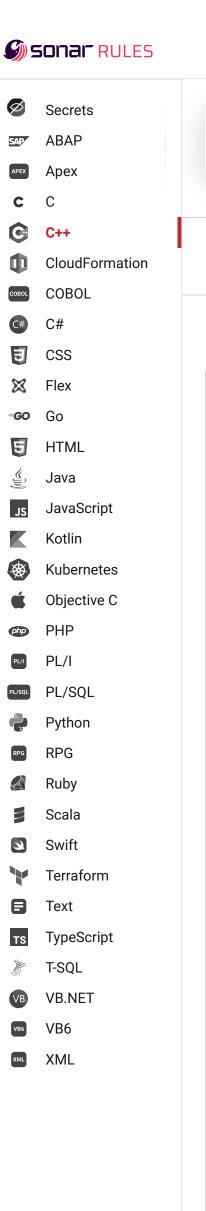
O Quick 68 Fix





ΑII

rules

578

## C++ static code analysis

6 Vulnerability (13)

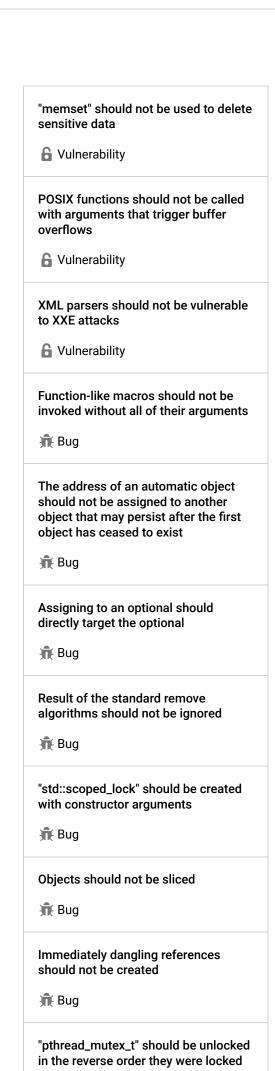
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C++ code

**R** Bug (111)

o Security

Tags

Hotspot



📆 Bug

📆 Bug

"pthread\_mutex\_t" should be properly

"pthread\_mutex\_t" should not be consecutively locked or unlocked

initialized and destroyed

An `if ( condition )` construct shall be followed by a compound statement. The else keyword shall Analyze your code be followed by either a compound statement, or another if statement ☼ Code Smell ♥ Minor ② misra-c++2008 This rule is a strict implementation of a MISRA (Motor Industry Software Reliability Association) rule. MISRA defines best practices for developing safety-critical software. You can learn more about this rule in the MISRA documents referenced below. If you are not concerned with MISRA compliance or with safety-critical software, you might consider using rule {rule:cpp:S121} instead; it targets the same kind of issue for general-purpose software. See • MISRA C++2008, 6-4-1 An if ( condition ) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement. Available In: sonarlint ⊖ | sonarcloud ☆ | sonarqube

⊗ Code (436)

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
<b>∰</b> 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
<b>ਜ਼ਿ</b> 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
<b>n</b> Bug
Destructors should be "noexcept"
🖟 Bug