Secrets
ABAP
Apex
AzureResourceManager
C
C#
C++
CloudFormation
COBOL
CSS
Dart
Docker
Flex
Go
HTML
Java
JavaScript
JCL
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

Docker static code analysis

Unique rules to find Vulnerabilities, Security Hotspots, and Code Smells in your DOCKER code

All rules 44 | Vulnerability 4 | Bug 4 | Security Hotspot 15 | Code Smell 21

Tags ⌄ | Impact ⌄ | Clean code attribute ⌄ | Search by name...

Instructions should be upper case
Code Smell

Allowing non-root users to modify resources copied to an image is security-sensitive
Security Hotspot

Automatically installing recommended packages is security-sensitive
Security Hotspot

Running containers as a privileged user is security-sensitive
Security Hotspot

Delivering code in production with debug features activated is security-sensitive
Security Hotspot

Use ADD instruction to retrieve remote resources
Code Smell

Arguments in long RUN instructions should be sorted
Code Smell

Track uses of "TODO" tags
Code Smell

Descriptive labels are mandatory
Code Smell

Use digest to pin versions of base images
Code Smell

Dockerfile parsing failure
Code Smell

Pulling an image based on its digest is security-sensitive
Security Hotspot

Pulling an image based on its digest is security-sensitive

Analyze your code

Responsibility - Trustworthy | Security ✓

Security Hotspot | Minor | dockerfile cwe

This rule is deprecated; use S6596 instead.

A container image digest uniquely and immutably identifies a container image. A tag, on the other hand, is a mutable reference to a container image.

This tag can be updated to point to another version of the container at any point in time.
In general, the use of image digests instead of tags is intended to keep determinism stable within a system or infrastructure for reliability reasons.

The problem is that pulling such an image prevents the resulting container from being updated or patched in order to remove vulnerabilities or significant bugs.

Ask Yourself Whether

- You expect to receive security updates of the base image.

There is a risk if you answer yes to this question.

Recommended Secure Coding Practices

Containers should get the latest security updates. If there is a need for determinism, the solution is to find tags that are not as prone to change as `latest` or shared tags.

To do so, favor a more precise tag that uses semantic versioning and target a major version, for example.

Sensitive Code Example

```
FROM mongo@sha256:8eb8f46e22f5ccf1feb7f0831d02032b187781b178cb971cd1222556a6cee9d1

RUN echo ls
```

Compliant Solution

Here, mongo:6.0 is better than using a digest, and better than using a more precise version, such as 6.0.4, because it would prevent 6.0.5 security updates:

```
FROM mongo:6.0

RUN echo ls
```

See

- Docker-Lock
- Skaffold, kpt, digester, kustomize, gke-deploy, ko, and Bazel
- GKE, Using Container Image Digests
- OpenShift, Builds and Image Streams

Available In:

sonarlint | sonarcloud | sonarqube

© 2008-2024 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE, and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrigh
property of their respective owners. All rights are expressly reserved.