sysdig

The Top 5 Best Practices for Image Scanning

How do you manage container security risk without slowing down application delivery?

One approach that can help address this challenge is image scanning, the process of analyzing the contents and build process of a container image to detect security issues, vulnerabilities, and bad practices. Image scanning can be embedded into a DevOps workflow to act as a first line of defense, detecting and blocking vulnerabilities before they can be exploited.

To help make security a seamless part of your workflow, we've rounded up our top five best practices to get you started on adopting an effective container image scanning strategy.

01

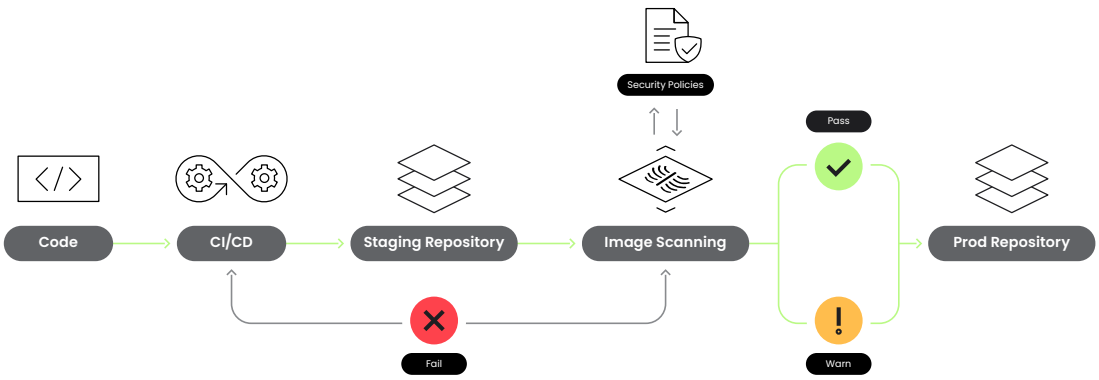
Bake image scanning into your CI/CD pipeline

When building container images, you should always scan them before publishing.

You can add an extra step for image scanning to the CI/CD pipeline you're already building for your DevOps workflows.

The basics of image scanning on a CI/CD pipeline are as follows: After your code is tested and built, you can push images to a staging repository instead of pushing them to the production repository. Then, you can run your image scanning tool. These tools usually return a report listing the different issues found, assigning different severities to each one. You can check these image scanning results in your CI/CD pipeline and fail the build if there is any critical issue.

Keep in mind that automation is key. By automating security into your CI/CD pipelines, you can catch vulnerabilities before they enter your registry, so issues never reach production.



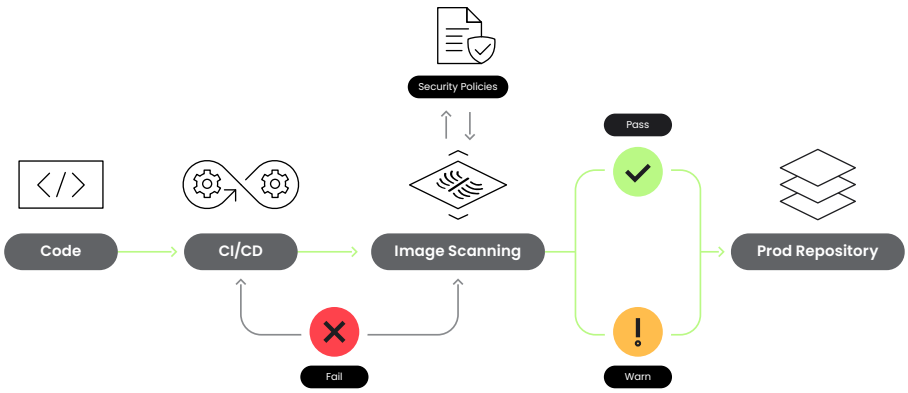
02

Adopt inline scanning to keep control of your privacy

Traditionally, image scanning in a CI/CD pipeline involves a staging repository. But what if your image contains some credentials by mistake? Those credentials could reach the wrong hands and end up being leaked.

Going a step further, you can implement inline image scanning, which scans your images directly from your CI/CD pipeline without needing a staging repository.

Only the scan metadata is sent to your scanning tool, helping you keep control of your privacy.

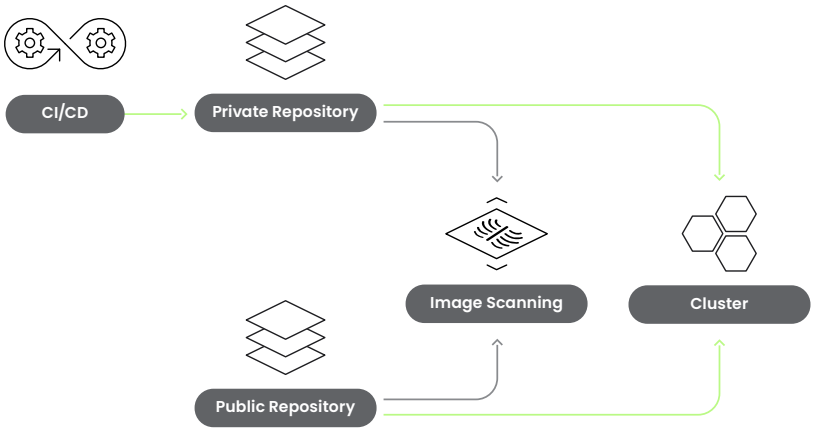


03

Regularly scan images in your container registry

By regularly scanning the images in a container registry, you can identify new vulnerabilities that affect previously scanned images.

Since you will be pulling images from your registries, it's important that you scan them so that you can identify any security risks that may arise after an image has already been checked into a registry.

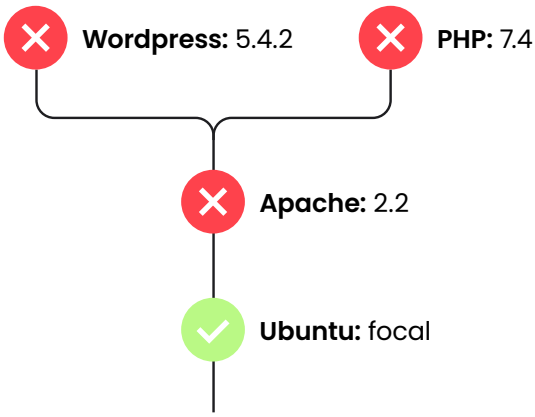


04

Scan for OS vulnerabilities

In general, the lighter the image, the better. A lighter image means faster builds, faster scans, and fewer dependencies with potential vulnerabilities.

New Docker images are usually built off of an existing base image. This base image is defined by the FROM statement in the image Dockerfile. The result is a layered architecture design that saves a lot of time in the most common tasks. For example, when it comes to image scanning, you only need to scan a base image once. If a parent image is vulnerable, any other images built on top of that one will be vulnerable too.



Even if you didn't introduce a new vulnerability in your image, it will be susceptible to those in the base image.

That's why your scanning tool should actively track vulnerability feeds for known vulnerable images and notify you if you're using an affected image.

05

Scan for vulnerabilities in third-party libraries

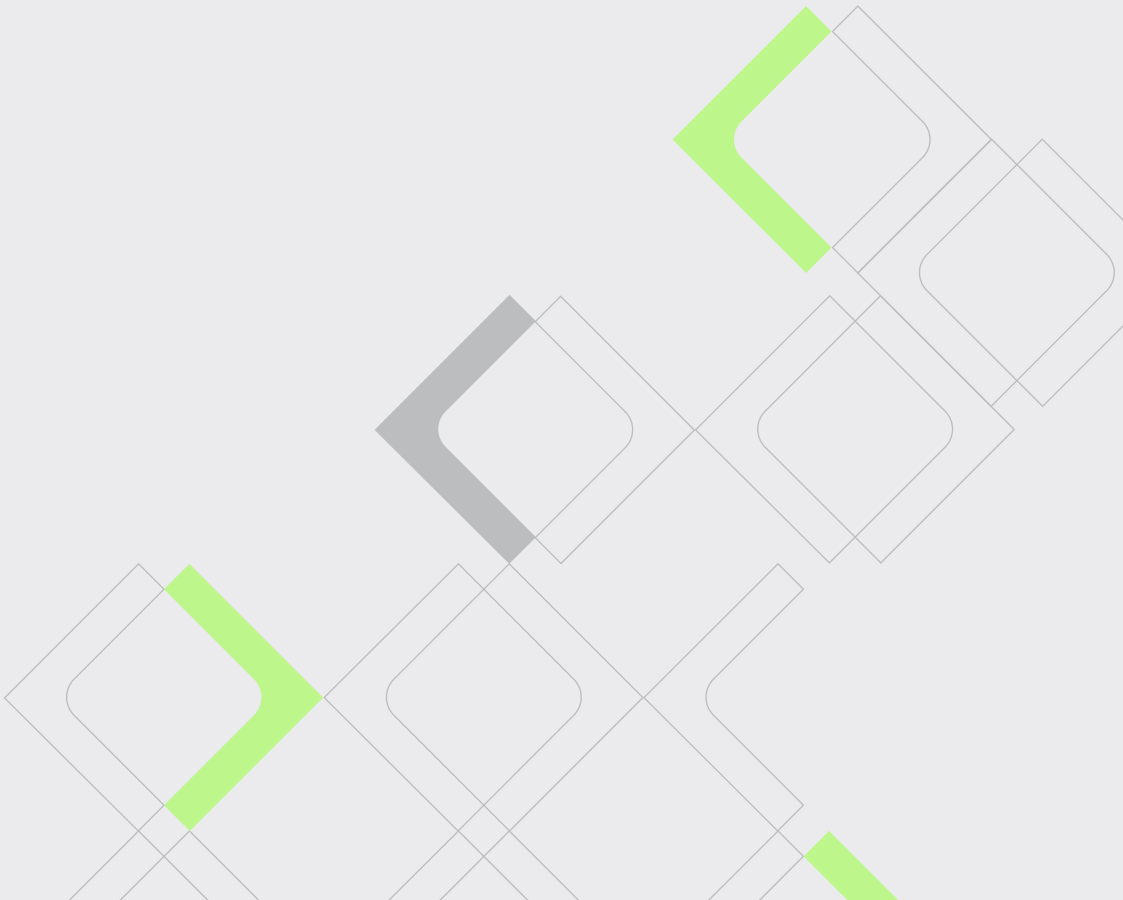
Applications use a lot of libraries — so many that libraries end up adding more lines of code than the actual code your team writes. This means you need to be aware of vulnerabilities not only in your own code, but also in all of its dependencies.

Luckily, those vulnerabilities are well tracked in the same vulnerability feeds that your scanner uses to warn you about OS vulnerabilities. Not all tools go as deep as to scan the libraries in your images, so make sure your image scanner digs deep enough and warns you about these vulnerabilities.

Image scanning is the first line of defense in a secure DevOps workflow. Following image scanning best practices will help you detect issues before they have a chance to become a real problem — all without slowing you down.

Want to learn more? Check out our ebook Securing the Cloud: A Guide to Effective Vulnerability Management:

DOWNLOAD NOW →



sysdig

BRIEF

PB-041 REV. A 2/25