

- 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...

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

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

Trivial properties should be auto-implemented

Code Smell

Runtime type checking should be simplified

Values should not be uselessly incremented

Analyze your code

Bug Major unused

A value that is incremented or decremented and then not stored is at best wasted code and at worst a bug.

Noncompliant Code Example

```
public int PickNumber()
{
    int i = 0;
    int j = 0;

    i = i++; // Noncompliant; i is still zero

    return j++; // Noncompliant; 0 returned
}
```





Compliant Solution

```
public int PickNumber()
{
    int i = 0;
    int j = 0;

    i++;
    return ++j;
}
```

Available In:

sonarlint | sonarcloud | sonarqube

 Code Smell
Boolean checks should not be inverted  Code Smell
Inheritance list should not be redundant  Code Smell
Redundant casts should not be used  Code Smell
Strings should not be concatenated using '+' in a loop