

- 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 **311**

Vulnerability **13**

Bug **74**

Security Hotspot **18**

Code Smell **206**

Quick Fix **14**

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

"pthread\_mutex\_t" should be unlocked in the reverse order they were locked

Bug

"pthread\_mutex\_t" should be properly initialized and destroyed

Bug

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

Bug

Functions with "noreturn" attribute should not return

Bug

"memcpy" should only be called with pointers to trivially copyable types with no padding

Bug

### Conditionally executed code should be reachable

Analyze your code

Bug

Major

cwe based-on-misra cert unused suspicious pitfall

Conditional expressions which are always true or false can lead to dead code. Such code is always buggy and should never be used in production.

#### Noncompliant Code Example

```
a = false;
if (a) { // Noncompliant
    doSomething(); // never executed
}

if (!a || b) { // Noncompliant; "!a" is always "true", "b" is
    doSomething();
} else {
    doSomethingElse(); // never executed
}
```

#### See

- MISRA C:2004, 13.7 - Boolean operations whose results are invariant shall not be permitted.
- MISRA C:2012, 14.3 - Controlling expressions shall not be invariant
- [MITRE, CWE-570](#) - Expression is Always False
- [MITRE, CWE-571](#) - Expression is Always True
- [CERT, MSC12-C](#) - Detect and remove code that has no effect or is never executed

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](#)

Stack allocated memory and non-owned memory should not be freed

 Bug

Closed resources should not be accessed

 Bug

Dynamically allocated memory should be released

 Bug

Freed memory should not be used