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...

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
☢ Code Smell

Assemblies should have version information
☢ Code Smell

Exception types should be "public"

Short-circuit logic should be used in boolean contexts

[Analyze your code]

☢ Code Smell   ⊘ Blocker ?   Quick Fix ?

The use of non-short-circuit logic in a boolean context is likely a mistake - one that could cause serious program errors as conditions are evaluated under the wrong circumstances.

Noncompliant Code Example

```
if (GetTrue() | GetFalse()) // Noncompliant; both sides eval
{
}
```

Compliant Solution

```
if (GetTrue() || GetFalse()) // true short-circuit logic
{
}
```

Available In:

sonarlint | sonarcloud | sonarqube

⊗ Code Smell

Cognitive Complexity of methods should not be too high

⊗ Code Smell

"params" should not be introduced on overrides

⊗ Code Smell

"[DefaultValue]" should not be used when "[DefaultParameterValue]" is meant

⊗ Code Smell

"[Optional]" should not be used on "ref" or "out" parameters