C#

C# static code analysis

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

Recursion should not be infinite

Analyze your code

Bug Blocker suspicious

Recursion happens when control enters a loop that has no exit. This can happen a method invokes itself, when a pair of methods invoke each other, or when `goto` statements are used to move between two segments of code. It can be a useful tool, but unless the method includes a provision to break out of the recursion and `return`, the recursion will continue until the stack overflows and the program crashes.

Noncompliant Code Example

```
int Pow(int num, int exponent)   // Noncompliant; no conditi
{
  num = num * Pow(num, exponent-1);
  return num;  // this is never reached
}

void WhileLoop()   // Noncompliant; no condition under which
{
  while (true)
  {
    var line = Console.ReadLine();
    Console.WriteLine(line);
  }
}

void InternalRecursion(int i)
{
  start:
    goto end;
  end:
    goto start; // Noncompliant; there's no way to break out
}
```

Compliant Solution

```
int Pow(int num, int exponent)
{
  if (exponent > 1) // recursion now conditional and stop-ab
  {
    num = num * Pow(num, exponent-1);
  }
  return num;
}

void WhileLoop()
{
  string line;
  while ((line = Console.ReadLine()) != null) // loop has cl
  {
    Console.WriteLine(line);
```

Cipher Block Chaining IVs should be unpredictable

Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

Vulnerability

Hashes should include an unpredictable salt

Vulnerability

Non-async "Task/Task<T>" methods should not return null

Bug

```
    }
  }
```

Available In:

sonarcloud