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... 🔍

"protected" members

⊙ Code Smell

Underscores should be used to make large numbers readable

⊙ Code Smell

"ToString()" calls should not be redundant

⊙ Code Smell

"==" should not be used when "Equals" is overridden

⊙ Code Smell

An abstract class should have both abstract and concrete methods

⊙ Code Smell

Multiple variables should not be declared on the same line

⊙ Code Smell

Culture should be specified for "string" operations

⊙ Code Smell

"switch" statements should have at least 3 "case" clauses

⊙ Code Smell

break statements should not be used except for switch cases

⊙ Code Smell

String literals should not be duplicated

⊙ Code Smell

Files should contain an empty newline at the end

⊙ Code Smell

Unused "using" should be removed

⊙ Code Smell

Sequential tests should not check the same condition

Analyze your code

⊙ Code Smell   ⚡ Minor ❓   🏷 suspicious clumsy

When the same condition is checked twice in a row, it is either confusing - why have separate checks? - or an error - some other condition should have been checked in the second test.

Noncompliant Code Example

```
if (a == b)
{
  doTheThing(b);
}
if (a == b) // Noncompliant; is this really what was intende
{
  doTheThing(c);
}
```

Compliant Solution

```
if (a == b)
{
  doTheThing(b);
  doTheThing(c);
}
```

or

```
if (a == b)
{
  doTheThing(b);
}
if (b == c)
{
  doTheThing(c);
}
```

Exceptions

Since it is a common pattern to test a variable, reassign it if it fails the test, then re-test it, that pattern is ignored.

Available In:

sonarlint | sonarcloud | sonarqube

A close curly brace should be located at the beginning of a line

⊗ Code Smell

Tabulation characters should not be used

⊗ Code Smell

Methods and properties should be named in PascalCase

⊗ Code Smell

Track uses of in-source issue suppressions

⊗ Code Smell