

- 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...



Parameters with "[DefaultParameterValue]" attributes should also be marked "[Optional]"

Code Smell

Interfaces should not simply inherit from base interfaces with colliding members

Code Smell

Variables should not be checked against the values they're about to be assigned

Code Smell

Methods should not return constants

Code Smell

Attribute, EventArgs, and Exception type names should end with the type being extended

Code Smell

Loops should be simplified with "LINQ" expressions

Code Smell

Namespaces should not be empty

Code Smell

Non-derived "private" classes and records should be "sealed"

Code Smell

"string.IsNullOrEmpty" should be used

Code Smell

Implementations should be provided for "partial" methods

Code Smell

Duplicate casts should not be made

Code Smell

Methods should not return values that are never used

Null pointers should not be dereferenced

Analyze your code

Bug Major cwe

A reference to `null` should never be dereferenced/accessed. Doing so will cause a `NullReferenceException` to be thrown. At best, such an exception will cause abrupt program termination. At worst, it could expose debugging information that would be useful to an attacker, or it could allow an attacker to bypass security measures.

Noncompliant Code Example

```
object o = null;
if (condition)
{
    M1(o.ToString()); // Noncompliant, always null
}
else
{
    o = new object();
}
M2(o.ToString());
```

Exceptions

Calls to extension methods are not reported because they can still operate on `null` values.





To create a custom null validation method declare an attribute with name `ValidatedNotNullAttribute` and mark the parameter that is validated for null in your method declaration with it:

```
using System;

public sealed class ValidatedNotNullAttribute : Attribute {

    public static class Guard
    {
        public static void NotNull<T>([ValidatedNotNull] this T
        {
            if (value == null)
                throw new ArgumentNullException(name);
        }
    }

    public static class Utils
    {
        public static string ToUpper(string value)
        {
            Guard.NotNull(value, nameof(value));
            if (value == null)
            {
                return value.ToString(); // Compliant, this code
            }
        }
    }
}
```

 Code Smell
Caller information arguments should not be provided explicitly
 Code Smell
Method calls should not resolve ambiguously to overloads with "params"
 Code Smell
"catch" clauses should do more than rethrow
 Code Smell
Generic exceptions should not be ignored

```
        return value.ToUpper();  
    }  
}
```

See

- [MITRE, CWE-476](#) - NULL Pointer Dereference

Available In:

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