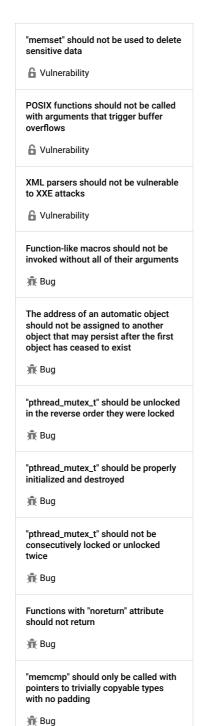




C static code analysis

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

| All (311) rules | 6 Vulnerability 13 | ∰ Bug 74) | Security (18) Hotspot | Code Smell 206 | Quick 14 | |
|-----------------|--------------------|--------------------------|-----------------------|--------------------|----------|--|
| | | Tags | | ✓ Search by name C | | |
| | | | | | | |



```
"typedef" names should be unique
                                                Analyze your code
identifiers
based-on-misra suspicious
Reusing a typedef name either as another typedef name or for any other purpose
may lead to developer confusion.
The same typedef shall not be duplicated anywhere in the project, even if the
declarations are identical.
Note that where the type definition is made in a header file, and that header file is
included in multiple source files, this rule is not violated.
Noncompliant Code Example
    typedef unsigned char uint8_t;
    typedef unsigned char uint8_t; // Noncompliant, redefinitio
    unsigned char uint8 t; // Noncompliant, reuse of uint8 t fo
Compliant Solution
  typedef unsigned char uint8_t;
  {
  }
  {
  }
    unsigned char myChar;
  • MISRA C:2004, 5.3 - A typedef name shall be a unique identifier.
 • MISRA C++:2008, 2-10-3 - A typedef name (including qualification, if any) shall be
   a unique identifier
 Available In:
 sonarlint ⊕ | sonarcloud ♦ | sonarqube
```

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 non- |
|-----------------|---------------------|
| owned memory | should not be freed |

🕕 Bug

Closed resources should not be accessed

<table-of-contents> Bug

Dynamically allocated memory should be released

👬 Bug

Freed memory should not be used