

RPG

Ruby Scala Swift Terraform

Text

T-SQL

VB.NET

VB6

XML

TypeScript



C static code analysis

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

All 311 & Vulnerability 13 第	E Bug 74 Security 18 Code 206 Quick Fix
	Tags V Search by name
"memset" should not be used to delete sensitive data Culperability	Size of bit fields should not exceed the size of their types Analyze your co
POSIX functions should not be called with arguments that trigger buffer overflows Uulnerability	Bit fields allow the developers to declare a class member with a specific size. However, the size of a bit field is also constrained by its type: even if the spe size is greater than the size of the type, the value of the bit field will not exce maximum value of this type. The extra bits will just create unused padding. The incompatibility of the size of the type with the specified size can have to causes: either the specified size is a typo error (that is the most probable cat the developer did not realize the size of the type he chose was too small.
XML parsers should not be vulnerable to XXE attacks Ulnerability	
Function-like macros should not be invoked without all of their arguments	Noncompliant Code Example
₩ Bug The address of an automatic object	<pre>class A { unsigned int b : 55; // Noncompliant, specified size };</pre>
should not be assigned to another object that may persist after the first object has ceased to exist	Compliant Solution
¶₹ Duy	class A {

"pthread_mutex_t" should be unlocked in the reverse order they were locked

"pthread_mutex_t" should be properly initialized and destroyed

👬 Bug

"pthread_mutex_t" should not be consecutively locked or unlocked

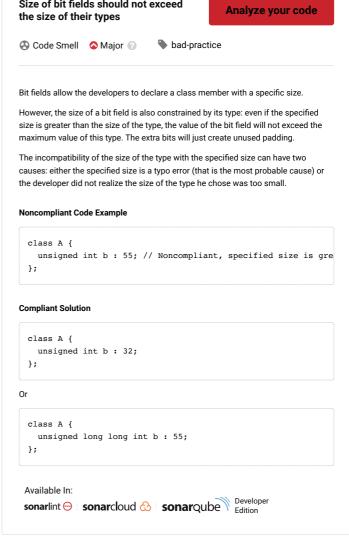
Bug

Functions with "noreturn" attribute should not return

👬 Bug

"memcmp" should only be called with pointers to trivially copyable types with no padding

👬 Bug



© 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