**Scanning Docker Images for Vulnerabilities: Using Trivy for Effective Security Analysis**



Trivy is an open-source **vulnerability scanner** specifically designed for containers. It is a lightweight and easy-to-use tool that helps **identify vulnerabilities in container images and filesystems.** Trivy focuses on **scanning container images for known vulnerabilities in the installed packages and libraries.**

Some key features of Trivy include:

1. **Comprehensive vulnerability database:** Trivy maintains an extensive vulnerability database, which it uses to compare the installed packages and libraries in a container image against known vulnerabilities. It supports various vulnerability databases, including CVE (Common Vulnerabilities and Exposures), Red Hat Security Data, and Ubuntu Security Notices.
2. **Fast and efficient scanning:** Trivy is designed to be fast and efficient, providing quick scan results for container images. It leverages a vulnerability detection method called static analysis, which helps expedite the scanning process.
3. **Easy integration:** Trivy can be easily integrated into your CI/CD pipeline or used as a standalone tool. It supports scanning local Docker images, remote container registries, and filesystems.
4. **Multiple output formats:** Trivy provides scan results in multiple output formats, such as JSON, table, and template, allowing you to choose the format that best suits your needs. This makes it easy to integrate Trivy with other tools or processes.
5. **Continuous updates:** Trivy is actively maintained and regularly updated with new vulnerability information. It is designed to stay up to date with the latest security advisories and vulnerability databases, ensuring accurate and relevant scan results.

*Trivy is widely used in the container security ecosystem due to its simplicity, speed, and effectiveness in identifying vulnerabilities in container images. It helps organizations improve the security of their containerized applications by enabling them to proactively address vulnerabilities before deploying containers into production environments.*

Trivy has ***scanners***that look for **security issues**, and *targets* where it can find those issues.

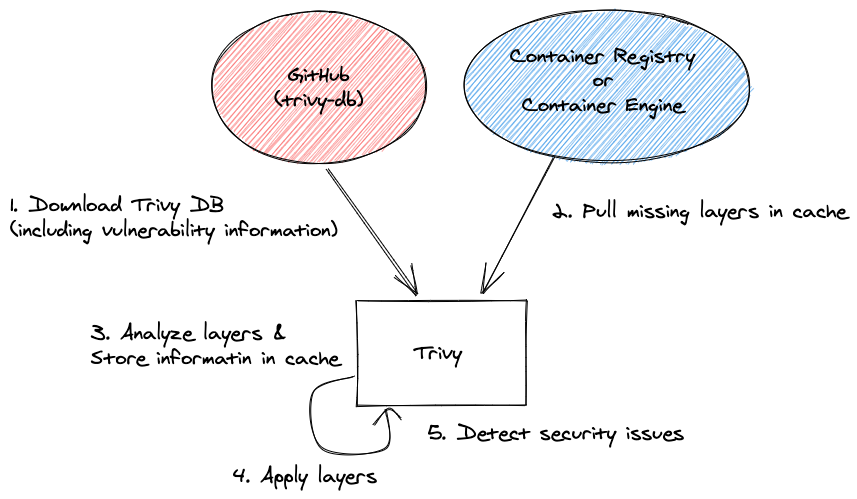
**Scanners:**

* OS packages and software dependencies in use (SBOM)
* Known vulnerabilities (CVEs)
* IaC issues and misconfigurations
* Sensitive information and secrets
* Software licenses

**Targets :**

* Container Image
* Kubernetes services
* Git Repository
* Filesystem
* Virtual Machine Image
* Kubernetes
* AWS

**A. Targeting to docker images :**



Container image scanning steps — [source](https://aquasecurity.github.io/trivy/v0.20.2/advanced/modes/standalone/)

1. **To install Trivy, follow these steps based on your operating system:**

curl -sfL https://github.com/aquasecurity/trivy/releases/latest/download/trivy\_1.9.2\_Linux-64bit.tar.gz | tar -xzf - trivy\_1.9.2/trivy

Move the extracted trivy binary to a directory in your system's $PATH:

sudo mv trivy\_1.9.2/trivy /usr/local/bin/

Test the installation by running trivy version:

trivy version

**2. Pull the Docker image:**

Yyou want to scan from a container registry. For example, if your image is named “myapp:latest,” use the following command to pull it:

docker pull myapp:latest

**3. Scan the Docker Image:**

With Trivy installed, you can now scan the Docker image for vulnerabilities.

trivy image myapp:latest

Trivy will analyze the layers of the Docker image and compare the installed packages and libraries against its vulnerability database. It will then provide a report highlighting any known vulnerabilities found.

The output of the trivy image myapp:latest command would typically provide a report of the vulnerabilities discovered in the Docker image. The report includes information such as vulnerability IDs, severity levels, affected packages, and affected versions. Here's an example of how the output might look:

vulnerability ID | PACKAGE | INSTALLED VERSION | FIXED VERSION | SEVERITY  
CVE-2021-1234 | openssl | 1.2.3 | 1.2.3-1 | CRITICAL  
CVE-2022-5678 | curl | 7.9.0 | 7.9.1 | HIGH  
CVE-2023-9012 | nginx | 1.18.0 | Not fixed | MEDIUM

In this example, the output shows three vulnerabilities with their respective details. The “vulnerability ID” column displays the unique identifier for each vulnerability, often assigned by the Common Vulnerabilities and Exposures (CVE) system. The “PACKAGE” column indicates the affected package or library. “INSTALLED VERSION” refers to the version currently installed in the Docker image, and “FIXED VERSION” indicates the recommended version that includes the security fix for the vulnerability.

The “SEVERITY” column provides an indication of the severity level assigned to each vulnerability, ranging from CRITICAL to LOW or UNKNOWN. The severity level helps prioritize the remediation efforts.

**4. Review the scan results:**

Trivy will display a report highlighting the vulnerabilities found, including details such as vulnerability IDs, severity levels, affected packages, and affected versions.

**5. Mitigate the Vulnerabilities:**

Depending on the severity and impact of the vulnerabilities, you can consider the following actions:

**5.1 Update base image:** If the vulnerabilities are related to the base image, check for any updates provided by the image maintainer. Pull the latest version of the base image and rebuild your Docker image with the updated base image.

**5.2 Update packages:** If specific packages or libraries within the image are affected by vulnerabilities, update those packages to their patched versions. Make the necessary changes to your Dockerfile or build process to ensure the updated packages are included.

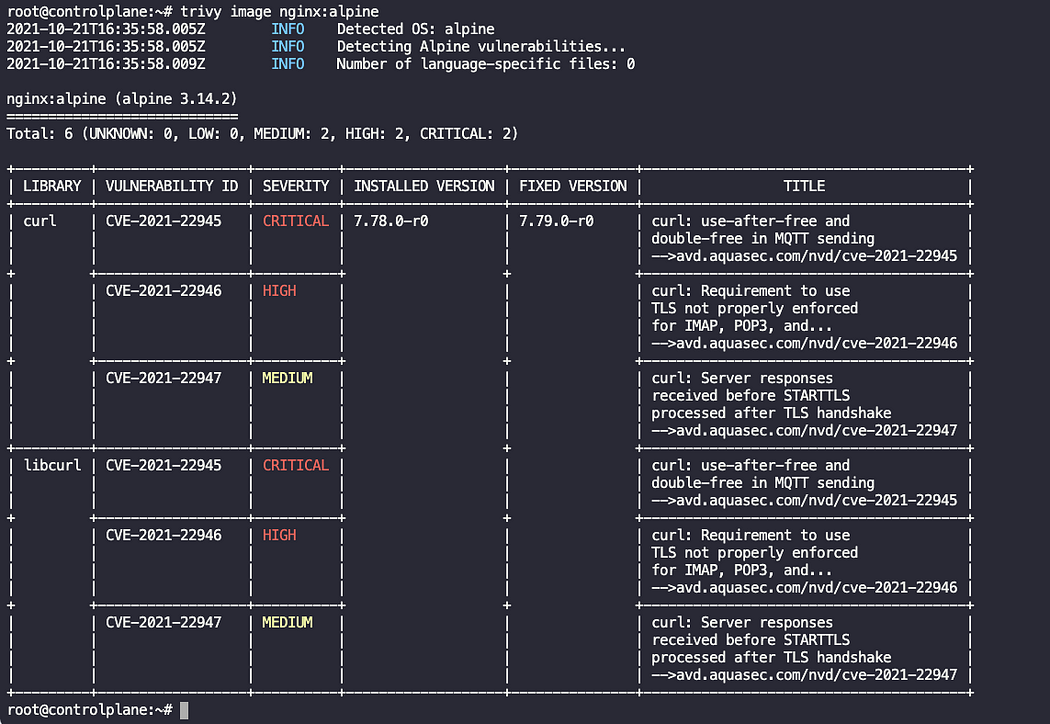
**6 Perform regular scans:** It is recommended to perform regular scans using Trivy as part of your CI/CD pipeline or periodically as part of your security practices. This helps you identify and mitigate vulnerabilities in your Docker images on an ongoing basis.

**Step by step to scan vulnerabilities from docker images — Example 02:**

docker pull nginx:alpine

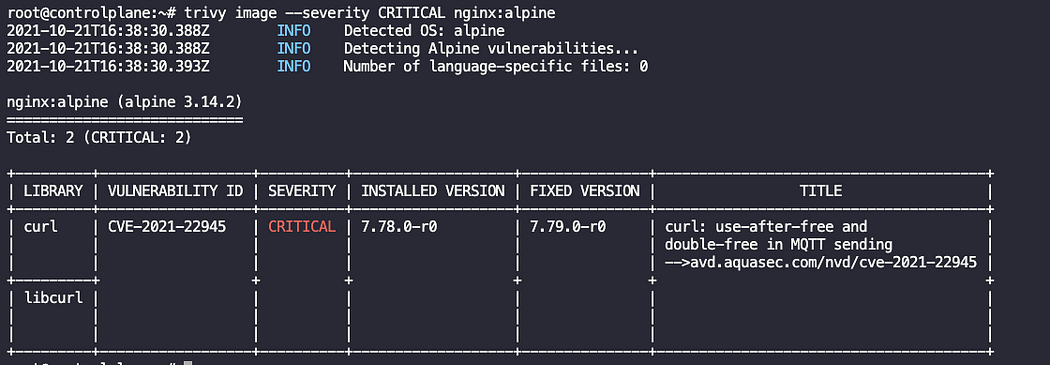
trivy image nginx:alpin  
  
2021-10-25T07:14:51.154-0700 INFO Need to update DB  
2021-10-25T07:14:51.154-0700 INFO Downloading DB...  
24.43 MiB / 24.43 MiB [------------------------------------------------------------------------------------------------------] 100.00% 6.19 MiB p/s 4s  
2021-10-25T07:15:15.099-0700 INFO Detected OS: alpine  
2021-10-25T07:15:15.099-0700 INFO Detecting Alpine vulnerabilities...  
2021-10-25T07:15:15.102-0700 INFO Number of language-specific files: 0  
2021-10-25T07:15:15.102-0700 WARN This OS version is no longer supported by the distribution: alpine 3.10.2  
2021-10-25T07:15:15.102-0700 WARN The vulnerability detection may be insufficient because security updates are not provided  
alpine:3.10.2 (alpine 3.10.2)  
=============================  
Total: 28 (UNKNOWN: 0, LOW: 4, MEDIUM: 14, HIGH: 9, CRITICAL: 1)

Out of scan, you will all kinds of severity like critical, medium, low and unknown.



If you want find out only critical severities then you can fire command:

trivy image --severity CRITICAL nginx:alpine



*You can mention multiple severities on a single line using :*

trivy image --severity CRITICAL,HIGH nginx:alpine

