

- 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... 🔍

Vulnerability

Cryptographic keys should be robust

Vulnerability

Weak SSL/TLS protocols should not be used

Vulnerability

Cipher Block Chaining IVs should be unpredictable

Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

Vulnerability

Hashes should include an unpredictable salt

Vulnerability

Non-async "Task/Task<T>" methods should not return null

Bug

Calls to delegate's method "BeginInvoke" should be paired with calls to "EndInvoke"

Bug

"Shared" parts should not be created with "new"

Bug

Getters and setters should access the expected fields

Bug

Right operands of shift operators should be integers

Bug

Shared resources should not be used for locking

Bug

Exceptions should not be thrown from unexpected methods

Analyze your code

Code Smell Blocker pitfall

It is expected that some methods should be called with caution, but others, such as ToString, are expected to "just work". Throwing an exception from such a method is likely to break callers' code unexpectedly.

An issue is raised when an exception is thrown from any of the following:

- Event accessors
- Object.Equals
- IEquatable.Equals
- GetHashCode
- ToString
- static constructors
- Module initializers
- IDisposable.Dispose
- operators ==, !=, <, >, <=, >=
- implicit cast operators

Noncompliant Code Example

```
public override string ToString()
{
    if (string.IsNullOrEmpty(Name))
    {
        throw new ArgumentException("..."); // Noncompliant
    }
    //...
```

Exceptions

System.NotImplementedException and its derivatives are ignored.

System.InvalidOperationException, System.NotSupportedException, and System.ArgumentException and their derivatives are ignored in event accessors.

Available In:

sonarlint | sonarcloud | sonarqube

Locks should be released



Using publicly writable directories is security-sensitive



Using clear-text protocols is security-sensitive



Expanding archive files without controlling resource consumption is security-sensitive

