





































-  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...



 Code Smell

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

 Code Smell

Expanded filenames should not become 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

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

Consent flag should be set to avoid manual input


Analyze your code

Intentionality - Logical

Maintainability ⬆

 Code Smell

 Major ⓘ

 shell

Various package and software management applications require manual input for execution confirmation by default. This confirmation is usually required when installing, updating, or removing programs and packages. General consent can be given to execute the command without manual input.

Why is this an issue?

How can I fix it?

More Info

Suppose a package manager invocation is part of a script that is executed automatically, and non-interactive mode is not enabled. Then, execution is aborted because there is no confirming manual input. As a result, instructions, such as installation or update of packages, cannot be performed in an automated way. This applies, among others, to the package manager used in Debian-based systems, Advanced Package Tool (APT).

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 copyrights are the property of their respective owners. All rights are expressly reserved.

Sonar helps developers write Clean Code.
[Privacy Policy](#) | [Cookie Policy](#)

