sonar RULES

Products

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...

Server-side requests should not be vulnerable to forging attacks
Vulnerability

Members should not have conflicting transparency annotations
Vulnerability

"PartCreationPolicyAttribute" should be used with "ExportAttribute"
Bug

"ConstructorArgument" parameters should exist in constructors
Bug

Windows Forms entry points should be marked with STAThread
Bug

Collection elements should not be replaced unconditionally
Bug

Exceptions should not be created without being thrown
Bug

Collection sizes and array length comparisons should make sense
Bug

Serialization event handlers should be implemented correctly
Bug

Deserialization methods should be provided for "OptionalField" members
Bug

All branches in a conditional structure should not have exactly the same implementation
Bug

Types should be defined in named namespaces

Shared resources should not be used for locking

Analyze your code

Bug Critical multi-threading

Shared resources should not be used for locking as it increases the chance of deadlocks. Any other thread could acquire (or attempt to acquire) the same lock for another unrelated purpose.

Instead, a dedicated `object` instance should be used for each shared resource, to avoid deadlocks or lock contention.

The following objects are considered as shared resources:

- `this`
- a `Type` object
- a string literal
- a string instance

Noncompliant Code Example

```
public void MyLockingMethod()
{
    lock (this) // Noncompliant
    {
        // ...
    }
}
```

Compliant Solution

```
private readonly object lockObj = new object();

public void MyLockingMethod()
{
    lock (lockObj)
    {
        // ...
    }
}
```

See

Microsoft Documentation: Managed Threading Best Practices

Available In:

sonarlint | sonarcloud | sonarqube

namespaces

Bug

Empty nullable value should not be accessed

Bug

Nullable type comparison should not be redundant

Bug

Methods with "Pure" attribute should return a value

Bug

One-way "OperationContract" methods should have "void" return type