



CloudFormation

Java

JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

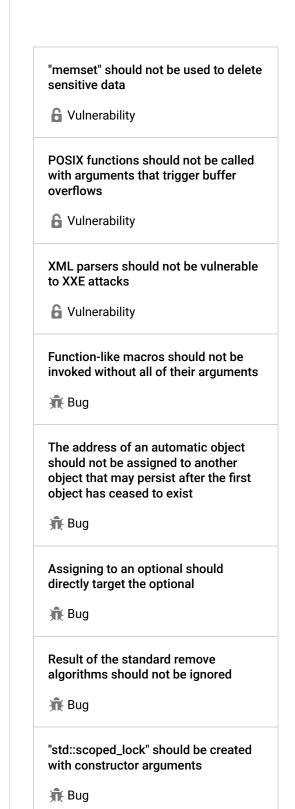


C++ static code analysis

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

o Security **⊗** Code (436) Quick 68 Fix All 578 6 Vulnerability (13) **R** Bug (111) rules Hotspot

Tags



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

📆 Bug

📆 Bug

📆 Bug

📆 Bug

Pointers should not be cast to Analyze your code integral types 🙀 Bug 🛮 春 Critical 🕝 based-on-misra cert The size of integer required to hold a memory address is implementation-dependent. Therefore, casting a pointer (i.e. a memory address) to any integral data type may result in data loss because the integral type is too small to hold the full address When treating a memory address as integer type is absolutely required, you should be sure to use a large enough type to hold all the data. **Noncompliant Code Example** int addr = (int) &p; See • MISRA C:2004, 11.3 - A cast should not be performed between a pointer type and an integral type. • MISRA C++:2008, 5-2-9 - A cast should not convert a pointer type to an integral • MISRA C:2012, 11.4 - A conversion should not be performed between a pointer to object and an integer type • CERT, INT36-C. - Converting a pointer to integer or integer to pointer Available In: sonarlint 😁 | sonarcloud 💩 | sonarqube | Developer

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