Objective C static code analysis

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

All rules 315 | Vulnerability 10 | Bug 75 | Security Hotspot 18 | Code Smell 212 | Quick Fix 13

Tags

Search by name...

"goto" statements should not be used to jump into blocks

Analyze your code

Code Smell Blocker based-on-misra brain-overload pitfall

Use of `goto` can lead to programs that are extremely difficult to comprehend and analyse, and possibly to unspecified behavior.

Unfortunately, removing `goto` from some code can lead to a rewritten version that is even more difficult to understand than the original. Therefore, limited use of `goto` is sometimes advised.

However, the use of `goto` to jump into or out of a sub-block of code, such as into the body of a `for` loop is never acceptable, because it is extremely difficult to understand and will likely yield results other than what is intended.

Noncompliant Code Example

```
void f1 (int a) {
  if (a <=0) {
    goto L2;  // Noncompliant; jumps into a different block
  }

  if (a == 0) {
  {
    goto L1; // Compliant
  }
  goto L2;  // Noncompliant; jumps into a block

L1:
  for (int i = 0; i < a; i++) {
  L2:
    //...  Should only have come here with a >=0. Loop is inf
  }
}
```

Compliant Solution

```
void f1 (int a) {
  if (a <=0) {
    // ...
  }

  if (a == 0) {
  {
    goto L1; // Compliant
  }

L1:
  for (int i = 0; i < a; i++) {
  L2:
    //...
  }
}
```

See

- MISRA C++:2008, 6-6-1 - Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement
- MISRA C:2012, 15.3 - Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement

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 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

Available In:

sonarcloud sonarqube Developer Edition