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...

Code Smell

A conditionally executed single line should be denoted by indentation
Code Smell

Conditionals should start on new lines
Code Smell

Assemblies should have version information
Code Smell

Exception types should be "public"
Code Smell

Cognitive Complexity of methods should not be too high
Code Smell

"params" should not be introduced on overrides
Code Smell

"[DefaultValue]" should not be used when "[DefaultParameterValue]" is meant
Code Smell

"[Optional]" should not be used on "ref" or "out" parameters
Code Smell

Non-flags enums should not be used in bitwise operations
Code Smell

Inner class members should not shadow outer class "static" or type members
Code Smell

"Explicit" conversions of "foreach" loops should not be used
Code Smell

Encryption algorithms should be used with secure mode and padding scheme

Analyze your code

Vulnerability | Critical | cwe privacy owasp sans-top25

Encryption operation mode and the padding scheme should be chosen appropriately to guarantee data confidentiality, integrity and authenticity:

- For block cipher encryption algorithms (like AES):
  - The GCM (Galois Counter Mode) mode which works internally with zero/no padding scheme, is recommended, as it is designed to provide both data authenticity (integrity) and confidentiality. Other similar modes are CCM, CWC, EAX, IAPM and OCB.
  - The CBC (Cipher Block Chaining) mode by itself provides only data confidentiality, it's recommended to use it along with Message Authentication Code or similar to achieve data authenticity (integrity) too and thus to prevent padding oracle attacks.
  - The ECB (Electronic Codebook) mode doesn't provide serious message confidentiality: under a given key any given plaintext block always gets encrypted to the same ciphertext block. This mode should not be used.
- For RSA encryption algorithm, the recommended padding scheme is OAEP.

Noncompliant Code Example

AesManaged object with insecure mode:

```
AesManaged aes4 = new AesManaged
{
  KeySize = 128,
  BlockSize = 128,
  Mode = CipherMode.ECB, // Noncompliant
  Padding = PaddingMode.PKCS7
};
```

RSACryptoServiceProvider object without OAEP padding:

```
RSACryptoServiceProvider RSA1 = new RSACryptoServiceProvider
encryptedData = RSA1.Encrypt(dataToEncrypt, false); // Nonco
```

Compliant Solution

AES with GCM mode with bouncycastle library:

```
GcmBlockCipher blockCipher = new GcmBlockCipher(new AesEngin
blockCipher.Init(true, new AeadParameters(new KeyParameter(s
```

AES with GCM mode with AesGcm object:

```
var aesGcm = new AesGcm(key); // Compliant
```

RSA with OAEP padding with RSACryptoServiceProvider object:

Instance members should not write to "static" fields

Code Smell

"IndexOf" checks should not be for positive numbers

Code Smell

Whitespace and control characters in string literals should be explicit

Code Smell

Properties should not make collection or array copies

Code Smell

```
RSACryptoServiceProvider RSA2 = new RSACryptoServiceProvider
encryptedData = RSA2.Encrypt(dataToEncrypt, true); // Compli
```

See

- OWASP Top 10 2021 Category A2 - Cryptographic Failures
- OWASP Top 10 2017 Category A6 - Security Misconfiguration
- MITRE, CWE-327 - Use of a Broken or Risky Cryptographic Algorithm
- SANS Top 25 - Porous Defenses

Available In:

sonarlint | sonarcloud | sonarqube