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

Functions without parameters should be declared with parameter type "void"

Analyze your code

Code Smell   Critical   based-on-misra cert pitfall

There is a real, functional difference between a function with an empty parameter list and one with an explicitly `void` parameter list: It is possible to pass parameters to a function with an empty list; the compiler won't complain. That is not the case for a function with a `void` list. Thus, it is possible, and even easy to invoke empty-list functions incorrectly without knowing it, and thereby introduce the kind of subtle bug that can be very difficult to track down.

Noncompliant Code Example

```
void myfunc ();  // Noncompliant

//...

void otherFunc() {
  int a = 4;
  //...
  myfunc(a); // Compiler allows this
}
```

Compliant Solution

```
void myfunc ( void );

//...

void otherFunc() {
  int a = 4;
  //...
  myfunc(a); // Compiler error!
}
```

See

- MISRA C:2004, 16.5 - Functions with no parameters shall be declared with parameter type void
- CERT, DCL20-C. - Explicitly specify void when a function accepts no arguments

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