



ABAP

- Apex
- С
- C++
- CloudFormation
- COBOL
- C#
- **CSS**
- Flex
- Go =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**

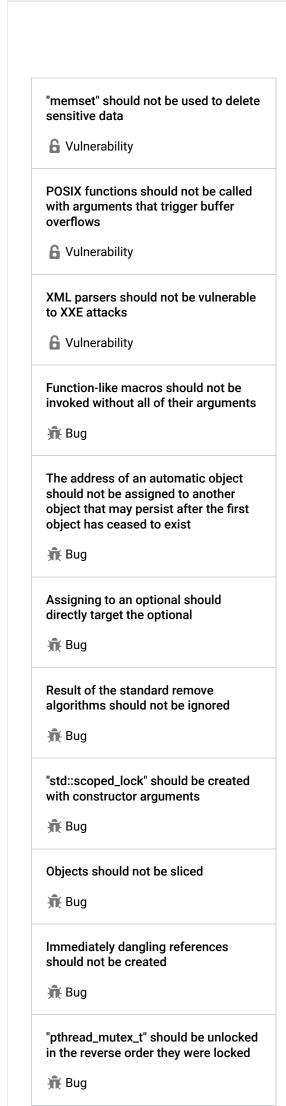


C++ static code analysis

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

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

Tags



"pthread_mutex_t" should be properly

"pthread_mutex_t" should not be consecutively locked or unlocked

initialized and destroyed

📆 Bug

#include directives in a file should only be preceded by other preprocessor directives or comments misra-c++2008 misra-c2004 confusing Code

Analyze your code

Major ? Smell misra-c2012

Search by name...

To aid code readability, all the #include directives in a particular code file should be grouped together near the top of the file. The only items which may precede an #include in a file are other preprocessor directives or comments.

Additionally, an #include may appear within an extern "C" block, this can be used for instance to include a C file from a C++ file.

Noncompliant Code Example

```
#include <h1.h> /* Compliant */
int32 t i;
#include <f2.h> /* Noncompliant */
```

Compliant Solution

```
#include <h1.h>
#include <f2.h>
extern "C" {
#include <f3.h>
int32_t i;
```

See

- MISRA C:2004, 19.1 #include statements in a file should only be preceded by other preprocessor directives or comments.
- MISRA C++:2008, 16-0-1 #include directives in a file shall only be preceded by other preprocessor directives or comments.
- MISRA C:2012, 20.1 #include directives should only be preceded by preprocessor directives or comments

Available In:

sonarlint ⊖ | sonarcloud 🖒 | sonarqube

© 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