- Secrets
- ABAP
- Apex
- **C**
- C++
- CloudFormation
- COBOL
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Kubernetes
- Objective C
- PHP
- PL/I
- PL/SQL
- 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

All rules `311` | 🔒 Vulnerability `13` | 🐞 Bug `74` | 🛡 Security Hotspot `18` | ⊗ Code Smell `206` | ⚡ Quick Fix `14`

Tags ⌄   Search by name... 🔍

"memset" should not be used to delete sensitive data

🔒 Vulnerability

POSIX functions should not be called with arguments that trigger buffer overflows

🔒 Vulnerability

XML parsers should not be vulnerable to XXE attacks

🔒 Vulnerability

Function-like macros should not be invoked without all of their arguments

🐞 Bug

The address of an automatic object should not be assigned to another object that may persist after the first object has ceased to exist

🐞 Bug

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

🐞 Bug

"pthread_mutex_t" should be properly initialized and destroyed

🐞 Bug

"pthread_mutex_t" should not be consecutively locked or unlocked twice

🐞 Bug

Functions with "noreturn" attribute should not return

🐞 Bug

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

🐞 Bug

Line continuation characters '\' should not be followed by trailing whitespace

Analyze your code

🐞 Bug   🟢 Minor   ❓

The standard mentions that the line continuation character (\) should be immediately followed by a newline or be the very last character of the file in order for the lines to be joined.

Several compilers relax this requirement by allowing whitespace after the \ character, but this is not portable because other compilers may not do the same.

Compliant Solution

```
// There should be no whitespace after the '\'
#define FOO BAR \
        BAZ
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

Stack allocated memory and non-owned 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