- 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...

- 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
- Dockerfile parsing failure
  Code Smell
- Pulling an image based on its digest is security-sensitive
  Security Hotspot

Delivering code in production with debug features activated is security-sensitive

Analyze your code

Consistency - Conventional | Security ✓

Security Hotspot | Minor | cwe error-handling debug user-experience

Development tools and frameworks usually have options to make debugging easier for developers. Although these features are useful during development, they should never be enabled for applications deployed in production. Debug instructions or error messages can leak detailed information about the system, like the application's path or file names.

Ask Yourself Whether

- The code or configuration enabling the application debug features is deployed on production servers or distributed to end users.
- The application runs by default with debug features activated.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

Do not enable debugging features on production servers or applications distributed to end users.

Sensitive Code Example

```
FROM example
# Sensitive
ENV APP_DEBUG=true
# Sensitive
ENV ENV=development
CMD /run.sh
```

Compliant Solution

```
FROM example
ENV APP_DEBUG=false
ENV ENV=production
CMD /run.sh
```

See

- OWASP - Top 10 2021 Category A5 - Security Misconfiguration
- OWASP - Top 10 2017 Category A3 - Sensitive Data Exposure
- CWE - CWE-489 - Active Debug Code
- CWE - CWE-215 - Information Exposure Through Debug Information

Available In:

sonarlint | sonarcloud | sonarqube