

- 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... 🔍

security-sensitive

Security Hotspot

Using command line arguments is security-sensitive

Security Hotspot

Using Sockets is security-sensitive

Security Hotspot

Encrypting data is security-sensitive

Security Hotspot

Using regular expressions is security-sensitive

Security Hotspot

Interface methods should be callable by derived types

Code Smell

Child class fields should not differ from parent class fields only by capitalization

Code Smell

Pointers to unmanaged memory should not be visible

Code Smell

Number patterns should be regular

Code Smell

"out" and "ref" parameters should not be used

Code Smell

Unchanged local variables should be "const"

Code Smell

"ConfigureAwait(false)" should be used

Code Smell

"P/Invoke" methods should not be visible

Analyze your code

Code Smell Major pinvoke

Methods marked with the `System.Runtime.InteropServices.DllImportAttribute` attribute use Platform Invocation Services to access unmanaged code and should not be exposed. Keeping them private or internal makes sure that their access is controlled and properly managed.

This rule raises an issue when a method declared with `DllImport` is public or protected.

Noncompliant Code Example

```
using System;
using System.Runtime.InteropServices;

namespace MyLibrary
{
    public class Foo
    {
        [DllImport("kernel32.dll", CharSet = CharSet.Unicode)
        public static extern bool RemoveDirectory(string name)
    }
}
```

Compliant Solution

```
using System;
using System.Runtime.InteropServices;

namespace MyLibrary
{
    public class Foo
    {
        [DllImport("kernel32.dll", CharSet = CharSet.Unicode)
        private static extern bool RemoveDirectory(string name)
    }
}
```

Available In:
sonarlint | sonarcloud | sonarqube

"interface" instances should not be cast to concrete types

 Code Smell

Literal boolean values should not be used in assertions

 Code Smell

Optional parameters should not be used

 Code Smell

Public constant members should not be used

 Code Smell