Code Smell 99



Php PHP PL/I

PL/SQL
Python

RPG

Ruby

Scala

Swift

Terraform

Text

Ts TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



Swift static code analysis

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

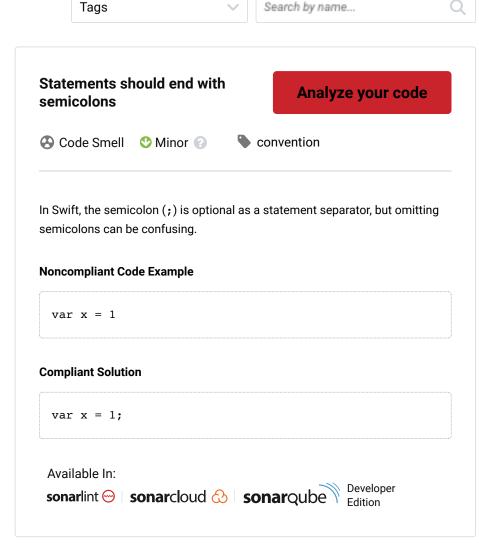
₩ Bug 14

Hard-coded c	redentials are security-
Security H	lotspot
	field names should not or differ only by
Code Sme	
Cipher algorit	thms should be robust
G Vulnerabil	ity
Using weak h security-sens	ashing algorithms is itive
Security H	lotspot
Cognitive Cor should not be	mplexity of functions too high
Code Sme	الغ
"try!" should r	not be used
Code Sme	ell .
String literals	should not be duplicated
Code Sme	 -
Functions and empty	d closures should not be
Code Sme	7
Collection ele	ements should not be onditionally
📆 Bug	
Collection siz make sense	es comparisons should
<table-of-contents> Bug</table-of-contents>	
	in a conditional structure ave exactly the same on
🕕 Bug	
Infiv operator	s that end with "=" should

📆 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