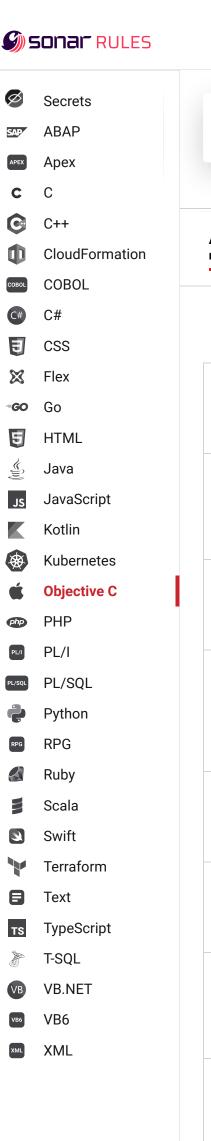
O Quick 13 Fix





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

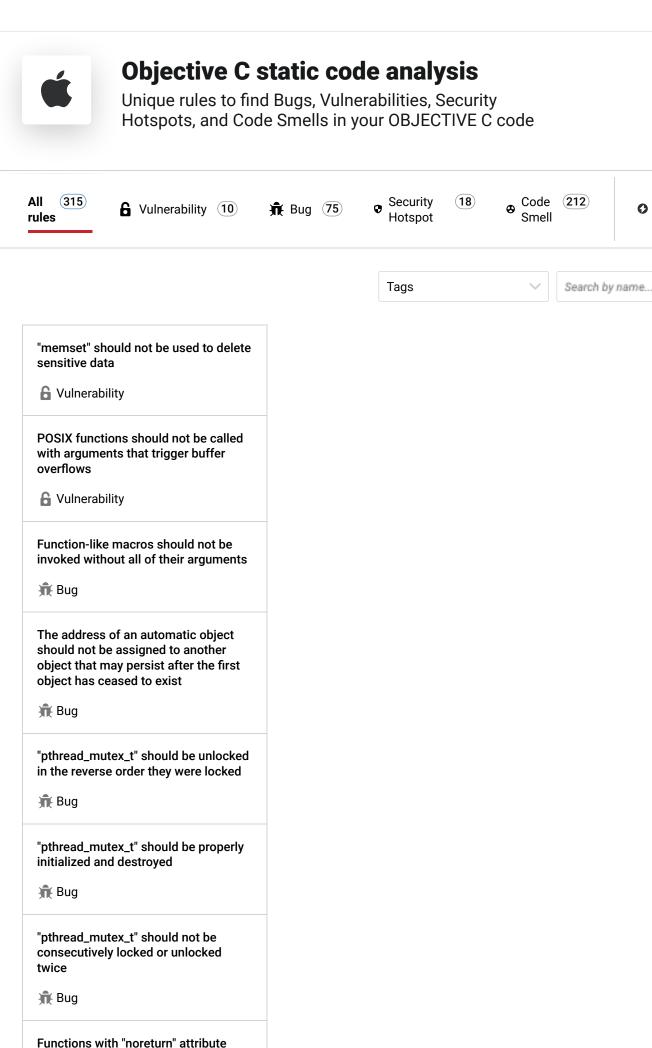
📆 Bug

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

Relational and subtraction operators should not be used with pointers to different arrays

Analyze your code

Rug Oritical 🕝



cppcoreguidelines based-on-misra

Attempting to make a comparison between pointers using >, >=, < or <= will produce undefined behavior if the two pointers point to different arrays.

Additionally, directly comparing two arrays for equality or inequality has been deprecated in C++.

However, equality or inequality between an array and a pointer is still valid

Noncompliant Code Example

```
void f1 ( )
 int a1[ 10 ];
  int a2[ 10 ];
 int * p1 = a1;
  if (p1 < a2) // Non-compliant, p1 and a2 point to differe
  if ( p1 - a2 > 0 ) // Non-compliant, p1 and a2 point to dif
  {
 if ( a1 == a2) // Non-compliant (in C++). Comparing differe
  {
 }
}
```

Compliant Solution

```
void f1 ( )
{
 int a1[ 10 ];
  int * p1 = a1;
  if ( p1 < a1 ) // Compliant, p1 and a1 point to the same ar
  if ( p1 - a1 > 0 ) // Compliant, p1 and a1 point to the sa
  {
 if ( p1 == a2 ) // Compliant, comparing a pointer and an ar
  {
 }
}
```

See

- MISRA C:2004, 17.3 >, >=, <, <= shall not be applied to pointer types except where they point to the same array.
- MISRA C++:2008, 5-0-18 >, >=, <, <= shall not be applied to objects of pointer type, except where they point to the same array.
- C++ Core Guidelines ES.62 Don't compare pointers into different arrays

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