

- Secrets
- ABAP
- Apex
- C
- C++**
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- 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

All rules **578**

Vulnerability **13**

Bug **111**

Security Hotspot **18**

Code Smell **436**

Quick Fix **68**

Tags

Search by name...



"memset" should not be used to delete sensitive data

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

Bug

"pthread_mutex_t" should be properly

Recursion should not be infinite

Analyze your code

Bug Blocker suspicious

Recursion happens when control enters a loop that has no exit. This can happen a method invokes itself, when a pair of methods invoke each other, or when goto statements are used to move between two segments of code. It can be a useful tool, but unless the method includes a provision to break out of the recursion and return, the recursion will continue until the stack overflows and the program crashes.

Noncompliant Code Example

```
int pow(int num, int exponent) { // Noncompliant; no condition
    num = num * pow(num, exponent-1);
    return num; // this is never reached
}

void internalRecursion(int i) {
    start:
        goto end;
    end:
        goto start; // Noncompliant; there's no way to break out
}
```

Compliant Solution

```
int pow(int num, int exponent) {
    if (exponent > 1) { // recursion now conditional and stop
        num = num * pow(num, exponent-1);
    }
    return num;
}
```

Available In:

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](#)

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