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

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

Bug

"switch" statements should not contain non-case labels

Analyze your code

Code Smell | Blocker | 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 Developer Edition

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