

© C++

CloudFormation

COBOL COBOL

C# C#

∃ css

Flex

€O Go

THTML

Java

Js JavaScript

Kotlin

Kubernetes

ó Objective C

PHP

PL/I

PL/SQL

襣 Python

RPG RPG

Ruby

Scala

Swift

Terraform

Text

TS TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



C++ static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C++ code

All 578
rules

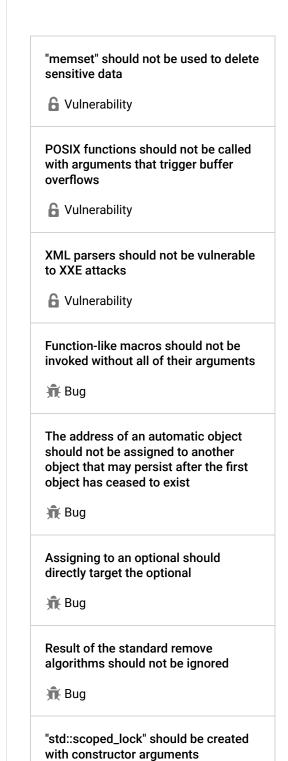
Vulnerability 13

Reg Bug 111

Security 18
Hotspot

Security 18

Tags



🖷 Bug

📆 Bug

📆 Bug

📆 Bug

📆 Bug

Objects should not be sliced

Immediately dangling references

"pthread_mutex_t" should be unlocked in the reverse order they were locked

"pthread_mutex_t" should be properly

"pthread_mutex_t" should not be consecutively locked or unlocked

initialized and destroyed

should not be created

Unions should not be used Analyze your code based-on-misra The use of unions to access an object in different ways may result in the data being misinterpreted. Therefore, this rule prohibits the use of unions for any purpose. **Noncompliant Code Example** union U1 { // Noncompliant float j; int i; }; See MISRA C:2004, 18.4 - Unions shall not be used. • MISRA C++:2008, 9-5-1 - Unions shall not be used. • MISRA C:2012, 19.2 - The union keyword should not be used Available In: sonarlint 😔 | sonarcloud 🖒 | sonarqube

Search by name...

© 2008-2022 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.

Privacy Policy

I
🖟 Bug
"std::move" and "std::forward" should not be confused
∰ Bug
A call to "wait()" on a "std::condition_variable" should have a condition
n Bug
A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic_cast
ਜ਼ਿ Bug
Functions with "noreturn" attribute should not return
👬 Bug
RAII objects should not be temporary
्रे Bug
"memcmp" should only be called with pointers to trivially copyable types with no padding
🙃 Bug
"memcpy", "memmove", and "memset" should only be called with pointers to trivially copyable types
🙃 Bug
"std::auto_ptr" should not be used
n Bug
Destructors should be "noexcept"
🖟 Bug