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...

Package update should not be executed without installing it
Code Smell

Cache should be cleaned after package installation
Code Smell

Deprecated instructions should not be used
Code Smell

Consent flag should be set to avoid manual input
Code Smell

Environment variables should not be unset on a different layer than they were set
Code Smell

Expanded filenames should not be options
Code Smell

Double quote to prevent globbing and word splitting
Code Smell

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

Package update should not be executed without installing it

Analyze your code

Intentionality - Efficient | Maintainability

Code Smell | Major

Running `update` of your package manager in a single `RUN` instruction stores the cache index in the file system. This cache is not needed for the installed software to work properly.

Why is this an issue? | How can I fix it? | More Info

Code examples

Noncompliant code example

```
RUN apk update
RUN apt-get update
RUN aptitude update
```

Here each line represents an update command for the most popular package managers. Each of them stores the cache index in the newly created layer.

Compliant solution

```
RUN apk update && apk add ...
RUN apt-get update && apt-get install ...
RUN aptitude update && aptitude install ...
```

Here in each line after the update, the package installation is executed. However, it happens in single `RUN` instruction so only one layer is created. After installing all packages the cleanup of the cache index should be done. For more details please see rule S6587.

How does this work?

Each execution of `RUN` instruction creates a new layer in Docker. If a single command `apt-get update` or equivalent is executed, the cache is stored in the new layer. This increases the size of the final image. Even removing those cache in the next `RUN` instruction doesn't decrease the size of the final image. This overhead is not needed in the Docker image. Updating the cache and installing packages should be executed in one step (one `RUN` instruction).

Available In:

sonarlint | sonarcloud | sonarqube