C# static code analysis

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

Redundant parentheses should not be used
Code Smell

"GC.SuppressFinalize" should not be invoked for types without destructors
Code Smell

Members should not be initialized to default values
Code Smell

Sequential tests should not check the same condition
Code Smell

Redundant modifiers should not be used
Code Smell

Methods and properties that don't access instance data should be static
Code Smell

"Exception" should not be caught when not required by called methods
Code Smell

"sealed" classes should not have "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

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

Analyze your code

Bug Minor cwe overflow sans-top25

When division is performed on `int`s, the result will always be an `int`. You can assign that result to a `double`, `float` or `decimal` with automatic type conversion, but having started as an `int`, the result will likely not be what you expect. If the result of `int` division is assigned to a floating-point variable, precision will have been lost before the assignment. Instead, at least one operand should be cast or promoted to the final type before the operation takes place.

Noncompliant Code Example

```
static void Main()
{
  decimal dec = 3/2; // Noncompliant
  Method(3/2); // Noncompliant
}

static void Method(float f) { }
```

Compliant Solution

```
static void Main()
{
  decimal dec = (decimal)3/2;
  Method(3.0F/2);
}

static void Method(float f) { }
```

See

- MITRE, CWE-190 - Integer Overflow or Wraparound
- SANS Top 25 - Risky Resource Management

Available In:

sonarlint | sonarcloud | sonarqube

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