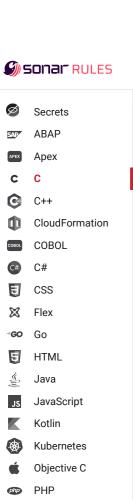
Search by name.



PL/I PL/I
PL/SQL PL/SQL

Python
RPG

Ruby

Swift

■ Terraform

Scala

**■** Text

Ts TypeScript

T-SQL

VB VB.NET

VB6 VB6

xmL XML



## C static code analysis

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

Tags

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

with no padding

🖷 Bug

"switch" statements should have at Analyze your code least 3 "case" clauses based-on-misra bad-practice switch statements are useful when there are many different cases depending on the value of the same expression. For just one or two cases however, the code will be more readable with if statements. Moreover, if statements are obviously more suitable when the condition of the switch is boolean. Here are the rules to count the cases: default is counted as a case. • If there is no default clause, the case count is incremented by one (to account for the else branch of an equivalent if). • All the cases falling through to default are not counted (they would all be the else branch of the equivalent if). Noncompliant Code Example switch (variable) { case 0: doSomething(); break; default: doSomethingElse(); break; } **Compliant Solution** if (variable == 0) { doSomething(); else { doSomethingElse(); • MISRA C:2012, 16.6 - Every switch statement shall have at least two switchclauses Available In: sonarlint ⊖ | sonarcloud ♠ | sonarqube | Develop 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

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