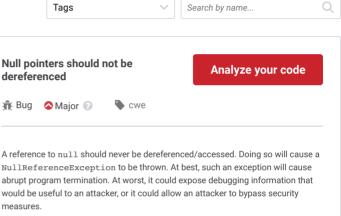


Code Smell

are never used

Methods should not return values that



Null IReferenceExcept ion to be thrown. At best, such an exception will cause abrupt program termination. At worst, it could expose debugging information that would be useful to an attacker, or it could allow an attacker to bypass security measures

Noncompliant Code Example

```
object o = null;
if (condition)
 M1(o.ToString()); // Noncompliant, always null
else
 o = new object();
M2(o.ToString());
```

Exceptions

Calls to extension methods are not reported because they can still operate on null

To create a custom null validation method declare an attribute with name ValidatedNotNullAttribute and mark the parameter that is validated for null in your method declaration with it:

```
using System:
public sealed class ValidatedNotNullAttribute : Attribute {
public static class Guard
    public static void NotNull<T>([ValidatedNotNull] this T
        if (value == null)
            throw new ArgumentNullException(name);
public static class Utils
    public static string ToUpper(string value)
        Guard.NotNull(value, nameof(value));
        if (value == null)
            return value. ToString(); // Compliant, this code
```

Caller information arguments should not be provided explicitly
Code Smell

Method calls should not resolve ambiguously to overloads with "params"

"catch" clauses should do more than rethrow

Code Smell

☼ Code Smell

Generic exceptions should not be ignored

```
return value.ToUpper();
}

See

• MITRE, CWE-476 - NULL Pointer Dereference

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