

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C# 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...



be assigned to floating point variables



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



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



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



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



Searching OS commands in PATH is security-sensitive



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



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



Literal suffixes should be upper case



Null checks should not be used with "is"



Method overloads should be grouped together



"params" should be used instead of "varargs"

Delegates should not be subtracted

Analyze your code



In C#, delegates can be added together to chain their execution, and subtracted to remove their execution from the chain.

Subtracting a chain of delegates from another one might yield unexpected results as shown hereunder - and is likely to be a bug.

Noncompliant Code Example

```
MyDelegate first, second, third, fourth;
first = () => Console.Write("1");
second = () => Console.Write("2");
third = () => Console.Write("3");
fourth = () => Console.Write("4");

MyDelegate chain1234 = first + second + third + fourth; // C
MyDelegate chain12 = chain1234 - third - fourth; // Compliant

MyDelegate chain14 = first + fourth; // creates a new MyDele
MyDelegate chain23 = chain1234 - chain14; // Noncompliant; (

// The chain sequence of "chain23" will be "1234" instead of
// Indeed, the sequence "1234" does not contain the subsequence
// (but note that "1234" contains both the "1" and "4" subsequence)
chain23 = chain1234 - (first + fourth); // Noncompliant

chain23(); // will print "1234"!
```

Compliant Solution

```
MyDelegate chain23 = chain1234 - first - fourth; // Compliant

chain23(); // will print "23"
```

Available In:

sonarlint sonarcloud sonarqube

 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 redundant

 Code Smell