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...

be used for value types
Bug

Doubled prefix operators "!!" and "~~" should not be used
Bug

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

"NaN" should not be used in comparisons
Bug

Conditionally executed code should be reachable
Bug

Null pointers should not be dereferenced
Bug

For-loop conditions should be true at least once
Bug

A "for" loop update clause should move the counter in the right direction
Bug

"ToString()" method should not return null
Bug

Return values from functions without side effects should not be ignored
Bug

Values should not be uselessly incremented
Bug

Collections should not be passed as arguments to their own methods

Inherited member visibility should not be decreased

Analyze your code

Code Smell | Critical | pitfall

Changing an inherited member to `private` will not prevent access to the base class implementation.

This rule raises an issue when a `private` method in an unsealed type has a signature that is identical to a `public` method declared in a base type.

Noncompliant Code Example

```
using System;

namespace MyLibrary
{
  public class Foo
  {
    public void SomeMethod(int count) { }
  }
  public class Bar:Foo
  {
    private void SomeMethod(int count) { } // Noncompliant
  }
}
```

Compliant Solution

```
using System;

namespace MyLibrary
{
  public class Foo
  {
    public void SomeMethod(int count) { }
  }
  public sealed class Bar : Foo
  {
    private void SomeMethod(int count) { }
  }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

© 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

Bug

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

Bug

Objects should not be created to be dropped immediately without being used

Bug

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

Bug

Loops with at most one iteration