

- 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... 🔍

"protected" members

Code Smell

Underscores should be used to make large numbers readable

Code Smell

"ToString()" calls should not be redundant

Code Smell

"==" should not be used when "Equals" is overridden

Code Smell

An abstract class should have both abstract and concrete methods

Code Smell

Multiple variables should not be declared on the same line

Code Smell

Culture should be specified for "string" operations

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

break statements should not be used except for switch cases

Code Smell

String literals should not be duplicated

Code Smell

Files should contain an empty newline at the end

Code Smell

Unused "using" should be removed

Code Smell

Mutable fields should not be "public static"

Analyze your code

Code Smell Minor cwe unpredictable

public static mutable fields of classes which are accessed directly should be protected to the degree possible. This can be done by reducing the accessibility of the field or by changing the return type to an immutable type.

This rule raises issues for public static fields with a type inheriting/implementing System.Array or System.Collections.Generic.ICollection<T>.

Noncompliant Code Example

```
public class A
{
    public static string[] strings1 = {"first","second"}; //
    public static List<String> strings3 = new List<String>();
    // ...
}
```

Compliant Solution

```
public class A
{
    protected static string[] strings1 = {"first","second"};
    protected static List<String> strings3 = new List<String>()
    // ...
}
```

Exceptions

No issue is reported:

- If the type of the field inherits/implements one (at least) of the following types:
 - System.Collections.ObjectModel.ReadOnlyCollection<T>
 - System.Collections.ObjectModel.ReadOnlyDictionary<TKey, TValue>
 - System.Collections.Immutable.IImmutableArray<T>
 - System.Collections.Immutable.IImmutableDictionary<TKey, TValue>
 - System.Collections.Immutable.IImmutableList<T>
 - System.Collections.Immutable.IImmutableSet<T>
 - System.Collections.Immutable.IImmutableStack<T>
 - System.Collections.Immutable.IImmutableQueue<T>
- If the field is readonly and is initialized inline with an immutable type (i.e. inherits/implements one of the types in the previous list) or null.

See

- [MITRE, CWE-582](#) - Array Declared Public, Final, and Static
- [MITRE, CWE-607](#) - Public Static Final Field References Mutable Object

A close curly brace should be located at the beginning of a line

 Code Smell

Tabulation characters should not be used

 Code Smell

Methods and properties should be named in PascalCase

 Code Smell

Track uses of in-source issue suppressions

 Code Smell

Available In:

sonarlint  | **sonarcloud**  | **sonarqube** 

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)