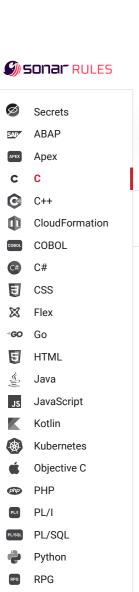
O Quick 14





Ruby

Scala

Text

**TypeScript** 

T-SQL

**VB.NET** 

VB6

XML



ΑII 311

rules

## C static code analysis

6 Vulnerability (13)

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

₩ Bug (74)

Security

Hotspot

Tags

18

"memset" should not be used to delete sensitive data 6 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 "pthread\_mutex\_t" should be properly initialized and destroyed # Bua "pthread\_mutex\_t" should not be consecutively locked or unlocked # Bug Functions with "noreturn" attribute should not return

₩ Bua

🖷 Bug

with no padding

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

There shall be at most one occurrence of the # or ## operators Analyze your code in a single macro definition preprocessor misra-c++2008 misra-c2004 Code Major 🕝 misra-c2012 Because the evaluation order of # and ## are not specified, the results of using them both in the same macro could be unpredictable. Therefore macros should contain at most once instance of either # or ##. Noncompliant Code Example #define NonCompliant(a, b) # a ## b int main() { std::cout << NonCompliant(Hello, World);</pre> The result of this code is unspecified. It will either print "HelloWorld" or trigger a compilation error. If ## is evaluated first this will print HelloWorld. If # is evaluated first this will cause a compilation error telling that "Hello"World is not a valid preprocessor token. Compliant Solution #define Stringfv(a) #a #define Compliant(a, b) Stringfy(a##b) int main(){ std::cout << Compliant(Hello, World);</pre> This example will always print "HelloWorld". See MISRA C:2004, 19.12 • MISRA C++ 2008, 16-3-1 • Related: MISRA C:2012, 20.11 Available In: sonarlint ⊖ | sonarcloud 🖒 | sonarqube 🤍

⊗ Code

Smell

206

Search by name.

© 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

Stack allocated memory and nonowned memory should not be freed

R
Bug

Closed resources should not be
accessed
Bug

Dynamically allocated memory should
be released
Bug

Freed memory should not be used