Swift static code analysis

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

All rules 119 | Vulnerability 3 | Bug 14 | Security Hotspot 3 | Code Smell 99

Tags | Search by name...

Related "if/else if" statements and "cases" in a "switch" should not have the same condition

Analyze your code

Bug | Major | unused pitfall

A `switch` and a chain of `if/else if` statements is evaluated from top to bottom. At most, only one branch will be executed: the first one with a condition that evaluates to `true`.

Therefore, duplicating a condition automatically leads to dead code. Usually, this is due to a copy/paste error. At best, it's simply dead code and at worst, it's a bug that is likely to induce further bugs as the code is maintained, and obviously it could lead to unexpected behavior.

Noncompliant Code Example

```
if param == 1 {
  openWindow()
} else if param == 2 {
  closeWindow()
} else if param == 1 {            // Noncompliant
  moveWindowToTheBackground()
}

switch i {
  case 1:
    //...
  case 3:
    //...
  case 1:                         // Noncompliant
    //...
  default:
    // ...
}
```

Compliant Solution

```
if param == 1 {
  openWindow()
} else if param == 2 {
  closeWindow()
} else if param == 3 {
  moveWindowToTheBackground()
}

switch i {
  case 1:
    //...
  case 3:
    //...
  default:
    // ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

Bug

Return values from functions without side effects should not be ignored

Bug

Related "if/else if" statements and "cases" in a "switch" should not have the same condition

Bug

Identical expressions should not be used on both sides of a binary operator

Bug

All code should be reachable

Bug

Loops with at most one iteration should be refactored

Bug

"IBInspectable" should be used correctly

Code Smell

Functions should not have identical implementations

Code Smell

Ternary operators should not be nested

Code Smell

Closure expressions should not be nested too deeply

Code Smell

Backticks should not be used around