

- 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

Object and function types should be explicitly stated in their declarations and definitions

Analyze your code

Code Smell Critical based-on-misra cert

The C90 standard allows implicit typing of variables and functions, and some C compilers still support legacy code by allowing implicit typing. But it should not be used for new code because it might lead to confusion.

### Noncompliant Code Example

```
extern x;
const x;
static fun(void);
typedef ( *pfi ) ( void );
```

### Compliant Solution

```
extern int16_t x;
const int16_t x;
static int16_t fun(void);
typedef int16_t ( *pfi ) ( void );
```

### See

- MISRA C:2004, 8.2 - Whenever an object or function is declared or defined, its type shall be explicitly stated
- MISRA C:2012, 8.1 - Types shall be explicitly specified
- [CERT, DCL31-C](#) - Declare identifiers before using them

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