



SAP ABAP

APEX Apex

C C

© C++

CloudFormation

COBOL COBOL

C# C#

**∃** CSS

X Flex

**ംഗ** ദറ

HTML

Js JavaScript

Kotlin

Kubernetes

**©** Objective C

PHP

PL/I

PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

■ Text

Ts TypeScript

T-SQL

VB.NET

VB6 VB6

xmL XML

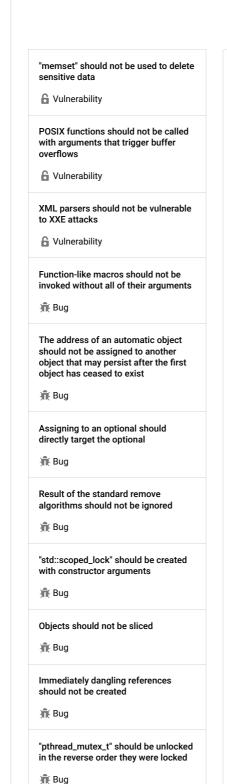


## C++ static code analysis

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

Tags

"case" ranges should cover multiple



"pthread\_mutex\_t" should be properly

```
Code Smell Blocker suspicious gnu

The GNU compiler extension that allows cases to be specified with ranges should only be used when a range is actually needed. Use it with the same number on both ends of the range, and you've either made a mistake because an actual range was intended, or you've used the syntax inappropriately in a way that is highly likely to confuse maintainers.

Noncompliant Code Example

switch (i) {
    case 0:
    //...
    break;
```

Search by name.

Analyze your code

## Compliant Solution

}

case 1 ... 2:

case 3 ... 3: // Noncompliant

//...

break;

//... break;

```
switch (i) {
  case 0:
    //...
    break;
  case 1 ... 2:
    //...
    break;
  case 3:
    //...
    break;
}
```

or

```
switch (i) {
  case 0:
    //...
    break;
  case 1 ... 2:
    //...
    break;
  case 3 ... 5:
    //...
    break;
}
```

initialized and destroyed

## Bug

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

## Bug

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

## Bug

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

Available In:

sonarlint ⊖ | sonarcloud ♂ | sonarqube Developer 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