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...

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

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

Code Smell

Unused local variables should be removed

Code Smell

Private fields only used as local

Objects should not be created to be dropped immediately without being used

Analyze your code

Bug Major

There is no good reason to create a new object to not do anything with it. Most of the time, this is due to a missing piece of code and so could lead to an unexpected behavior in production.

If it was done on purpose because the constructor has side-effects, then that side-effect code should be moved into a separate, static method and called directly.

Noncompliant Code Example

```
if (x < 0)
  new ArgumentException("x must be nonnegative");
```

Compliant Solution

```
if (x < 0)
  throw new ArgumentException("x must be nonnegative");
```

Available In:

sonarlint sonarcloud sonarqube

variables in methods should become local variables

Code Smell

A "while" loop should be used instead of a "for" loop

Code Smell

"Equals" and the comparison operators should be overridden when implementing "IComparable"

Code Smell

Nested code blocks should not be used

Code Smell