



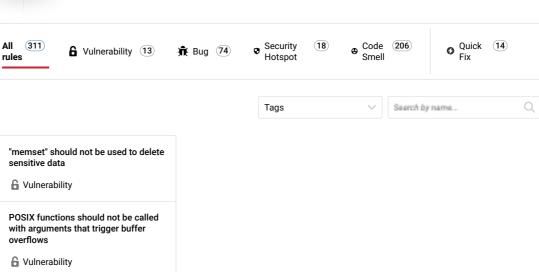
👬 Bug

👬 Bug

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



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



Stack allocated memory and nonowned memory should not be freed

Bug

Closed resources should not be accessed

₩ Bua

Dynamically allocated memory should be released

Bug

Freed memory should not be used

Braces should be used to indicate and match the structure in the nonzero initialization of arrays and structures

Analyze your code

based-on-misra

ISO/IEC 14882:2003 [1] requires initializer lists for arrays, structures and union types to be enclosed in a single pair of braces (though the behaviour if this is not done is undefined). The rule given here goes further in requiring the use of additional braces to indicate nested structures.

This forces the developer to explicitly consider and demonstrate the order in which elements of complex data types are initialized (e.g. multi-dimensional arrays).

The zero initialization of arrays or structures shall only be applied at the top level.

The non-zero initialization of arrays or structures requires an explicit initializer for each element

A similar principle applies to structures, and nested combinations of structures, arrays and other types.

Note also that all the elements of arrays or structures can be initialized (to zero or NULL) by giving an explicit initializer for the first element only. If this method of initialization is chosen then the first element should be initialized to zero (or NULL). and nested braces need not be used.

Noncompliant Code Example

```
int a1[3][2] = { 1, 2, 3, 4, 5, 6 }; // Noncompliant
int a2[5] = { 1, 2, 3 }; // Noncompliant, partial initializat
int a3[2][2] = { { }, { 1, 2 } }; // Noncompliant, zero initi
```

Compliant Solution

```
int al[3][2] = { { 1, 2 }, { 3, 4 }, { 5, 6 } }; // Compliant
int a2[5] = { 1, 2, 3, 0, 0 }; // Compliant, Non-zero initial
int a2[5] = { 0 }; // Compliant, zero initialization
int a3[2][2] = { }; // Compliant, zero initialization
```

- MISRA C:2004, 9.2 Braces shall be used to indicate and match the structure in the non-zero initialization of arrays and structures.
- MISRA C++:2008, 8-5-2 Braces shall be used to indicate and match the structure in the nonzero initialization of arrays and structures.
- MISRA C:2012, 9.2 The initializer of an aggregate or union shall be enclosed in
- MISRA C:2012, 9.3 Arrays shall not be partially initialized.

Available In:

sonarlint ⊕ | sonarcloud 🖒 | sonarqube 🤊

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOLIRCE SONARI INT SONAROLIBE and SONARCLOLID 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