Search by name.



Python **RPG**

Ruby Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

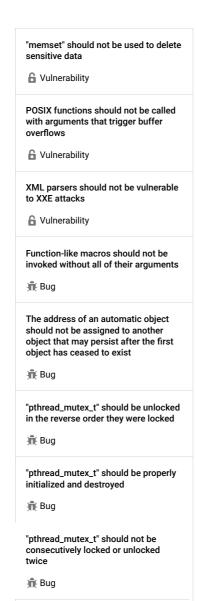


C static code analysis

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

⊗ Code O Quick 14 ΑII 311 Security 18 206 6 Vulnerability (13) **₩** Bug (74) rules Hotspot Smell

Tags



Functions with "noreturn" attribute

"memcmp" should only be called with

pointers to trivially copyable types

should not return

with no padding

Bug

🖷 Bug

```
"pthread_mutex_t" should not be
                                                  Analyze your code
consecutively locked or unlocked
twice
                            symbolic-execution multi-threading
# Bug Blocker
Mutexes are synchronization primitives that allow to manage concurrency.
  • non recursive mutexes are targeted by this rule. They can be locked/unlocked
   only once. Any locking/unlocking sequence that contains two consecutive
   identical operations leads to an undefined behaviour.
  • recursive mutexes are not target by this rule. They can be locked several times
   and unlocked several times as long as the number of locks/unlocks is the same.
This rule raises an issue when a pthread_mutex_t is locked or unlocked several
times in a row. We assume that all pthread mutex_t are non-recursive (this is the
most common case).
Noncompliant Code Example
  pthread_mutex_t mtx1;
  void bad1(void)
    pthread_mutex_lock(&mtx1);
    pthread_mutex_lock(&mtx1);
  void bad2(void)
    pthread mutex unlock(&mtx1);
```

Compliant Solution

pthread mutex unlock(&mtx1);

```
pthread_mutex_t mtx1;
void ok(void)
 pthread_mutex_lock(&mtx1);
 pthread_mutex_unlock(&mtx1);
```

See

• The Open Group pthread_mutex_init, pthread_mutex_destroy

Available In:

sonarlint o | sonarcloud o | sonarqube | Develo



Stack allocated memory and nonowned memory should not be freed

Bug

Closed resources should not be
accessed
Bug

Dynamically allocated memory should
be released
Bug

Freed memory should not be used

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