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...

"base.Equals" should not be used to check for reference equality in "Equals" if "base" is not "object"

Bug

Property assignments should not be made for "readonly" fields not constrained to reference types

Bug

Flags enumerations should explicitly initialize all their members

Bug

"GetHashCode" should not reference mutable fields

Bug

Results of integer division should not be assigned to floating point variables

Bug

Integral numbers should not be shifted by zero or more than their number of bits-1

Bug

"Equals(Object)" and "GetHashCode()" should be overridden in pairs

Bug

Having a permissive Cross-Origin Resource Sharing policy is security-sensitive

Security Hotspot

Delivering code in production with debug features activated is security-sensitive

Security Hotspot

Searching OS commands in PATH is security-sensitive

Security Hotspot

Creating cookies without the "HttpOnly" flag is security-sensitive

Static fields should appear in the order they must be initialized

Analyze your code

Bug | Major

Static field initializers are executed in the order in which they appear in the class from top to bottom. Thus, placing a static field in a class above the field or fields required for its initialization will yield unexpected results.

Noncompliant Code Example

```
class MyClass
{
  public static int X = Y; // Noncompliant; Y at this time i
  public static int Y = 42;
}
```

Compliant Solution

```
class MyClass
{
  public static int Y = 42;
  public static int X = Y;
}
```

or

```
class MyClass
{
  public static int X;
  public static int Y = 42;

  static MyClass()
  {
    X = Y;
  }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

"HttpOnly" flag is security-sensitive

Security Hotspot

Creating cookies without the "secure" flag is security-sensitive

Security Hotspot

Literal suffixes should be upper case

Code Smell

Null checks should not be used with "is"

Code Smell

Method overloads should be grouped together