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...

A space before the equal sign in key-value pair may lead to unintended behavior
Bug

Allowing downgrades to a clear-text protocol is security-sensitive
Security Hotspot

Allowing shell scripts execution during package installation is security-sensitive
Security Hotspot

Using host operating system namespaces is security-sensitive
Security Hotspot

Setting loose POSIX file permissions is security-sensitive
Security Hotspot

Reduce the amount of consecutive RUN instructions
Code Smell

Prefer COPY over ADD for copying local resources
Code Smell

WORKDIR instruction should only be used with absolute path
Code Smell

Too long RUN instruction should be split into multiple lines
Code Smell

Prefer Exec form for ENTRYPOINT and CMD instructions
Code Smell

"WORKDIR" instruction should be used instead of "cd" commands
Code Smell

A space before the equal sign in key-value pair may lead to unintended behavior

Analyze your code

Intentionality - Logical | Reliability

Bug | Major

In places where key-value pairs are used, a space before the equal sign may lead to unintended behavior.

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

Code examples

Noncompliant code example

```
ENV BUILD_NUMBER =1
RUN echo $BUILD_NUMBER
```

This will lead to print `=1`, which is not expected.

Compliant solution

```
ENV BUILD_NUMBER=1
RUN echo $BUILD_NUMBER
```

This will lead to print `1`, which is expected.

Noncompliant code example

```
ENV USER = bob MODE = all
RUN echo $USER
```

This will lead to print `= bob MODE = all`, which is not expected.

Compliant solution

```
ENV USER=bob MODE=all
RUN echo $USER
```

This will lead to print `bob`, which is expected.

How does this work?

The `ENV` instruction allows alternative syntax `ENV <key> <value>` and in case of space before equal sign, the `=1` is evaluated as value. The `LABEL` instruction will be also evaluated to `=1`. The `ARG` instruction will cause the build error.

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 Sc S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.