

- 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...



Equals() base is not object



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



Flags enumerations should explicitly initialize all their members



"GetHashCode" should not reference mutable fields



Results of integer division should not 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



### Classes directly extending "object" should not call "base" in "GetHashCode" or "Equals"

Analyze your code

Bug Major ?

Making a base call in an overriding method is generally a good idea, but not in GetHashCode and Equals for classes that directly extend object because those methods are based on the object reference. Meaning that no two objects that use those base methods will ever be equal or have the same hash.

#### Noncompliant Code Example

```
public class Point
{
    private readonly int x;
    public MyClass(int x)
    {
        this.x = x;
    }
    public override int GetHashCode()
    {
        return x.GetHashCode() ^ base.GetHashCode(); //Noncompliant
    }
}
```





#### Compliant Solution

```
public class Point
{
    private readonly int x;
    public MyClass(int x)
    {
        this.x = x;
    }
    public override int GetHashCode()
    {
        return x.GetHashCode();
    }
}
```

#### Exceptions

This rule doesn't report on guard conditions checking for reference equality.

```
public class Point
{
    public override bool Equals(object obj)
    {
        if (base.Equals(obj)) // Compliant, although it could be
        {
            return true;
        }
    }
}
```

<b>Creating cookies without the "secure" flag is security-sensitive</b>  Security Hotspot
<b>Literal suffixes should be upper case</b>  Code Smell
<b>Null checks should not be used with "is"</b>  Code Smell
<b>Method overloads should be grouped together</b>  Code Smell
<b>"params" should be used instead of</b>

```
...
}
}
```

Available In:  
**sonarlint**  | **sonarcloud**  | **sonarqube** 