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 not duplicate the names of their methods
⊘ Code Smell

Track use of "NotImplementedException"
⊘ Code Smell

Empty "default" clauses should be removed
⊘ Code Smell

Redundant property names should be omitted in anonymous classes
⊘ Code Smell

Declarations and initializations should be as concise as possible
⊘ Code Smell

Default parameter values should not be passed as arguments
⊘ Code Smell

Constructor and destructor declarations should not be redundant
⊘ Code Smell

Method parameters should be declared with base types
⊘ Code Smell

The simplest possible condition syntax should be used
⊘ Code Smell

Redundant parentheses should not be used
⊘ Code Smell

"GC.SuppressFinalize" should not be invoked for types without destructors
⊘ Code Smell

Members should not be initialized to

"string.ToCharArray()" should not be called redundantly

Analyze your code

🐛 Bug   ⊘ Minor ?   Quick Fix ?   🏷 clumsy

`ToCharArray` can be omitted when the operation on the array could have been done directly on the string, such as when iterating over the characters in a string, and when accessing a character in a string via an array index. In those cases, explicit `ToCharArray` calls should be omitted.

Noncompliant Code Example

```
string str = "some string";
foreach (var c in str.ToCharArray()) // Noncompliant
{
  // ...
}
```

Compliant Solution

```
string str = "some string";
foreach (var c in str)
{
  // ...
}
```

Available In:

sonarlint ⊙ | sonarcloud ⚛ | sonarqube ⋙

Members should not be initialized to default values

⊗ Code Smell

Sequential tests should not check the same condition

⊗ Code Smell

Redundant modifiers should not be used

⊗ Code Smell

Methods and properties that don't access instance data should be static

⊗ Code Smell

"Exception" should not be caught