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...

Using non-standard cryptographic algorithms is security-sensitive

Security Hotspot

Using pseudorandom number generators (PRNGs) is security-sensitive

Security Hotspot

Parameter names should match base declaration and other partial definitions

Code Smell

"ValueTask" should be consumed correctly

Code Smell

String offset-based methods should be preferred for finding substrings from offsets

Code Smell

"default" clauses should be first or last

Code Smell

Unread "private" fields should be removed

Code Smell

Base class methods should not be hidden

Code Smell

Inherited member visibility should not be decreased

Code Smell

Threads should not lock on objects with weak identity

Code Smell

A conditionally executed single line should be denoted by indentation

Code Smell

Conditionals should start on new lines

TestCases should contain tests

Analyze your code

Code Smell Blocker tests unused confusing

There's no point in having a test class without any test methods.This could lead a maintainer to assume a class is covered by tests even though it is not.

Supported test frameworks are `NUnit` and `MSTest` (not applicable to `xUnit`).

This rule will raise an issue when any of these conditions are met:

- For **NUnit**, a class is marked with `TestFixture` but does not contain any method marked with `Test`, `TestCase`, `TestCaseSource` or `Theory`.
- For **MSTest**, a class is marked with `TestClass` but does not contain any method marked with `TestMethod` or `DataTestMethod`.

Noncompliant Code Example

```
[TestFixture]
public class SomeClassTest { } // Noncompliant - no test

[TestClass]
public class SomeOtherClassTest { } // Noncompliant - no tes
```

Compliant Solution

```
[TestFixture]
public class SomeClassTest
{
    [Test]
    public void SomeMethodShouldReturnTrue() { }
}

[TestClass]
public class SomeOtherClassTest
{
    [TestMethod]
    public void SomeMethodShouldReturnTrue() { }
}
```

Exceptions

- abstract classes
- derived classes that inherit from a base class that does have test methods
- in **MSTest**, classes that contain methods marked with either `AssemblyInitialize` or `AssemblyCleanup`.

Available In:

sonarlint | sonarcloud | sonarqube

Conditionals should start on new lines

Code Smell

Assemblies should have version information

Code Smell

Exception types should be "public"

Code Smell

Cognitive Complexity of methods should not be too high

Code Smell

"params" should not be introduced on overrides