O Quick 14



 $\bowtie$ Flex

-GO

Go 5 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



ΑII 311

rules

## C static code analysis

6 Vulnerability (13)

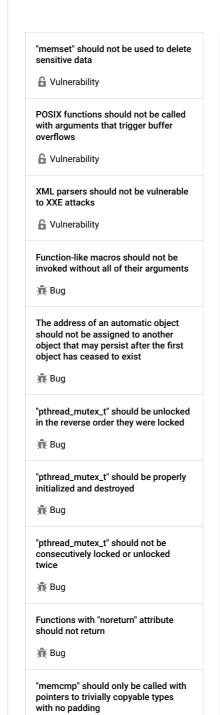
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C code

⊗ Code (206)

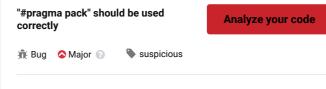
₩ Bug (74)

Tags Search by name.

Smell



🖷 Bug



#pragma pack is a non standard extension used to specify the packing alignment for structure, union and class members.

It is useful to

· remove padding and decrease the size of objects

Security

Hotspot

18

· align members to better fit optimal cpu alignment

However, the pragma pack directives need to be correctly defined to work properly.

This rule raises an issue if:

- the specified packing value is incorrect: it can only be 1, 2, 4, 8, or 16
- a parameter is ill-formed
- $\bullet$  the pop variant of this  $\mbox{\tt\#pragma}$  is called with both arguments identifier and value: such a call is undefined behavior
- a #pragma pack(push...) is performed but there is not corresponding use of #pragma pack(pop...)
- a #pragma pack(pop...) is performed but there is not corresponding use of #pragma pack(push...)
- a #pragma pack is in effect across several files: this becomes too complex, and could easily lead to undefined behavior, the same structure having different layout when seen from different translation units

## Noncompliant Code Example

```
#pragma pack(5) // Noncompliant, value is invalid
#pragma pack(2+2) // Noncompliant, value should be a literal
#include "myFile.h" // Noncompliant, the specified alignement
struct T {
   int i;
   short j;
   double k;
#pragma pack(push, r1, 16)
#pragma pack(pop, r1, 4) // Noncompliant, call to pop with t
#pragma pack(push, r2, 16)
#pragma pack(pop, r3) // Noncompliant, call to pop with no m
#pragma pack(push, 8) // Noncompliant, unmatched push
```

## **Compliant Solution**

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

```
#include "myFile.h"
#pragma pack(4)
struct T {
  int i;
  short j;
  double k;
#pragma pack(push, r1, 16)
#pragma pack(pop, r1)
#pragma pack(push, r2, 16)
#pragma pack(pop, r2)
#pragma pack(push, 8)
#pragma pack(pop)
```

Available In:

Developer
SonarSource, SonarLint, SonarQue and SonarCource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy