





































-  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...

Permissions of sensitive mount points should be restrictive

 Vulnerability

Server certificates should be verified during SSL/TLS connections

 Vulnerability

Weak SSL/TLS protocols should not be used

 Vulnerability

Disabling builder sandboxes is security-sensitive

 Security Hotspot

Exposing administration services is security-sensitive

 Security Hotspot

Recursively copying context directories is security-sensitive

 Security Hotspot

Using clear-text protocols is security-sensitive

 Security Hotspot

Using weak hashing algorithms is security-sensitive

 Security Hotspot

Malformed JSON in Exec form leads to unexpected behavior

 Bug

Dockerfile should only have one ENTRYPOINT and CMD instruction

 Bug

Access variable which is not available in the current scope

 Bug

## Permissions of sensitive mount points should be restrictive

Analyze your code

Intentionality - Complete

Security 

 Vulnerability

 Critical



 dockerfile cwe

For mounts types `secret` and `ssh`, Dockerfile’s `RUN` instruction supports a `mode` option for setting permissions. If you set this mode so that any user of the operating system can access the mount, it is vulnerable to leaks.

Why is this an issue?

How can I fix it?

More Info

## Documentation

- Dockerfile Reference - [RUN --mount=type=secret](#)
- Red Hat - [Linux file permissions explained](#)

## Standards

- CWE - [CWE-732 - Incorrect Permission Assignment for Critical Resource](#)
- STIG Viewer - [Application Security and Development: V-222430](#) - The application must execute without excessive account permissions.

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](#)

