Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
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

Go static code analysis

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

All rules ㊳   🐞 Bug ⑦   🛡 Security Hotspot ②   ☢ Code Smell ㉙

Tags ⌄   Search by name... 🔍

Hard-coded credentials are security-sensitive

🛡 Security Hotspot

Cognitive Complexity of functions should not be too high

☢ Code Smell

String literals should not be duplicated

☢ Code Smell

Functions should not be empty

☢ Code Smell

All branches in a conditional structure should not have exactly the same implementation

🐞 Bug

"=+" should not be used instead of "+="

🐞 Bug

Related "if/else if" statements should not have the same condition

🐞 Bug

Identical expressions should not be used on both sides of a binary operator

🐞 Bug

All code should be reachable

🐞 Bug

Variables should not be self-assigned

🐞 Bug

Functions should not have identical implementations

☢ Code Smell

Two branches in a conditional structure should not have exactly the same implementation

☢ Code Smell

"switch" statements should not have too many "case" clauses

☢ Code Smell

Octal values should not be used   **Analyze your code**

☢ Code Smell   ❗ Blocker ?   🏷 pitfall

Integer literals starting with a zero are octal rather than decimal values. While using octal values is fully supported, most developers do not have experience with them. They may not recognize octal values as such, mistaking them instead for decimal values.

Noncompliant Code Example

```
func printTen() {
        myNumber := 010 // Noncompliant. myNumber will hold 8
        fmt.Println(myNumber)
}
```

Compliant Solution

```
func printTen() {
        myNumber := 10
        fmt.Println(myNumber)
}
```

Available In:

sonarcloud ☁ | sonarqube

Track uses of "FIXME" tags

♨ Code Smell

Redundant pairs of parentheses should be removed

♨ Code Smell

Nested blocks of code should not be left empty

♨ Code Smell

Functions should not have too many parameters

♨ Code Smell

Using hardcoded IP addresses is security-sensitive

🛡 Security Hotspot

Multi-line comments should not be empty

♨ Code Smell

Boolean checks should not be inverted

♨ Code Smell

Local variable and function parameter names should comply with a naming convention

♨ Code Smell

Boolean literals should not be redundant

♨ Code Smell

Function and method names should comply with a naming convention