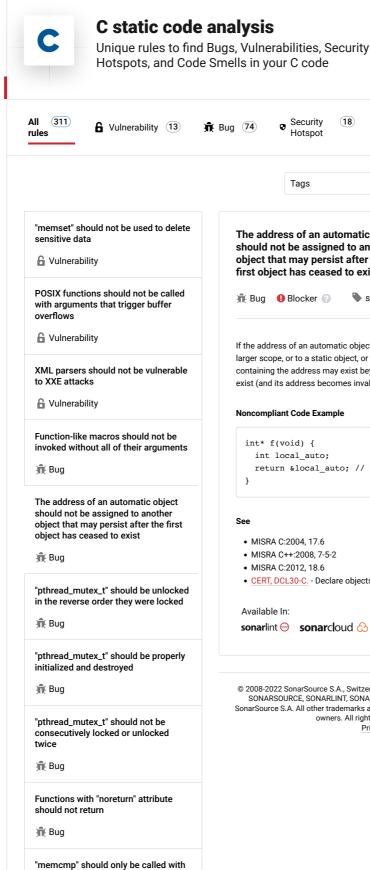
O Quick 14







pointers to trivially copyable types

with no padding

🖷 Bug

The address of an automatic object should not be assigned to another Analyze your code object that may persist after the first object has ceased to exist Rug Blocker 🕝 symbolic-execution based-on-misra cert If the address of an automatic object is assigned to another automatic object of larger scope, or to a static object, or returned from a function then the object containing the address may exist beyond the time when the original object ceases to exist (and its address becomes invalid). **Noncompliant Code Example** int* f(void) { int local_auto; return &local_auto; // Noncompliant, returning address of a See • MISRA C:2004, 17.6 MISRA C++:2008, 7-5-2 • MISRA C:2012, 18.6 • CERT, DCL30-C. - Declare objects with appropriate storage durations Available In: sonarlint ⊖ | sonarcloud ☆ | sonarqube Develor Developer

⊗ Code

Smell

206

Search by name.

Security

Hotspot

Tags

18

© 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