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...

Local variables should not shadow class fields
Code Smell

Redundant pairs of parentheses should be removed
Code Smell

Inheritance tree of classes should not be too deep
Code Smell

Nested blocks of code should not be left empty
Code Smell

Methods should not have too many parameters
Code Smell

Collapsible "if" statements should be merged
Code Smell

OS commands should not be vulnerable to argument injection attacks
Vulnerability

Logging should not be vulnerable to injection attacks
Vulnerability

Empty collections should not be accessed or iterated
Bug

Mutable, non-private fields should not be "readonly"
Bug

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

"base.Equals" should not be used to

Methods with "Pure" attribute should return a value

Analyze your code

Bug Major

Marking a method with the `Pure` attribute specifies that the method doesn't make any visible changes; thus, the method should return a result, otherwise the call to the method should be equal to no-operation. So `Pure` on a `void` method is either a mistake, or the method doesn't do any meaningful task.

Noncompliant Code Example

```
class Person
{
  private int age;
  [Pure] // Noncompliant. In this case the method makes a po
  void ConfigureAge(int age)
  {
    ...
    this.age = age;
  }
  ...
}
```

Compliant Solution

```
class Person
{
  private int age;

  void ConfigureAge(int age)
  {
    ...
    this.age = age;
  }
  ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube

"base.Equals" should not be used to check for reference equality in "Equals" if "base" is not "object"

Bug

Property assignments should not be made for "readonly" fields not constrained to reference types

Bug

Flags enumerations should explicitly initialize all their members

Bug

"GetHashCode" should not reference mutable fields

Bug