

"pthread_mutex_t" shou consecutively locked or WB.NET

WB6

WML

"pthread_mutex_t" shou consecutively locked or Bug

Functions with "noretur not return

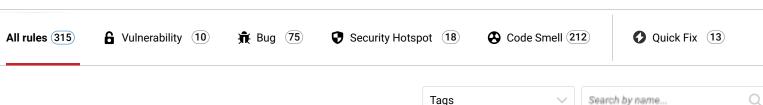
Bug

"memcmp" should only pointers to trivially consecutively locked or consecutively locked or consecutively locked or memcmp should only pointers to trivially consecutively locked or cons

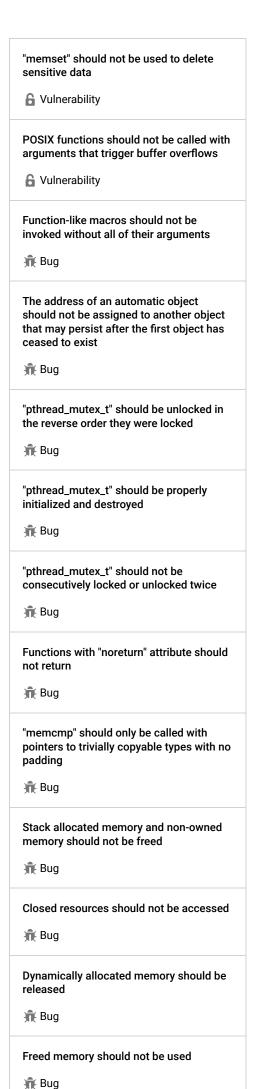


Objective C static code analysis

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



Code Smell Blocker



Memory locations should not be released

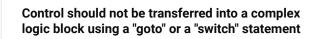
Memory access should be explicitly

bounded to prevent buffer overflows

more than once

📆 Bug

📆 Bug



Analyze your code

Having a switch and its cases wholly encompassed by a control structure such as a try, @try, catch, @catch, or a loop is perfectly acceptable. (try and catch are used hereafter to refer to both variants.) It is also acceptable to have a goto and its target label wholly encompassed in a control structure.

lock-in cert misra-c++2008 pitfall

What is not acceptable is using a goto or case to suddenly jump into the body of a try, catch, Objective-C @finally, or loop structure. Tangling labels or switch blocks with other control structures results in code that is difficult, if not impossible to understand. More importantly, when it compiles (some of these constructs won't compile under ISO-conformant compilers), it can lead to unexpected results. Therefore this usage should be strictly avoided.

This C++ code sample, which is also applicable to Objective-C if try and catch are converted to @try and @catch, demonstrates jumping into a switch and into a try and catch:

Noncompliant Code Example

```
void f ( int32_t i )
 if ( 10 == i )
   goto Label_10; // Noncompliant; goto transfers control into try block
 if ( 11 == i )
   goto Label_11; // Noncompliant; goto transfers control into catch block
 switch ( i )
   case 1:
     try
       Label 10:
       case 2: // Noncompliant; switch transfers control into try block
         // Action
         break;
     catch ( ... )
       Label_11:
       case 3: // Noncompliant; switch transfers control into catch block
         // Action
         break;
     break:
   default:
     // Default Action
     break;
}
```

Compliant Solution

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

📆 Bug

Recursion should not be infinite

Rug 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

The right-hand operands of && and || should not contain side effects

Code Small

```
default:
     // Default Action
     break;
 try
   if ( 2 == i || 10 == i)
     // Action
   }
 catch ( ... )
   if (3 == i || 11 == i)
     // Action
 }
}
```

See

- MISRA C++:2008, 15-0-3 Control shall not be transferred into a try or catch block using goto or switch statement
- CERT, MSC20-C. Do not use a switch statement to transfer control into a complex block

Available In:

sonarcloud 🖒 | sonarqube | Developer Edition

© 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