

- 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... 🔍

Assemblies should be marked as OLE compliant

Code Smell

"Generic.List" instances should not be part of public APIs

Code Smell

Collections should implement the generic interface

Code Smell

Generic event handlers should be used

Code Smell

Event Handlers should have the correct signature

Code Smell

"Assembly.GetExecutingAssembly" should not be called

Code Smell

Arguments of public methods should be validated against null

Code Smell

Value types should implement "IEquatable<T>"

Code Smell

Finalizers should not be empty

Code Smell

"[ExpectedException]" should not be used

Code Smell

"this" should not be exposed from constructors

Code Smell

Types should not have members with visibility set higher than the type's visibility

Boolean expressions should not be gratuitous

Analyze your code

Code Smell Major ? cwe suspicious redundant

If a boolean expression doesn't change the evaluation of the condition, then it is entirely unnecessary, and can be removed. If it is gratuitous because it does not match the programmer's intent, then it's a bug and the expression should be fixed.

Noncompliant Code Example

```
public void Sample(bool b, bool c, string s)
{
    var a = true;
    if (a) // Noncompliant
    {
        DoSomething();
    }

    if (b && a) // Noncompliant; "a" is always "true"
    {
        DoSomething();
    }

    if (c || !a) // Noncompliant; "!a" is always "false"
    {
        DoSomething();
    }

    string d = null;
    var v1 = d ?? "value"; // Noncompliant, d is always null
    var v2 = s ?? d; // Noncompliant, d is always null and t
}
```

Compliant Solution

```
public void Sample(bool b, bool c, string s)
{
    var a = true;
    if (Foo(a))
    {
        DoSomething();
    }

    if (b)
    {
        DoSomething();
    }

    if (c)
    {
        DoSomething();
    }
}
```

 Code Smell

Fields should be private

 Code Smell

"try" statements with identical "catch" and/or "finally" blocks should be merged

 Code Smell

NullPointerException should not be caught

 Code Smell

Functions should not have too many lines of code

```
var v1 = "value";  
var v2 = s;  
}
```

See

- [MITRE, CWE-571](#) - Expression is Always True
- [MITRE, CWE-570](#) - Expression is Always False

Available In:

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