



ABAP

Apex

С

CloudFormation

COBOL

C++

C#

CSS

Flex

=GO

HTML 5

Go

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



Objective C static code analysis

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

ΑII 315 rules

6 Vulnerability 10

R Bug (75)

o Security Hotspot

⊗ Code (212)

O Quick 13
Fix

Analyze your code

Tags

"case" ranges should cover

multiple values

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

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

Stack allocated memory and nonowned memory should not be freed

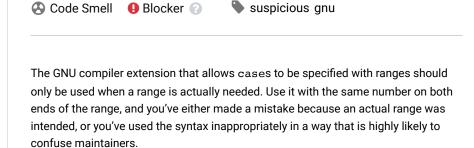
📆 Bug

Closed resources should not be accessed

📆 Bug

Dynamically allocated memory should be released

📆 Bug



Noncompliant Code Example

```
switch (i) {
  case 0:
    //...
    break;
  case 1 ... 2:
    //...
    break;
  case 3 ... 3: // Noncompliant
    //...
    break;
}
```

Compliant Solution

```
switch (i) {
  case 0:
    //...
    break;
  case 1 ... 2:
    //...
    break;
  case 3:
    //...
    break;
```

```
switch (i) {
  case 0:
    //...
    break;
  case 1 ... 2:
    //...
    break;
  case 3 ... 5:
    //...
    break;
```

Available In:

sonarcloud 🚳 sonarqube Developer

Freed memory should not be used
Memory locations should not be released more than once
Memory access should be explicitly bounded to prevent buffer overflows Bug
Printf-style format strings should not lead to unexpected behavior at runtime Bug
Recursion should not be infinite
Resources should be closed
Hard-coded credentials are security-sensitive
Hard-coded credentials are security- sensitive Security Hotspot "goto" should jump to labels declared later in the same function

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.

<u>Privacy Policy</u>