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

C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code

All rules 409 | Vulnerability 34 | Bug 76 | Security Hotspot 28 | Code Smell 271 | Quick Fix 52

Tags

Search by name...

Method calls should not resolve ambiguously to overloads with "params"
Code Smell

"catch" clauses should do more than rethrow
Code Smell

Generic exceptions should not be ignored
Code Smell

Mutable fields should not be "public static"
Code Smell

Enumeration type names should not have "Flags" or "Enum" suffixes
Code Smell

Enumeration types should comply with a naming convention
Code Smell

Trivial properties should be auto-implemented
Code Smell

Runtime type checking should be simplified
Code Smell

Boolean checks should not be inverted
Code Smell

Inheritance list should not be redundant
Code Smell

Redundant casts should not be used
Code Smell

Strings should not be concatenated

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

Analyze your code

Bug | Major | unused pitfall

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.

Noncompliant Code Example

```
if (param == 1)
{
  OpenWindow();
}
else if (param == 2)
{
  CloseWindow();
}
else if (param == 1) // Noncompliant
{
  MoveWindowToTheBackground();
}
```

Compliant Solution

```
if (param == 1)
{
  OpenWindow();
}
else if (param == 2)
{
  CloseWindow();
}
else if (param == 3)
{
  MoveWindowToTheBackground();
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Strings should not be concatenated using '+' in a loop

Code Smell

Unused local variables should be removed

Code Smell

Private fields only used as local variables in methods should become local variables

Code Smell

A "while" loop should be used instead of a "for" loop

Code Smell