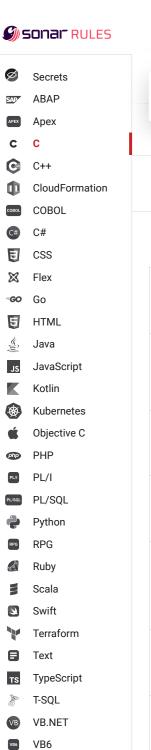
O Quick 14



XML



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

"m	emset" should not be used to delete
	nsitive data
6	Vulnerability
wit	SIX functions should not be called h arguments that trigger buffer erflows
6	Vulnerability
	IL parsers should not be vulnerable XXE attacks
6	Vulnerability
	nction-like macros should not be oked without all of their arguments
Ĥ	Bug
sho obj	e address of an automatic object ould not be assigned to another ject that may persist after the first ject has ceased to exist
Ĥ	Bug
	hread_mutex_t" should be unlocked the reverse order they were locked
Ĥ	Bug
	hread_mutex_t" should be properly ialized and destroyed
Ĥ	Bug
	hread_mutex_t" should not be nsecutively locked or unlocked ce
Ĥ	Bug

should not return

"memcmp" should only be called with

pointers to trivially copyable types with no padding

👬 Bug

🖷 Bug

Init-declarator-lists and memberdeclarator-lists should consist of Analyze your code single init-declarators and member-declarators respectively convention based-on-misra A Code Ouick Minor 0 Smell Fix Where multiple declarators appear in the same declaration the type of an identifier may not meet developer expectations. **Noncompliant Code Example** int il; int jl; // Compliant, but not preferred int i2, *j2; // Noncompliant int *i3. &j3 = i2; // Noncompliant **Compliant Solution** int il: int j1; int i2; int *j2; int *i3; int &j3 = i2; MISRA C++:2008. 8-0-1 - An init-declarator-list or a member-declarator-list shall consist of a single init-declarator or member-declarator respectively • CERT, DCL52-J. - Do not declare more than one variable per declaration • CERT, DCL04-C. - Do not declare more than one variable per declaration • C++ Core Guidelines - ES.10 - Declare one name (only) per declaration Available In:

⊗ 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