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...

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

A space before the equal sign in key-value pair may lead to unintended behavior
Bug

Server certificates should be verified during SSL/TLS connections

Analyze your code

Responsibility - Trustworthy | Security

Vulnerability | Critical | cwe privacy ssl

This vulnerability makes it possible that an encrypted communication is intercepted.

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

Code examples

The following code contains examples of disabled certificate validation.

Noncompliant code example

```
FROM ubuntu:22.04

# Noncompliant
RUN curl --insecure -O https://expired.example.com/downloads/install.sh
```

Compliant solution

```
FROM ubuntu:22.04

RUN curl -O https://new.example.com/downloads/install.sh
```

How does this work?

Addressing the vulnerability of disabled TLS certificate validation primarily involves re-enabling the default validation.

To avoid running into problems with invalid certificates, consider the following sections.

Using trusted certificates

If possible, always use a certificate issued by a well-known, trusted CA for your server. Most programming environments come with a predefined list of trusted root CAs, and certificates issued by these authorities are validated automatically. This is the best practice, and it requires no additional code or configuration.

Working with self-signed certificates or non-standard CAs

In some cases, you might need to work with a server using a self-signed certificate, or a certificate issued by a CA not included in your trusted roots. Rather than disabling certificate validation in your code, you can add the necessary certificates to your trust store.

Available In:

sonarlint | sonarcloud | sonarqube