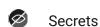
Code Smell 99

Search by name...





SAP ABAP

Apex Apex

C C

C++

CloudFormation

COBOL

C# C#

CSS

X Flex

GO Go

👙 Java

Js JavaScript

Kotlin

Kubernetes

Objective C

PHP PHP

PL/I

PL/SQL

Python

RPG 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

R Bug (14)

Tags

All rules 119 🔓 Vulnerability 3
Hard-coded credentials are security- sensitive
Security Hotspot
Methods and field names should not be the same or differ only by capitalization
Code Smell
Cipher algorithms should be robust
Using weak hashing algorithms is security-sensitive
Security Hotspot
Cognitive Complexity of functions should not be too high
Code Smell
"try!" should not be used
String literals should not be duplicated
Code Smell
Functions and closures should not be empty
Code Smell
Collection elements should not be replaced unconditionally
j Bug
Collection sizes comparisons should make sense
∰ Bug
All branches in a conditional structure should not have exactly the same implementation
₩ Bug

Infix operators that end with "=" should

update their left operands

Precedence and associativity of standard operators should not be

🙀 Bug

changed

```
Enumeration members should
                                        Analyze your code
comply with a naming
convention
☼ Code Smell ♥ Minor ②
                               convention
Shared coding conventions allow teams to collaborate efficiently. This rule
checks that all enumeration member names match a provided regular
expression.
Noncompliant Code Example
With default provided regular expression: ^[a-z][a-zA-z0-9]*$
 enum SomeEnumeration {
      case SomeMember // Non-Compliant
 }
Compliant Solution
 enum SomeEnumeration {
      case someMember
 Available In:
                                               Developer
sonarlint ⊖ | sonarcloud ↔ | sonarqube
```

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