

- 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... 🔍

Testcases should contain tests

Code Smell

Short-circuit logic should be used in boolean contexts

Code Smell

JWT should be signed and verified with strong cipher algorithms

Vulnerability

Cipher algorithms should be robust

Vulnerability

Encryption algorithms should be used with secure mode and padding scheme

Vulnerability

Insecure temporary file creation methods should not be used

Vulnerability

Server certificates should be verified during SSL/TLS connections

Vulnerability

LDAP connections should be authenticated

Vulnerability

Cryptographic keys should be robust

Vulnerability

Weak SSL/TLS protocols should not be used

Vulnerability

Cipher Block Chaining IVs should be unpredictable

Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

Vulnerability

## SQL keywords should be delimited by whitespace

Analyze your code

Bug Blocker ? sql

Badly formed SQL is likely to cause errors at runtime.

This rule raises an issue when the spacing around SQL keywords appears to be missing.

### Noncompliant Code Example





```
string select = "SELECT p.fname, p.lname, p.street1, p.stree
                "FROM person p" + // Noncompliant; concatenates to:
                "WHERE p.id = @ID"; // Noncompliant; concatenates t
```

### Compliant Solution

```
string select = "SELECT p.fname, p.lname, p.street1, p.stree
                " FROM person p" +
                " WHERE p.id = @ID";
```

Available In:

sonarlint | sonarcloud | sonarqube

<p>Hashes should include an unpredictable salt</p> <p> Vulnerability</p>
<p>Non-async "Task/Task&lt;T&gt;" methods should not return null</p> <p> Bug</p>
<p>Calls to delegate's method "BeginInvoke" should be paired with calls to "EndInvoke"</p> <p> Bug</p>
<p>"Shared" parts should not be created with "new"</p> <p> Bug</p>