- 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...

- Credentials should not be hard-coded
  Vulnerability
- Using ENV or ARG to handle secrets is security-sensitive
  Security Hotspot
- 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

Weak SSL/TLS protocols should not be used

Analyze your code

Responsibility - Trustworthy | Security

Vulnerability | Critical | cwe privacy

This vulnerability exposes encrypted data to a number of attacks whose goal is to recover the plaintext.

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

Encryption algorithms are essential for protecting sensitive information and ensuring secure communications in a variety of domains. They are used for several important reasons:

- Confidentiality, privacy, and intellectual property protection
- Security during transmission or on storage devices
- Data integrity, general trust, and authentication

When selecting encryption algorithms, tools, or combinations, you should also consider two things:

1. No encryption is unbreakable.
2. The strength of an encryption algorithm is usually measured by the effort required to crack it within a reasonable time frame.

For these reasons, as soon as cryptography is included in a project, it is important to choose encryption algorithms that are considered strong and secure by the cryptography community.

To provide communication security over a network, SSL and TLS are generally used. However, it is important to note that the following protocols are all considered weak by the cryptographic community, and are officially deprecated:

- SSL versions 1.0, 2.0 and 3.0
- TLS versions 1.0 and 1.1

When these unsecured protocols are used, it is best practice to expect a breach: that a user or organization with malicious intent will perform mathematical attacks on this data after obtaining it by other means.

What is the potential impact?

After retrieving encrypted data and performing cryptographic attacks on it on a given timeframe, attackers can recover the plaintext that encryption was supposed to protect.

Depending on the recovered data, the impact may vary.

Below are some real-world scenarios that illustrate the potential impact of an attacker exploiting the vulnerability.

Additional attack surface

By modifying the plaintext of the encrypted message, an attacker may be able to trigger additional vulnerabilities in the code. An attacker can further exploit a system to obtain more information.
Encrypted values are often considered trustworthy because it would not be possible for a third party to modify them under normal circumstances.

Breach of confidentiality and privacy

When encrypted data contains personal or sensitive information, its retrieval by an attacker can lead to privacy violations, identity theft, financial loss, reputational damage, or unauthorized access to confidential systems.

In this scenario, the company, its employees, users, and partners could be seriously affected.

The impact is twofold, as data breaches and exposure of encrypted data can undermine trust in the organization, as customers, clients and stakeholders may lose confidence in the organization's ability to protect their sensitive data.

Legal and compliance issues

In many industries and locations, there are legal and compliance requirements to protect sensitive data. If encrypted data is compromised and the plaintext can be recovered, companies face legal consequences, penalties, or violations of privacy laws.

Available In:

sonarlint | sonarcloud | sonarqube