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...

"protected" members
Code Smell

Underscores should be used to make large numbers readable
Code Smell

"ToString()" calls should not be redundant
Code Smell

"==" should not be used when "Equals" is overridden
Code Smell

An abstract class should have both abstract and concrete methods
Code Smell

Multiple variables should not be declared on the same line
Code Smell

Culture should be specified for "string" operations
Code Smell

"switch" statements should have at least 3 "case" clauses
Code Smell

break statements should not be used except for switch cases
Code Smell

String literals should not be duplicated
Code Smell

Files should contain an empty newline at the end
Code Smell

Unused "using" should be removed
Code Smell

Serialization constructors should be secured

Analyze your code

Vulnerability | Major | serialization owasp

Because serialization constructors allocate and initialize objects, security checks that are present on regular constructors must also be present on a serialization constructor. Failure to do so would allow callers that could not otherwise create an instance to use the serialization constructor to do this.

This rule raises an issue when a type implements the `System.Runtime.Serialization.ISerializable` interface, is not a delegate or interface, is declared in an assembly that allows partially trusted callers and has a constructor that takes a `System.Runtime.Serialization.SerializationInfo` object and a `System.Runtime.Serialization.StreamingContext` object which is not secured by a security check, but one or more of the regular constructors in the type is secured.

Noncompliant Code Example

```
using System;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Security;
using System.Security.Permissions;

[assembly: AllowPartiallyTrustedCallersAttribute()]
namespace MyLibrary
{
    [Serializable]
    public class Foo : ISerializable
    {
        private int n;

        [FileIOPermissionAttribute(SecurityAction.Demand, Un
        public Foo()
        {
           n = -1;
        }

        protected Foo(SerializationInfo info, StreamingConte
        {
           n = (int)info.GetValue("n", typeof(int));
        }

        void ISerializable.GetObjectData(SerializationInfo i
        {
           info.AddValue("n", n);
        }
    }
}
```

Compliant Solution

A close curly brace should be located at the beginning of a line

Code Smell

Tabulation characters should not be used

Code Smell

Methods and properties should be named in PascalCase

Code Smell

Track uses of in-source issue suppressions

Code Smell

```
using System;
using System.IO;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Security;
using System.Security.Permissions;

[assembly: AllowPartiallyTrustedCallersAttribute()]
namespace MyLibrary
{
    [Serializable]
    public class Foo : ISerializable
    {
        private int n;

        [FileIOPermissionAttribute(SecurityAction.Demand, Un
        public Foo()
        {
           n = -1;
        }

        [FileIOPermissionAttribute(SecurityAction.Demand, Un
        protected Foo(SerializationInfo info, StreamingConte
        {
           n = (int)info.GetValue("n", typeof(int));
        }

        void ISerializable.GetObjectData(SerializationInfo i
        {
           info.AddValue("n", n);
        }
    }
}
```

See

- OWASP Top 10 2021 Category A8 - Software and Data Integrity Failures
- OWASP Top 10 2017 Category A8 - Insecure Deserialization

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