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...

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

Automatically installing recommended packages is security-sensitive

Analyze your code

Responsibility - Trustworthy | Security

Security Hotspot | Minor | dockerfile

Installing recommended packages automatically can lead to vulnerabilities in the Docker image.

Potentially unnecessary packages are installed via a known Debian package manager. These packages will increase the attack surface of the created container as they might contain unidentified vulnerabilities or malicious code. Those packages could be used as part of a broader supply chain attack. In general, the more packages are installed in a container, the weaker its security posture is.

Depending on the introduced vulnerabilities, a malicious actor accessing such a container could use these for privilege escalation. Removing unused packages can also significantly reduce your Docker image size.

To be secure, remove unused packages where possible and ensure images are subject to routine vulnerability scans.

Ask Yourself Whether

- Container vulnerability scans are not performed.

There is a risk if you answered yes to the question.

Recommended Secure Coding Practices

- Avoid installing package dependencies that are not strictly required.

Sensitive Code Example

```
FROM ubuntu:22.04

# Sensitive
RUN apt install -y build-essential

# Sensitive
RUN apt-get install -y build-essential

# Sensitive
RUN aptitude install -y build-essential
```

Compliant Solution

```
FROM ubuntu:22.04

RUN apt --no-install-recommends install -y build-essential

RUN apt-get --no-install-recommends install -y build-essential

RUN aptitude --without-recommends install -y build-essential
```

See

- Debian Documentation - Binary Dependencies
- Ubuntu Blog - Container size reduction

Available In:

sonarlint | sonarcloud | sonarqube