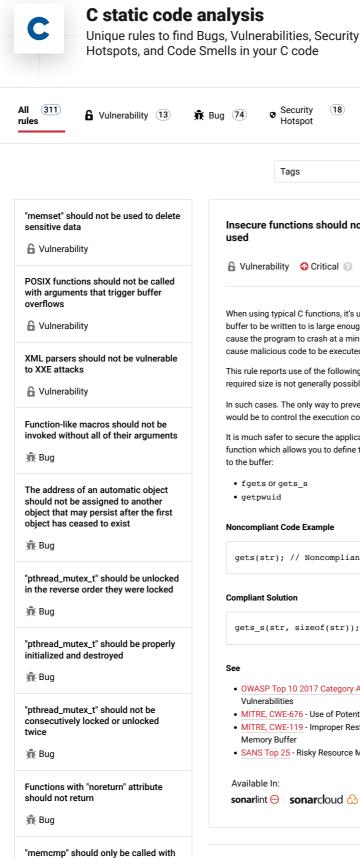
O Quick 14







pointers to trivially copyable types

with no padding

🖷 Bug

Tags Search by name. Insecure functions should not be Analyze your code used cwe sans-top25 owasp cert When using typical C functions, it's up to the developer to make sure the size of the buffer to be written to is large enough to avoid buffer overflows. Buffer overflows can  $% \left\{ 1,2,...,n\right\}$ cause the program to crash at a minimum. At worst, a carefully crafted overflow can cause malicious code to be executed. This rule reports use of the following insecure functions, for which knowing the required size is not generally possible: gets() and getpw(). In such cases. The only way to prevent buffer overflow while using these functions would be to control the execution context of the application. It is much safer to secure the application from within and to use an alternate, secure function which allows you to define the maximum number of characters to be written to the buffer: • fgets or gets s • getpwuid Noncompliant Code Example gets(str); // Noncompliant; `str` buffer size is not checked **Compliant Solution** gets\_s(str, sizeof(str)); // Prevent overflows by enforcing a OWASP Top 10 2017 Category A9 - Using Components with Known Vulnerabilities • MITRE, CWE-676 - Use of Potentially Dangerous Function • MITRE, CWE-119 - Improper Restriction of Operations within the Bounds of a Memory Buffer • SANS Top 25 - Risky Resource Management Available In: sonarlint sonarcloud sonaroube

⊗ Code

Smell

206

Security

Hotspot

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