

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C# 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...

Bug

The length returned from a stream read should be checked

Bug

Method parameters, caught exceptions and foreach variables' initial values should not be ignored

Bug

Controlling permissions is security-sensitive

Security Hotspot

Writing cookies is security-sensitive

Security Hotspot

Methods should be named according to their synchronicities

Code Smell

Extensions should be in separate namespaces

Code Smell

Extension methods should not extend "object"

Code Smell

Operator overloads should have named alternatives

Code Smell

Non-abstract attributes should be sealed

Code Smell

Overloads with a "StringComparison" parameter should be used

Code Smell

Overloads with a "CultureInfo" or an "IFormatProvider" parameter should be used

Code Smell

"Obsolete" attributes should include explanations

Analyze your code

Code Smell Major obsolete bad-practice

The Obsolete attribute can be applied with or without arguments, but marking something Obsolete without including advice as to why it's obsolete or on what to use instead will lead maintainers to waste time trying to figure those things out - every single time the warning is encountered.

Noncompliant Code Example

```
public class Car
{
    [Obsolete] // Noncompliant
    public void CrankEngine(int turnsOfCrank)
    { ... }
}
```


Compliant Solution

```
public class Car
{
    [Obsolete("Replaced by the automatic starter")]
    public void CrankEngine(int turnsOfCrank)
    { ... }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Types should not extend outdated base types

 Code Smell

Properties should be preferred

 Code Smell

Generics should be used when appropriate

 Code Smell

Type names should not match namespaces

 Code Smell