

Secrets

ABAP

- Apex
- С
- C++
- CloudFormation
- COBOL
- C#
- **CSS**
- Flex
- Go =GO
- HTML 5
- 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

All 315 6 Vulnerability 10 rules

R Bug (75)

o Security Hotspot

⊗ Code (212)

O Quick 13
Fix

Tags

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

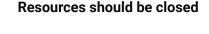
📆 Bug

Closed resources should not be accessed

📆 Bug

Dynamically allocated memory should be released

📆 Bug



Analyze your code

Blocker ?

cwe symbolic-execution leak denial-of-service cert

A call to the fopen()/open() function must be matched with a call to fclose()/close. Otherwise, you run the risk of using up all the OS's file handles, which could lock up not just your program but potentially everything on the box.

Noncompliant Code Example

```
int fun() {
  FILE *f = fopen("file", "r");
  if (f == NULL) {
    return -1;
  // ...
  return 0; // Noncompliant, file f has not been closed
```

Compliant Solution

```
int fun() {
 FILE *f = fopen("file", "r");
  if (f == NULL) {
    return -1;
  // ...
  fclose(f);
  return 0;
```

See

- MITRE, CWE-459 Incomplete Cleanup
- MITRE, CWE-772 Missing Release of Resource after Effective Lifetime
- CERT, FI004-J. Release resources when they are no longer needed
- CERT, FIO42-C. Close files when they are no longer needed
- Try With Resources

Available In:

sonarcloud 🚳 | sonarqube | Developer Edition

© 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**

| Freed memory should not be used Bug Memory locations should not be released more than once Bug 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 Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
|--|
| Memory locations should not be released more than once Bug 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 Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| 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 Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| bounded to prevent buffer overflows Bug Printf-style format strings should not lead to unexpected behavior at runtime Bug Recursion should not be infinite Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| Printf-style format strings should not lead to unexpected behavior at runtime Bug Recursion should not be infinite Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| lead to unexpected behavior at runtime Bug Recursion should not be infinite Bug Resources should be closed Bug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| Recursion should not be infinite Resources should be closed Resources should be closed Resources should be closed Sug Hard-coded credentials are security-sensitive Security Hotspot "goto" should jump to labels declared |
| Resources should be closed Resources should be closed Bug Hard-coded credentials are security- sensitive Security Hotspot "goto" should jump to labels declared |
| Resources should be closed Bug Hard-coded credentials are security- sensitive Security Hotspot "goto" should jump to labels declared |
| Hard-coded credentials are security- sensitive Security Hotspot "goto" should jump to labels declared |
| Hard-coded credentials are security- sensitive Security Hotspot "goto" should jump to labels declared |
| sensitive Security Hotspot "goto" should jump to labels declared |
| "goto" should jump to labels declared |
| |
| later in the same function |
| ⇔ Code Smell |
| Only standard forms of the "defined" directive should be used |
| ☼ Code Smell |
| Switch labels should not be nested inside non-switch blocks |
| ⇔ Code Smell |