





































-  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...



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

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

## Exposing administration services is security-sensitive

Analyze your code

Intentionality - Complete

Security



 Security Hotspot

 Critical



 dockerfile cwe

Exposing administration services can lead to unauthorized access to containers or escalation of privilege inside of containers.

A port that is commonly used for administration services is marked as being open through the `EXPOSE` command. Administration services like SSH might contain vulnerabilities, hard-coded credentials, or other security issues that increase the attack surface of a Docker deployment. Even if the ports of the services do not get forwarded to the host system, by default they are reachable from other containers in the same network. A malicious actor that gets access to one container could use such services to escalate access and privileges.

Removing the `EXPOSE` command is not sufficient to be secure. The port is still open and the service accessible. To be secure, no administration services should be started. Instead, try to access the required information from the host system. For example, if the administration service is included to access logs or debug a service, you can do this from the host system instead. Docker allows you to read out any file that is inside of a container and to spawn a shell inside of a container if necessary.

Ask Yourself Whether

- The container starts an administration service.

There is a risk if you answered yes to the question.

Recommended Secure Coding Practices

- Do not start SSH, VNC, RDP or similar administration services in containers.

Sensitive Code Example

```
FROM ubuntu:22.04
# Sensitive
EXPOSE 22
CMD [ "/usr/sbin/sshd", "-f", "/etc/ssh/sshd_config", "-D" ]
```

See

- CWE - [CWE-284 - Improper Access Control](#)
- [Dockerfile reference](#) - EXPOSE

Available In:

sonarlint



sonarcloud



sonarqube

