




Secrets


ABAP


Apex


C


C++


CloudFormation


COBOL


C#


CSS


Flex


Go


HTML


Java


JavaScript


Kotlin


Objective C


PHP


PL/I


PL/SQL


Python


RPG


Ruby


Scala


Swift


Terraform


Text


TypeScript

T-SQL

VB.NET

VB6

XML



# Scala static code analysis

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

All rules 41


Bug 6

Security Hotspot 2


Code Smell 33


Tags ▾

Search by name... 🔍


Bug


Methods should not have identical implementations

Code Smell


Code Smell


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

Code Smell


Code Smell


"match" expressions should not have too many "case" clauses

Code Smell


Code Smell


Sections of code should not be commented out

Code Smell

Code Smell

Unused function parameters should be removed

Code Smell

Code Smell

Unused "private" methods should be removed

Code Smell

Code Smell

Track uses of "FIXME" tags

Code Smell

Code Smell

Nested blocks of code should not be left empty

Code Smell

Code Smell

Functions should not have too many parameters

Code Smell

Code Smell

Collapsible "if" statements should be merged




Code Smell

Code Smell

Using hardcoded IP addresses is security-sensitive

## Related "if"/"else if" statements and "case" in a "match" should not have the same condition

Analyze your code

 Bug  Major  unused pitfall

A match and a chain of `if/else if` statements is evaluated from top to bottom. At most, only one branch will be executed: the first one with a condition that evaluates to `true`.

Therefore, duplicating a condition automatically leads to dead code. Usually, this is due to a copy/paste error. At best, it's simply dead code and at worst, it's a bug that is likely to induce further bugs as the code is maintained, and obviously it could lead to unexpected behavior.

For a match, the second case will never be executed, rendering it dead code. Worse there is the risk in this situation that future maintenance will be done on the dead case, rather than on the one that's actually used.

### Noncompliant Code Example

```
if (param == 1) {
  openWindow
} else if (param == 2) {
  closeWindow
} else if (param == 1) { // Noncompliant
  moveWindowToTheBackground
}

param match {
  case 1 =>
    // ...
  case 3 =>
    // ...
  case 1 => // Noncompliant
    // ...
  case _ =>
    //...
}
```






### Compliant Solution

```
if (param == 1) {
  openWindow
} else if (param == 2) {
  closeWindow
} else if (param == 3) {
  moveWindowToTheBackground
}

param match {
  case 1 =>
```

https://rules.sonarsource.com/scala/RSPEC-1862

1/2

 Security Hotspot
<b>Multi-line comments should not be empty</b>  Code Smell
<b>Boolean checks should not be inverted</b>  Code Smell
<b>Unused local variables should be removed</b>  Code Smell
<b>Local variable and function parameter names should comply with a naming convention</b>  Code Smell

```
// ...
case 3 =>
// ...
case _ =>
// ...
}
```

Available In:

 |  | 

© 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](#)