

- 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... 🔍

Bug

Identical expressions should not be used on both sides of a binary operator

Bug

Loops with at most one iteration should be refactored

Bug

Variables should not be self-assigned

Bug

Constructing arguments of system commands from user input is security-sensitive

Security Hotspot

Deserializing objects without performing data validation is security-sensitive

Security Hotspot

Disabling ASP.NET "Request Validation" feature is security-sensitive

Security Hotspot

Allowing requests with excessive content length is security-sensitive

Security Hotspot

Setting loose file permissions is security-sensitive

Security Hotspot

Formatting SQL queries is security-sensitive

Security Hotspot

Using hardcoded IP addresses is security-sensitive

Security Hotspot

"goto" statement should not be used

Code Smell

"[DefaultValue]" should not be used when "[DefaultParameterValue]" is meant

Analyze your code

Code Smell Critical Quick Fix suspicious

The use of [DefaultValue] with [Optional] has no more effect than [Optional] alone. That's because [DefaultValue] doesn't actually do anything; it merely indicates the intent for the value. More than likely, [DefaultValue] was used in confusion instead of [DefaultParameterValue].

Noncompliant Code Example

```
class MyClass
{
    public void DoStuff([Optional][DefaultValue(4)]int i, in
    {
        Console.WriteLine(i);
    }

    public static void Main()
    {
        new MyClass().DoStuff(); // prints 0
    }
}
```

Compliant Solution

```
class MyClass
{
    public void DoStuff([Optional][DefaultParameterValue(4)]
    {
        Console.WriteLine(i);
    }

    public static void Main()
    {
        new MyClass().DoStuff(); // prints 4
    }
}
```

Available In:
 sonarlint sonarcloud sonarqube

"new Guid()" should not be used

 Code Smell

**Parameter validation in
"async"/"await" methods should be
wrapped**

 Code Smell

**Parameter validation in yielding
methods should be wrapped**

 Code Smell

Events should have proper arguments

 Code Smell