

- 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... 🔍

Methods should not return constants

Code Smell

Attribute, EventArgs, and Exception type names should end with the type being extended

Code Smell

Loops should be simplified with "LINQ" expressions

Code Smell

Namespaces should not be empty

Code Smell

Non-derived "private" classes and records should be "sealed"

Code Smell

"string.IsNullOrEmpty" should be used

Code Smell

Implementations should be provided for "partial" methods

Code Smell

Duplicate casts should not be made

Code Smell

Methods should not return values that are never used

Code Smell

Caller information arguments should not be provided explicitly

Code Smell

Method calls should not resolve ambiguously to overloads with "params"

Code Smell

"catch" clauses should do more than rethrow

Code Smell

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

Analyze your code

Bug Major ?

A `for` loop with a counter that moves in the wrong direction is not an infinite loop. Because of wraparound, the loop will eventually reach its stop condition, but in doing so, it will run many, many more times than anticipated, potentially causing unexpected behavior.

Noncompliant Code Example

```
public void DoSomething(string[] strings)
{
    for (int i = 0; i < strings.Length; i--) // Noncompliant
    {
        string s = strings[i]; // IndexOutOfRangeException when
        // do stuff
    }
}
```

Compliant Solution

```
public void DoSomething(string[] strings)
{
    for (int i = 0; i < strings.Length; i++)
    {
        string s = strings[i];
        // do stuff
    }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Generic exceptions should not be ignored

 Code Smell

Mutable fields should not be "public static"

 Code Smell

Enumeration type names should not have "Flags" or "Enum" suffixes

 Code Smell

Enumeration types should comply with a naming convention

 Code Smell