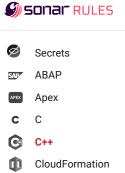
Search by name.





COBOL

C#

3 CSS

 $\mathbb{X}$ Flex

-GO Go

5 HTML 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

⊗ Code O Quick 68 Fix ΑII 578 Security 18 436 6 Vulnerability (13) **R** Bug (111) rules Hotspot Smell

Tags

"memset" should not be used to delete sensitive data 6 Vulnerability POSIX functions should not be called with arguments that trigger buffer overflows ♠ Vulnerability XML parsers should not be vulnerable to XXE attacks Vulnerability Function-like macros should not be invoked without all of their arguments 👬 Bug The address of an automatic object should not be assigned to another object that may persist after the first object has ceased to exist 👬 Bug Assigning to an optional should directly target the optional # Bug Result of the standard remove algorithms should not be ignored 👬 Bug "std::scoped\_lock" should be created with constructor arguments # Bug Objects should not be sliced # Bug Immediately dangling references should not be created # Bug

"pthread\_mutex\_t" should be unlocked

in the reverse order they were locked

"pthread\_mutex\_t" should be properly

# Bug

```
"switch" statements should not
                                                Analyze your code
contain non-case labels
based-on-misra suspicious
Even if it is legal, mixing case and non-case labels in the body of a switch statement
is very confusing and can even be the result of a typing error.
Noncompliant Code Example
The code is syntactically correct but the behavior is not the expected one
  switch (day) {
   case MONDAY:
    case TUESDAY:
    WEDNESDAY: // instead of "case WEDNESDAY"
      doSomething();
      break:
Compliant Solution
  switch (day) {
   case MONDAY:
    case TUESDAY:
    case WEDNESDAY:
      doSomething();
      break;
  }
See
  • MISRA C:2004, 15.0 - The MISRA C switch syntax shall be used.

    MISRA C++:2008 6-4-3 - A switch statement shall be a well-formed switch

   statement
 • MISRA C:2012, 16.1 - All switch statements shall be well-formed
 Available In:
 sonarlint ⊖ | sonarcloud ☆ | sonarqube | Develor Edition
```

© 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

initialized and destroyed

in Bug

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

in Bug

"std::move" and "std::forward" should not be confused

in Bug

A call to "wait()" on a "std::condition\_variable" should have a