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...

Bug

Locks should be released

Bug

Using publicly writable directories is security-sensitive

Security Hotspot

Using clear-text protocols is security-sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot

Configuring loggers is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot

Disabling CSRF protections is security-sensitive

Security Hotspot

Using non-standard cryptographic algorithms is security-sensitive

Security Hotspot

Using pseudorandom number generators (PRNGs) is security-sensitive

Security Hotspot

Parameter names should match base declaration and other partial definitions

Code Smell

"ValueTask" should be consumed correctly

Code Smell

Methods named "Dispose" should implement "IDisposable.Dispose"

Analyze your code

Code Smell Blocker pitfall

`Dispose` as a method name should be used exclusively to implement `IDisposable.Dispose` to prevent any confusion.

It may be tempting to create a `Dispose` method for other purposes, but doing so will result in confusion and likely lead to problems in production.

Noncompliant Code Example

```
public class GarbageDisposal
{
  private int Dispose()  // Noncompliant
  {
    // ...
  }
}
```

Compliant Solution

```
public class GarbageDisposal : IDisposable
{
  public void Dispose()
  {
    // ...
  }
}
```

or

```
public class GarbageDisposal
{
  private int Grind()
  {
    // ...
  }
}
```

Exceptions

Methods named `Dispose` and invoked from the `IDisposable.Dispose` implementation are not reported.

```
public class GarbageDisposal  :  IDisposable
{
  protected virtual void Dispose(bool disposing)
  {
    //...
  }
```

String offset-based methods should be preferred for finding substrings from offsets

Code Smell

"default" clauses should be first or last

Code Smell

Unread "private" fields should be removed

Code Smell

Base class methods should not be hidden

```
  public void Dispose()
  {
    Dispose(true);
    GC.SuppressFinalize(this);
  }
}
```

Available In:

sonarlint | sonarcloud | sonarqube