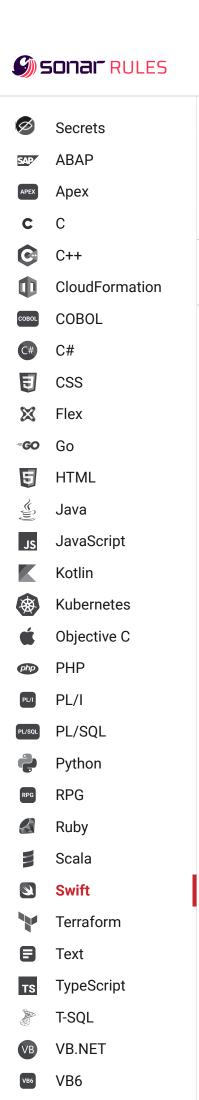
Code Smell 99



XML



Swift static code analysis

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

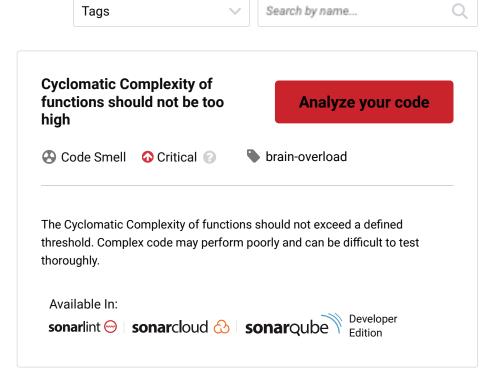
♣ Bug 14

All rules (119)
Hard-coded credentials are security- sensitive
Security Hotspot
Methods and field names should not be the same or differ only by capitalization
☼ Code Smell
Cipher algorithms should be robust
Using weak hashing algorithms is security-sensitive
Security Hotspot
Cognitive Complexity of functions should not be too high
Code Smell
"try!" should not be used
String literals should not be duplicated
☼ Code Smell
Functions and closures should not be empty
Code Smell
Collection elements should not be replaced unconditionally
n Bug
Collection sizes comparisons should make sense
👚 Bug
All branches in a conditional structure should not have exactly the same implementation
👚 Bug
Infix operators that end with "=" should update their left operands

📆 Bug

changed

Precedence and associativity of standard operators should not be



Security Hotspot 3

© 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

∰ 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
Rug
Identical expressions should not be used on both sides of a binary operator
🖟 Bug
All code should be reachable
Rug
Loops with at most one iteration should be refactored
🖟 Bug
"IBInspectable" should be used correctly
Functions should not have identical implementations
Ternary operators should not be nested
Closure expressions should not be nested too deeply
Code Smell
Backticks should not be used around