- 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...

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

Locks should be released

Bug

Using publicly writable directories is security-sensitive

Security Hotspot

Using clear-text protocols is security-sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot

Configuring loggers is security

Method overloads with default parameter values should not overlap

Analyze your code

Code Smell   Blocker   unused pitfall

The rules for method resolution are complex and perhaps not properly understood by all coders. Having overloads with optional parameter values makes the matter even harder to understand.

This rule raises an issue when an overload with default parameter values is hidden by one without the optional parameters.

Noncompliant Code Example

```
public class MyClass
{
  void Print(string[] messages) {...}
  void Print(string[] messages, string delimiter = "\n") {..
}

// ...
MyClass myClass = new MyClass();

myClass.Print(new string[3] {"yes", "no", "maybe"});  // whi
```

Available In:

sonarlint | sonarcloud | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
Privacy Policy

Configuring loggers is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot

Disabling CSRF protections is security-sensitive

Security Hotspot

Using non-standard cryptographic algorithms is security-sensitive

Security Hotspot