Code Smell 99

Search by name...





ABAP

Apex

С C

C++

CloudFormation

COBOL

C#

CSS

 \bowtie Flex

Go

5 HTML

Java

JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML



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	
	ard-coded credentials are security- ensitive
	Security Hotspot
b	lethods and field names should not e the same or differ only by apitalization
Q	Code Smell
С	ipher algorithms should be robust
6	Vulnerability
	sing weak hashing algorithms is ecurity-sensitive
	Security Hotspot
	ognitive Complexity of functions nould not be too high
6	Code Smell
"t	ry!" should not be used
9	Code Smell
S	tring literals should not be duplicated
Q	Code Smell
	unctions and closures should not be mpty
Q	Code Smell
_	ollection elements should not be
Ì	¥ Bug
	ollection sizes comparisons should hake sense
Ą	B Bug
sl	Il branches in a conditional structure nould not have exactly the same nplementation
Ť	¥ Bug

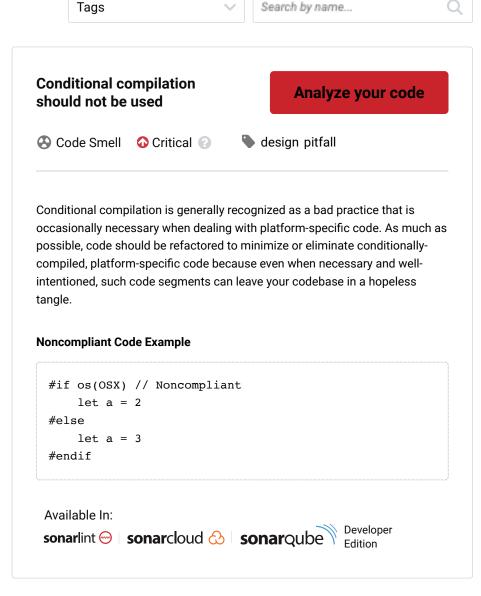
Infix operators that end with "=" should

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