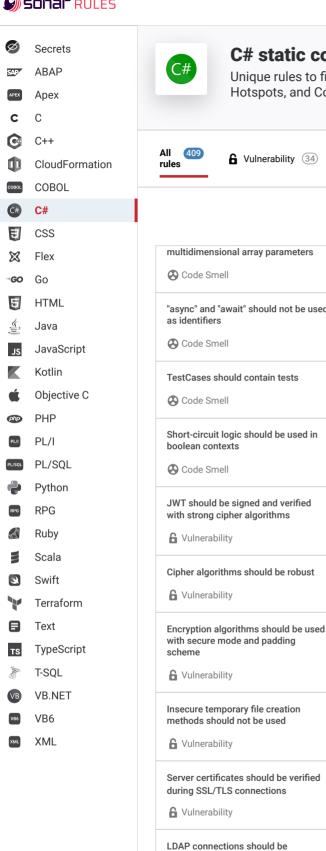
Q





authenticated

Vulnerability

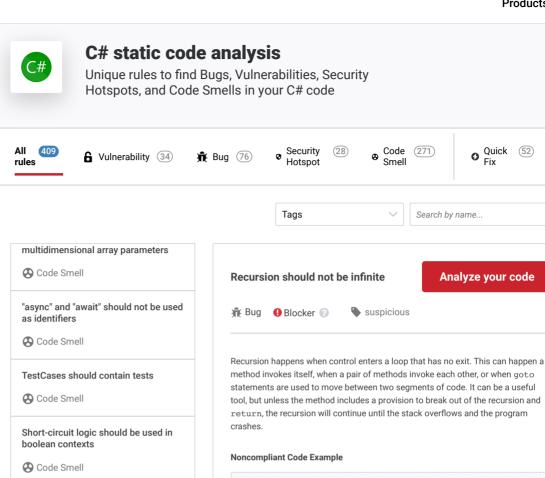
■ Vulnerability

Vulnerability

be used

Cryptographic keys should be robust

Weak SSL/TLS protocols should not



Compliant Solution

start:

goto end;

{

while (true)

```
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);
```

goto start; // Noncompliant; there's no way to break out

int Pow(int num, int exponent) // Noncompliant; no conditi

// Noncompliant; no condition under which

num = num * Pow(num, exponent-1); return num; // this is never reached

var line = Console.ReadLine();

Console.WriteLine(line):

void InternalRecursion(int i)

Cipher Block Chaining IVs should be unpredictable G Vulnerability	}
Deguler commercione chauld not be	Available In:
Regular expressions should not be vulnerable to Denial of Service attacks	Solial Cloud
6 Vulnerability	
	© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR.
Hashes should include an unpredictable salt	SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective
6 Vulnerability	owners. All rights are expressly reserved. <u>Privacy Policy</u>

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

🕦 Bug