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

Objective C static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your OBJECTIVE C code

All rules **315** | 🔒 Vulnerability **10** | 🐛 Bug **75** | Security Hotspot **18** | Code Smell **212** | Quick Fix **13**

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

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

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

Functions should not be declared at block scope

Analyze your code

⊘ Code Smell   ⌄ Major ⍰   🏷 misra-c++2008 misra-c2004

A function declared at block scope will refer to a member of the enclosing namespace, and so the declaration should be explicitly placed at the namespace level.

Additionally, where a declaration statement could either declare a function or an object, the compiler will choose to declare the function. To avoid potential developer confusion over the meaning of a declaration, functions should not be declared at block scope.

Noncompliant Code Example

```
class A {
};

void fun() {
  void nestedFun();  // Noncompliant; declares a function in

  A a();       // Noncompliant; declares a function at block s
}
```

See

- MISRA C:2004, 8.6 - Functions shall be declared at file scope
- MISRA C++:2008, 3-1-2 - Functions shall not be declared at block scope

Available In:

sonarcloud ⊙ | sonarqube))) Developer Edition

Freed memory should not be used

🐞 Bug

Memory locations should not be released more than once

🐞 Bug

Memory access should be explicitly bounded to prevent buffer overflows

🐞 Bug

Printf-style format strings should not lead to unexpected behavior at runtime

🐞 Bug

Recursion should not be infinite

🐞 Bug

Resources should be closed

🐞 Bug

Hard-coded credentials are security-sensitive

🛡 Security Hotspot

"goto" should jump to labels declared later in the same function

☢ Code Smell

Only standard forms of the "defined" directive should be used

☢ Code Smell

Switch labels should not be nested inside non-switch blocks

☢ Code Smell