sonar RULES

Products

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...

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

Locks should be released
Bug

Using publicly writable directories is security-sensitive
Security Hotspot

Using clear-text protocols is security-

"operator==" should not be overloaded on reference types

Analyze your code

Code Smell | Blocker | pitfall

The use of == to compare two objects is expected to do a reference comparison. That is, it is expected to return `true` if and only if they are the same object instance. Overloading the operator to do anything else will inevitably lead to the introduction of bugs by callers. On the other hand, overloading it to do exactly that is pointless; that's what == does by default.

Noncompliant Code Example

```
public static bool operator== (MyType x, MyType y) // Noncom
{
```

Exceptions

- Classes with overloaded `operator +` or `operator -` methods are ignored.
- Classes that implement `IComparable<T>` or `IEquatable<T>` most probably behave as a value-type objects and so are ignored.

Available In:

sonarlint | sonarcloud | sonarqube

sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot

Configuring loggers is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot