quick review and handling of any exposures.

By default, Trivy's secret scanning is enabled. It systematically scans plaintext files in container images, file systems, and repositories for secrets using predefined or custom rules. The secret detection process applies to:

- **Container Images**: It reviews environment variables and other configuration data.
- **Filesystems**: Any sensitive files that might be stored inadvertently on a local or remote filesystem.
- **Git Repositories**: It inspects code repositories for leaked secrets, which is crucial in preventing source code leaks.

Trivy's secret scanning features make it an essential tool for securing applications and avoiding sensitive data exposure, especially in production-grade container images.

Database Usage in Trivy

Trivy relies on two types of databases to detect vulnerabilities and gather relevant information during scans:

1. **Vulnerability Database**
2. **Java Index Database**

1. Vulnerability Database

The **Vulnerability Database** contains up-to-date vulnerability information and is automatically built and maintained by Aqua Security every six hours. Trivy downloads and caches this database during execution, so the user doesn't need to worry about database maintenance.

- **Default Database**: `ghcr.io/aquasecurity/trivy-db`

Private Hosting: Users have the flexibility to host this database on their own OCI registry. This is useful for organizations that want to maintain an isolated environment.
bash

- If your registry requires authentication, it can be configured similarly to scanning private images.

2. Java Index Database

The **Java Index Database** is used when scanning JAR files and helps Trivy identify Java dependencies such as groupId, artifactId, and version. It is built daily on GitHub and automatically updated when scanning Java artifacts.

- **Default Java Database**: `ghcr.io/aquasecurity/trivy-java-db`
- **Private Hosting**: Like the vulnerability database, it is also possible to host the Java database privately.

Data Handling and Privacy

Trivy is designed with a focus on user privacy and ensuring that no sensitive information is collected or transmitted outside the system being scanned. Here's a breakdown of how Trivy handles data:

1. Data Collection

- **Local Scanning**: Trivy performs scans locally on container images, filesystems, or Git repositories. It reads configurations, environment variables, and files for scanning purposes but does not send this data externally. This ensures that sensitive data (like credentials and API keys) remain within your system or infrastructure.
- **Vulnerability Database**: Trivy automatically downloads its vulnerability database from the GitHub Container Registry (GHCR) during execution. This is a one-way download that does not involve transmitting data from the user's system to Trivy or Aqua Security.
- **Secret Data**: Trivy scans for secrets and only identifies their presence and location. **It does not store, expose, or transmit the actual content of any detected secrets**. Sensitive information remains within the scanned environment.

2. How Trivy Uses Data

- **Vulnerability Database Usage**: Trivy uses its downloaded vulnerability database to compare the software components in the images or filesystems it scans, generating a local report for the user. The scan results, including any detected vulnerabilities or misconfigurations, are not sent outside your system.
- **Java Index Database Usage**: This database is used only for identifying Java artifacts when scanning JAR files. Again, this is a local process with no data being transmitted externally.

3. No Telemetry

- **No Telemetry or Tracking**: Trivy does not include telemetry or tracking mechanisms. It does not collect or send scan details (such as image details, detected secrets, or

configuration data) back to Aqua Security or any other third party. All data remains local to the user's environment.

4. User-Controlled Configurations

- **Private Hosting Options**: Users can choose to host Trivy's databases (vulnerability or Java index) in their own OCI registry. This provides greater control over where the database is sourced from and further ensures that no data leaves the user's infrastructure.

Summary

Trivy offers a robust solution for scanning and securing container images, filesystems, and repositories. Its key features include:

- **Comprehensive Misconfiguration Detection**: Ensures that your container images comply with security best practices.
- **Secret Detection**: Detects exposed credentials and other sensitive data in configuration files.
- **Automated Database Updates**: Automatically maintains up-to-date vulnerability databases, eliminating manual intervention.
- **Privacy by Design**: Trivy does not collect or transmit user data, secrets, or scan results. All scanning is performed locally, and sensitive information remains within the user's system.