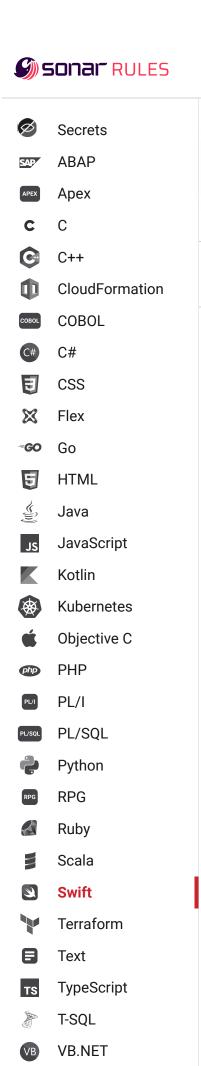
Code Smell 99



VB6

XML



Swift static code analysis

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

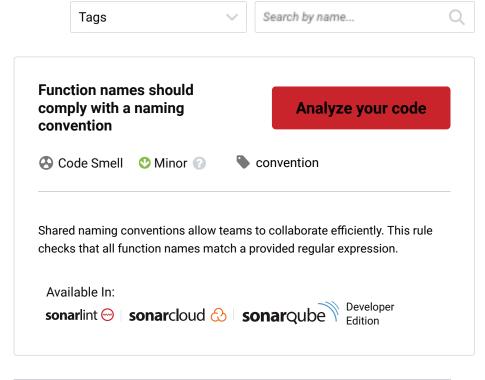
♣ Bug 14

All rules (119)	6 Vulnerability (3)
Hard-coded cree	dentials are security-
sensitive	dentials are security
Security Hot	spot
Methods and fie be the same or c capitalization	eld names should not differ only by
Code Smell	
Cipher algorithn	ns should be robust
G Vulnerability	
Using weak has security-sensitiv	hing algorithms is ve
Security Hote	spot
Cognitive Comp	lexity of functions to high
Code Smell	
"try!" should not	be used
Code Smell	
String literals sh	nould not be duplicated
Code Smell	
Functions and c	losures should not be
Code Smell	
Collection eleme	ents should not be ditionally
👚 Bug	
Collection sizes make sense	comparisons should
🖟 Bug	
	a conditional structure exactly the same
Rug	
Infix operators t update their left	hat end with "=" should operands

Rug 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