

- 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

The length returned from a stream read should be checked

Analyze your code

Bug Minor ?

You cannot assume that any given stream reading call will fill the `byte[]` passed in to the method with the number of bytes requested. Instead, you must check the value returned by the `read` method to see how many bytes were read. Fail to do so, and you introduce a bug that is both harmful and difficult to reproduce.

This rule raises an issue when a `Stream.Read` or a `Stream.ReadAsync` method is called, but the return value is not checked.

Noncompliant Code Example





```
public void DoSomething(string fileName)
{
    using (var stream = File.Open(fileName, FileMode.Open))
    {
        var result = new byte[stream.Length];
        stream.Read(result, 0, (int)stream.Length); // Noncompliant
        // ... do something with result
    }
}
```

Compliant Solution

```
public void DoSomething(string fileName)
{
    using (var stream = File.Open(fileName, FileMode.Open))
    {
        var buffer = new byte[1024];
        using (var ms = new MemoryStream())
        {
            int read;
            while ((read = stream.Read(buffer, 0, buffer.Length)) > 0)
            {
                ms.Write(buffer, 0, read);
            }
            // ... do something with ms
        }
    }
}
```

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