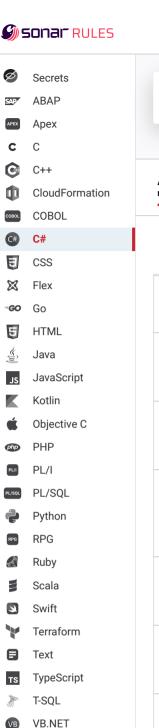
Quick 52 Fix



VB<sub>6</sub>

XML

```
Unique rules to find Bugs, Vulnerabilities, Security
               Hotspots, and Code Smells in your C# code
ΔII
     409
               6 Vulnerability (34)
rules
 Literal suffixes should be upper case
 Code Smell
 Null checks should not be used with
 A Code Smell
 Method overloads should be grouped
 together
 Code Smell
 "params" should be used instead of
  .
'varargs'
 Code Smell
 "static" fields should be initialized
 inline
 Code Smell
 Classes that provide "Equals(<T>)"
 should implement "IEquatable<T>"
 Code Smell
 Jump statements should not be
 Code Smell
 Member initializer values should not
 be redundant
 Code Smell
 Unassigned members should be
 removed
 Code Smell
 Empty "case" clauses that fall through
 to the "default" should be omitted
 Code Smell
 Parameters with '
 [DefaultParameterValue]" attributes
 should also be marked "[Optional]"
 Code Smell
```

C# static code analysis

**#** Bug (76)

```
Tags
                                          Search by name.
"ThreadStatic" fields should not be
                                                Analyze your code
initialized
Rug Major 🕝
                          multi-threading
When an object has a field annotated with ThreadStatic, that field is shared
within a given thread, but unique across threads. Since a class' static initializer is
only invoked for the first thread created, it also means that only the first thread will
have the expected initial values.
Instead, allow such fields to be initialized to their default values or make the
initialization lazy.
Noncompliant Code Example
  public class Foo
    [ThreadStatic]
    public static object PerThreadObject = new object(); // No
Compliant Solution
  public class Foo
    [ThreadStatic]
    public static object _perThreadObject;
    public static object PerThreadObject
      get
         if (_perThreadObject == null)
            perThreadObject = new object();
         return _perThreadObject;
 Available In:
 sonarlint ⊕ | sonarcloud ♦ | sonarqube
```

⊗ Code

Smell

Security

Hotspot

(28)

© 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

Interfaces should not simply inherit from base interfaces with colliding members

Code Smell

Variables should not be checked against the values they're about to be assigned

Code Smell

Methods should not return constants

Code Smell

Attribute, EventArgs, and Exception type names should end with the type being extended

Code Smell