sonar RULES

Products

- 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...

- 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
- 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

Double quote to prevent globbing and word splitting

Analyze your code

Intentionality - Complete | Maintainability

Code Smell | Major

Variable references should be encapsulated with double quotes to avoid globbing and word splitting.

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

Surround variable reference with double quotes.

Code examples

Noncompliant code example

This example demonstrates pathname expansion using the `echo` command:

```
RUN test="command t*.sh" && echo $test
```

Suppose this code is executed in a directory that contains two files: `temp1.sh` and `temp2.sh`. This code will print `"command temp1.sh temp2.sh"`, as `*` is substituted with matching files in the current folder.

This example demonstrates word splitting using the `echo` command:

```
RUN test=" Hello World " && echo $test
```

This code will print `"Hello World"`, omitting the leading and trailing whitespaces.

Compliant solution

This example demonstrates pathname expansion using the `echo` command, which will print `"command t*.sh"` as intended:

```
RUN test="command t*.sh" && echo "$test"
```

This example demonstrates word splitting using the `echo` command, which will print `" Hello World "` as intended:

```
RUN test=" Hello World " && echo "$test"
```

Available In:

sonarlint sonarcloud sonarqube