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...

security-sensitive

Security Hotspot

Using clear-text protocols is security-sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot

Configuring loggers is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot

Disabling CSRF protections is security-sensitive

Security Hotspot

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

Tests should include assertions

Analyze your code

Code Smell | Blocker | tests

A test case without assertions ensures only that no exceptions are thrown. Beyond basic runnability, it ensures nothing about the behavior of the code under test.

This rule raises an exception when no assertions from any of the following frameworks are found in a test:

- `MSTest`
- `NUnit`
- `xUnit`
- `FluentAssertions` (4.x and 5.x)
- `NFluent`
- `NSubstitute`
- `Shoudly`

Noncompliant Code Example

```
[TestMethod]
public void MyMethod_WhenSomething_ExpectsSomething()
{
    var myClass = new Class();
    var result = myClass.GetFoo();
}
```

Compliant Solution

```
[TestMethod]
public void MyMethod_WhenSomething_ExpectsSomething()
{
    var myClass = new Class();
    var result = myClass.GetFoo();
    Assert.IsTrue(result);
}
```

Exceptions

To create a custom assertion method declare an attribute with name `AssertionMethodAttribute` and mark the method with it:

```
using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;

[TestClass]
public class CustomTest
{
    [TestMethod]
    public void TestMethod1() => Validator.CustomMethod(42);
}
```

"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

```
public static class Validator
{
    [AssertionMethod]
    public static void CustomMethod(int value) { }
}

public class AssertionMethodAttribute : Attribute { }
```

Available In:

sonarlint sonarcloud sonarqube