

RPG

Ruby

Scala

Swift

Text

T-SQL

VB.NET

VB6

XML

Terraform

TypeScript



Go static code analysis

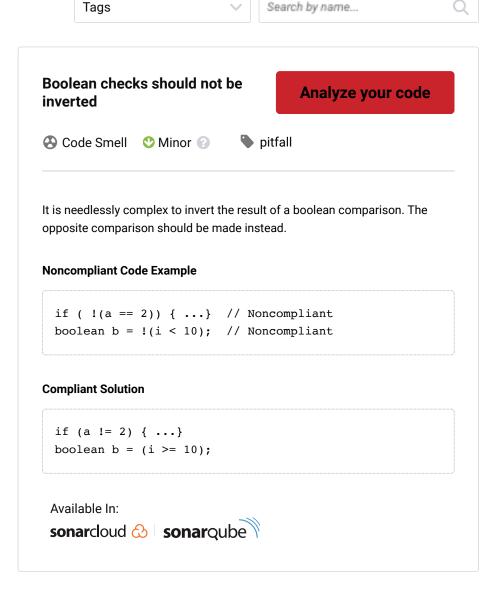
Unique rules to find Bugs, Security Hotspots, and Code Smells in your GO code

	Т
Hard-coded credentials are security- sensitive	Boolea
Security Hotspot	inverte
Cognitive Complexity of functions should not be too high	⊗ Code
Code Smell	It is need
String literals should not be duplicated Code Smell	Noncom
Functions should not be empty	if (
☼ Code Smell	Complia
All branches in a conditional structure should not have exactly the same implementation	if (a
n Bug	
"=+" should not be used instead of "+="	Availat sonar
Related "if/else if" statements should not have the same condition	© 2008-2
n Bug	SONAI tradem pro
Identical expressions should not be used on both sides of a binary operator	b.ot
n Bug	
All code should be reachable	
n Bug	
Variables should not be self-assigned	
n Bug	
Functions should not have identical implementations	
Code Smell	

Two branches in a conditional structure should not have exactly the

same implementation

Code Smell



Code Smell (29)

© 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

"switch" statements should not have too many "case" clauses
Code Smell
Track uses of "FIXME" tags
Redundant pairs of parentheses should be removed
☼ Code Smell
Nested blocks of code should not be left empty
Functions should not have too many parameters
Using hardcoded IP addresses is security-sensitive
Security Hotspot
Multi-line comments should not be empty
Boolean checks should not be inverted
Local variable and function parameter names should comply with a naming convention
☼ Code Smell
Boolean literals should not be redundant