

PL/I PL/SQL PL/SQL

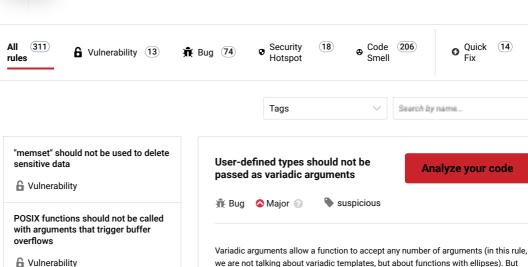
PHP

- **Python**
- RPG Ruby
- Scala
- Swift
- **Terraform**
- Text
- TS TypeScript
- T-SQL
- VB VB.NET
- VB6 VB6
- xmL XML



C static code analysis

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



parsers should not be vulnerable

This rules reports an issue if the type of the argument:

KE attacks

 is a non trivially copyable, movable or deletable type: there is no guarantee that it will work.

these arguments have to respect some criteria to be handled properly.

- is a class type that is trivially copyable, movable and deletable: in this case, we consider that the user intention was probably not to directly pass it as an argument but to call a method on it (c_str() for example)
- Noncompliant Code Example

```
class A {
   char* toStr();
};
void v(...);

void f() {
   A a;
   v(a); // Noncompliant

std::string myString = "foo";
   v(myString); // Noncompliant; string is not a POD type
}
```

Compliant Solution

```
class A {
   char* toStr();
}
void v(...);

void f() {
   A a;
   v(a.toStr()); // Compliant

std::string myString = "foo";
   v(myString.c_str()); // Compliant
}
```

Available In:

sonarlint ⊖ | sonarcloud & | sonarqube

sonarqube Developer

♠ 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 "pthread_mutex_t" should be properly initialized and destroyed # Bua "pthread_mutex_t" should not be consecutively locked or unlocked # Bug Functions with "noreturn" attribute should not return ₩ Bua "memcmp" should only be called with pointers to trivially copyable types with no padding 🖷 Bug

© 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 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