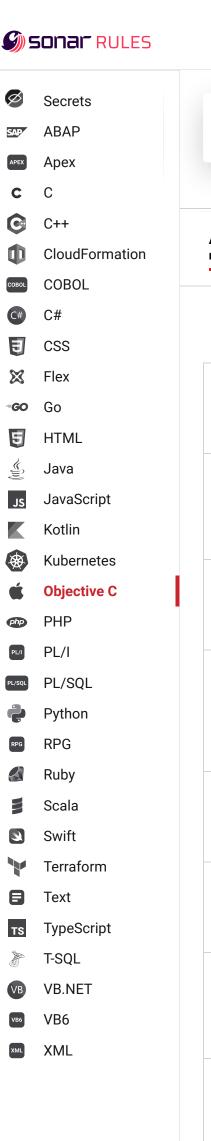
O Quick 13 Fix





should not return

with no padding

"memcmp" should only be called with pointers to trivially copyable types

Stack allocated memory and nonowned memory should not be freed

Closed resources should not be

Dynamically allocated memory should

📆 Bug

📆 Bug

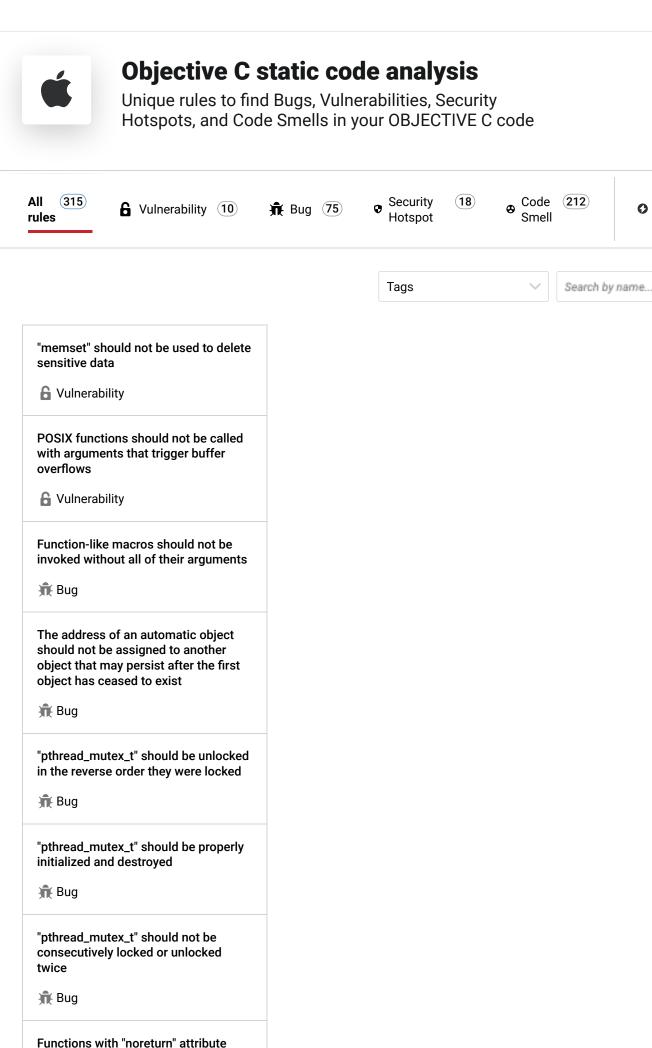
📆 Bug

accessed

📆 Bug

be released

📆 Bug



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 securitysensitive

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

"bool" expressions should not be used as operands to built-in operators other than =, &&, ||, !, ==, !=, unary &, and the conditional operator

**Analyze your code** 



based-on-misra suspicious

The use of bool operands with other operators is unlikely to be meaningful (or intended). Best case it will be confusing to maintainers, worst case it will not have the intended effect. Either way, it is highly recommended to stick to boolean operators when dealing with bool operands.

This rule allows the detection of such uses, which often occur because the logical operators (&&, | | and !) can be easily confused with the bitwise operators (&, | and

## **Noncompliant Code Example**

```
bool b1 = true;
bool b2 = false;
int8_t s8a;
if ( b1 \& b2 ) // Noncompliant
if ( ~b1 ) // Noncompliant
if ( b1 < b2 ) // Noncompliant
if ( b1 ^ b2 ) // Noncompliant
```

## **Compliant Solution**

```
if ( b1 && b2 )
if (!b1)
if ( b1 == false )
if (b1 == b2)
if ( b1 != b2 )
s8a = b1 ? 3 : 7;
```

## **Exceptions**

Operators |= and &= are ignored when used with bool operands. Operator ++ is also ignored with a bool operand because it is covered by rule {rule:cpp:S2668}.

```
void test(bool b1, bool b2, int i1) {
 b1 |= b2; // ignored
 b1++; // ignored here, handled by S2668
 b1 &= b2; // ignored
 b1 &= i1; // Noncompliant; right operand is not a bool
```

## See

• MISRA C++:2008, 4-5-1 - Expressions with type bool shall not be used as operands to built-in operators other than the assignment operator =, the logical operators &&, ||, !, the equality operators == and !=, the unary & operator, and the conditional operator.

Available In:



© 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