

- 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... 🔍

Console logging should not be used	Vulnerability
Generic parameters not constrained to reference types should not be compared to "null"	Bug
The length returned from a stream read should be checked	Bug
Method parameters, caught exceptions and foreach variables' initial values should not be ignored	Bug
Controlling permissions is security-sensitive	Security Hotspot
Writing cookies is security-sensitive	Security Hotspot
Methods should be named according to their synchronicities	Code Smell
Extensions should be in separate namespaces	Code Smell
Extension methods should not extend "object"	Code Smell
Operator overloads should have named alternatives	Code Smell
Non-abstract attributes should be sealed	Code Smell
Overloads with a "StringComparison" parameter should be used	

Track uses of "FIXME" tags

Analyze your code

- Code Smell
- Major ?
- cwe

FIXME tags are commonly used to mark places where a bug is suspected, but which the developer wants to deal with later.

Sometimes the developer will not have the time or will simply forget to get back to that tag.

This rule is meant to track those tags and to ensure that they do not go unnoticed.

Noncompliant Code Example





```
private int Divide(int numerator, int denominator)
{
    return numerator / denominator; // FIXME denominator va
}
```

See

- [MITRE, CWE-546](#) - Suspicious Comment

Available In:

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

 Code Smell
Overloads with a "CultureInfo" or an "IFormatProvider" parameter should be used  Code Smell
Types should not extend outdated base types  Code Smell
Properties should be preferred  Code Smell
Generics should be used when appropriate