Code Smell 99



PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Text

T-SQL

VB.NET

VB6

XML

Terraform

TypeScript



Swift static code analysis

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

R Bug (14)

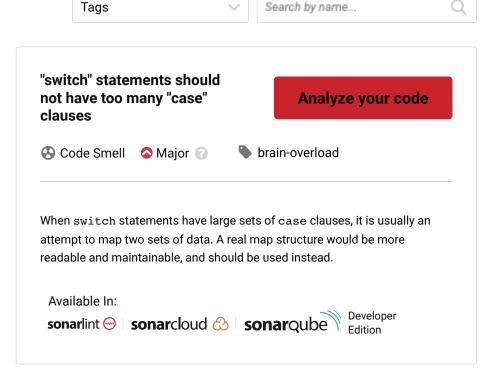
All rules (119)	6 Vulnerability 3
Hard-coded cr	redentials are security-
Security H	otspot
	field names should not or differ only by
Code Sme	II
Cipher algorit	hms should be robust
6 Vulnerabili	ty
Using weak has security-sensi	ashing algorithms is itive
Security H	otspot
Cognitive Con should not be	nplexity of functions too high
Code Sme	II
"try!" should n	ot be used
Code Sme	II
String literals	should not be duplicated
Code Sme	II
Functions and empty	I closures should not be
Code Sme	II
Collection eler	ments should not be anditionally
🛊 Bug	
Collection size make sense	es comparisons should
🛊 Bug	
	n a conditional structure ve exactly the same on
🙀 Bug	
	s that end with "=" should

update their left operands

Precedence and associativity of standard operators should not be

Rug Bug

changed



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