




































-  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
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

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

Analyze your code

Code SmellMajor?design suspicious

Having two cases in a switch statement or two branches in an if chain with the same implementation is at best duplicate code, and at worst a coding error. If the same logic is truly needed for both instances, then in an if chain they should be combined, or for a switch, one should fall through to the other.

Noncompliant Code Example

```
switch i {
case 1:
    doFirstThing()
    doSomething()
case 2:
    doSomethingElse()
case 3: // Noncompliant; duplicates case 1's implementat
    doFirstThing()
    doSomething()
default:
    doTheRest()
}

if a >= 0 && a < 10 {
    doFirstThing()
    doSomething()
} else if a >= 10 && a < 20 {
    doSomethingElse()
} else if a >= 20 && a < 50 {
    doFirstThing()
    doSomething() // Noncompliant; duplicates first
} else {
    doTheRest()
}
```

Exceptions

Blocks in an if chain or case blocks that contain a single line of code are ignored.

```
if a == 1 {
    doSomething() //no issue, usually this is done on pur
} else if a == 2 {
    doSomethingElse()
} else {
    doSomething()
}
```

Available In: sonarcloud | sonarqube