C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code

Deserialization methods should be provided for "OptionalField" members

Analyze your code

Bug Major serialization

Fields marked with `System.Runtime.Serialization.OptionalFieldAttribute` are serialized just like any other field. But such fields are ignored on deserialization, and retain the default values associated with their types. Therefore, deserialization event handlers should be declared to set such fields during the deserialization process.

This rule raises when at least one field with the `System.Runtime.Serialization.OptionalFieldAttribute` attribute is declared but one (or both) of the following event handlers `System.Runtime.Serialization.OnDeserializingAttribute` or `System.Runtime.Serialization.OnDeserializedAttribute` are not present.

Noncompliant Code Example

```
[Serializable]
public class Foo
{
    [OptionalField(VersionAdded = 2)]
    int optionalField = 5;
}
```

Compliant Solution

```
[Serializable]
public class Foo
{
    [OptionalField(VersionAdded = 2)]
    int optionalField = 5;

    [OnDeserializing]
    void OnDeserializing(StreamingContext context)
    {
            optionalField = 5;
    }

    [OnDeserialized]
    void OnDeserialized(StreamingContext context)
    {
        // Set optionalField if dependent on other deseriali
    }
}
```

Available In:

sonarlint sonarcloud sonarqube

Code Smell

Nested blocks of code should not be left empty

Code Smell

Methods should not have too many parameters

Code Smell

Collapsible "if" statements should be merged

Code Smell

OS commands should not be vulnerable to argument injection