

Go HTML 5

Java

JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

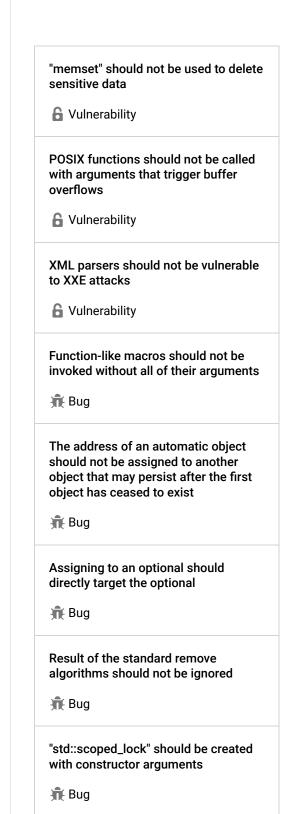
XML



C++ static code analysis

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

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

```
Functions should not be declared
                                                Analyze your code
at block scope
misra-c++2008 misra-c2004
A function declared at block scope will refer to a member of the enclosing
namespace, and so the declaration should be explicitly placed at the namespace
Additionally, where a declaration statement could either declare a function or an
object, the compiler will choose to declare the function. To avoid potential developer
confusion over the meaning of a declaration, functions should not be declared at
block scope.
Noncompliant Code Example
  class A {
  };
  void fun() {
    void nestedFun(); // Noncompliant; declares a function in
                  // Noncompliant; declares a function at block s
    A a();
See
  • MISRA C:2004, 8.6 - Functions shall be declared at file scope
 • MISRA C++:2008, 3-1-2 - Functions shall not be declared at block scope
 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