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 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
Code Smell

Boolean checks should not be inverted
Code Smell

Collections should not be passed as arguments to their own methods

Analyze your code

Bug | Major

Passing a collection as an argument to the collection's own method is either an error - some other argument was intended - or simply nonsensical code.

Further, because some methods require that the argument remain unmodified during the execution, passing a collection to itself can result in an unexpected behavior.

Noncompliant Code Example

```
var list = new List<int>();

list.AddRange(list); // Noncompliant
list.Concat(list); // Noncompliant

list.Union(list); // Noncompliant; always returns list
list.Except(list); // Noncompliant; always empty
list.Intersect(list); // Noncompliant; always list
list.SequenceEqual(list); // Noncompliant; always true

var set = new HashSet<int>();
set.UnionWith(set); // Noncompliant; no changes
set.ExceptWith(set); // Noncompliant; always empty
set.IntersectWith(set); // Noncompliant; no changes
set.IsProperSubsetOf(set); // Noncompliant; always false
set.IsProperSupersetOf(set); // Noncompliant; always false
set.IsSubsetOf(set); // Noncompliant; always true
set.IsSupersetOf(set); // Noncompliant; always true
set.Overlaps(set); // Noncompliant; always true
set.SetEquals(set); // Noncompliant; always true
set.SymmetricExceptWith(set); // Noncompliant; always empty
```

Available In:

sonarlint | sonarcloud | sonarqube

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