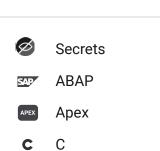
Code Smell 99





C++

CloudFormation

COBOL COBOL

C# C#

e css

💢 Flex

-co Go

THIML

Java

Js JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I PL/I

PL/SQL PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

Text

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

R Bug (14)

ll r	ules (119) 6 Vulnerability (3
	rd-coded credentials are security- nsitive
0	Security Hotspot
be	ethods and field names should not the same or differ only by pitalization
(3)	Code Smell
Cip	oher algorithms should be robust
6	Vulnerability
	ing weak hashing algorithms is curity-sensitive
0	Security Hotspot
	gnitive Complexity of functions ould not be too high
(3)	Code Smell
"try	y!" should not be used
(3)	Code Smell
Str	ing literals should not be duplicated
(3)	Code Smell
	nctions and closures should not be apty
8	Code Smell
	llection elements should not be placed unconditionally
Ĥ	Bug
	llection sizes comparisons should ake sense
ĴŖ	Bug
sh	branches in a conditional structure ould not have exactly the same plementation
	Bug

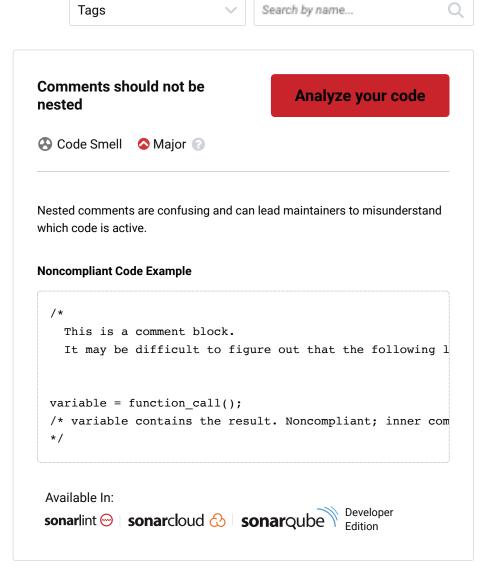
Infix operators that end with "=" should

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