GRAMMATECH

# CWE WEAKNESS IDS

## MAPPED TO CODESONAR® 6.2 C# WARNING CLASSES

TRUSTED LEADERS OF SOFTWARE ASSURANCE AND ADVANCED CYBER-SECURITY SOLUTIONS

WWW.GRAMMATECH.COM

## INTRODUCTION

The Common Weakness Enumeration (CWE™) is a list of software weakness types. Creating the list is a community initiative aimed at creating specific and succinct definitions for each common weakness type.

Every CodeSonar 6.2 warning report includes the numbers of any CWE weakness IDs that are closely mapped to the warning's class. (The close mapping for a warning class is the set of categories—including CWE weakness IDs—that most closely match the class, if any).

You can configure CodeSonar to enable and disable warning classes mapped to specific CWE weakness IDs, or use build presets to enable all warning classes that are closely mapped to any CWE weakness IDs. In addition, you can use the CodeSonar search function to find warnings related to specific CWE weakness IDs.

CodeSonar 6.2 is using CWE v4.4 (released March 15, 2021).

For more information on Common Weakness Enumeration:

https://cwe.mitre.org/data/index.html

The remainder of this document comprises two tables:

- A table showing the close mapping between CodeSonar C and C++ warning classes and CWEweakness IDs.
- A table showing the broad mapping between CodeSonar C and C++ warning classes and CWE weakness IDs. The broad CWE mapping for a CodeSonar warning class combines CWEweakness IDs from four sources:
    1. The close CWE mapping for the class.
    2. Other CWE weakness IDs that are related to the class in a meaningful way, but noteligible for the close mapping.
    3. For all CWE weakness IDs from sources 1 and 2, all *ancestors* in the CWE hierarchy.
    4. For all CWE weakness IDs from sources 1 and 2, all *descendants* in the CWE hierarchy.

A separate document, CWE Weakness IDs Mapped to CodeSonar® Java Warning Classes, lists the CodeSonar Java warning classes that are closely and broadly mapped to CWE weakness IDs.

GrammaTech is a leading global provider of application testing (AST) solutions used by the world'smost security conscious organizations to detect, measure, analyze and resolve vulnerabilitiesfor software they develop or use. The company is also a trusted cybersecurity and artificial intelligence research partner for the nation's civil, defense, and intelligence agencies.

## CWE CLOSE MAPPING: C# (CODESONAR V6.2)

The following table lists the CodeSonar C# warning classes that are closely mapped to CWE weakness IDs.

| Category ID | Category Name | CodeSonar Class Name |
|---|---|---|
| CWE:20 | Improper Input Validation | Disabled Input Validation (C#) |
| CWE:22 | Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') | Tainted Path (C#) |
| CWE:74 | Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection') | DLL Injection (C#)<br>DOS Injection (C#)<br>Tainted Control (C#)<br>Tainted Hardware Device Property (C#)<br>Tainted Network Address (C#)<br>Tainted Resource (C#)<br>Tainted URL (C#)<br>Tainted XAML (C#)<br>Tainted XML (C#) |
| CWE:78 | Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection') | Command Injection (C#) |
| CWE:79 | Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') | Cross Site Scripting (C#) |
| CWE:89 | Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection') | SQL Injection (C#) |
| CWE:90 | Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection') | Tainted LDAP Attribute (C#)<br>Tainted LDAP Filter (C#) |
| CWE:94 | Improper Control of Generation of Code ('Code Injection') | Code Injection (C#) |
| CWE:95 | Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection') | Tainted Expression Evaluation (C#) |
| CWE:113 | Improper Neutralization of CRLF Sequences in HTTP Headers ('HTTP Response Splitting') | Tainted HTTP Response (C#) |
| CWE:114 | Process Control | DLL Injection (C#) |
| CWE:117 | Improper Output Neutralization for Logs | Tainted Log (C#) |
| CWE:190 | Integer Overflow or Wraparound | Cast: int Computation to long (C#) |
| CWE:192 | Integer Coercion Error | Cast: Integer to Floating Point (C#) |
| CWE:197 | Numeric Truncation Error | Approximate e Constant (C#)<br>Approximate pi Constant (C#) |
| CWE:227 | 7PK - API Abuse | Ambiguous Call from Inner Class (C#)<br>Ignored Return Value for Pure Function (C#)<br>Inappropriate Instanceof (C#)<br>Method Should Not Return null (C#)<br>Non-Object compareTo Parameter (C#)<br>Redundant Call for Integral Argument (C#)<br>Shadowed Identifier (C#)<br>compareTo in Non-Comparable Class (C#)<br>equals Parameter Should Be Object (C#) |
| CWE:252 | Unchecked Return Value | Call Might Return Null (C#)<br>Ignored Return Value (C#) |
| CWE:259 | Use of Hard-coded Password | Hardcoded Password (C#) |
| CWE:287 | Improper Authentication | Anonymous LDAP Authentication (C#)<br>Hostname in Condition (C#)<br>Missing Authentication Annotation (C#) |
| CWE:319 | Cleartext Transmission of Sensitive Information | Tainted Message (C#) |
| CWE:326 | Inadequate Encryption Strength | Insecure Key Derivation (C#) |

| | | |
|---|---|---|
| CWE:327 | Use of a Broken or Risky Cryptographic Algorithm | Cryptographic Algorithm with Risky Default Cipher (C#)<br>Cryptographic Algorithm with Weak Cipher (C#)<br>Cryptographic Algorithm with Weak Hash (C#)<br>Deprecated Cryptography Provider (C#)<br>Risky Cipher Algorithm (C#)<br>Risky Cipher Field (C#)<br>Risky Cryptographic Algorithm (C#)<br>Risky Cryptographic Field (C#)<br>Unsafe Base64 Encoding (C#) |
| CWE:328 | Reversible One-Way Hash | Weak Hash Algorithm (C#)<br>Weak Hash Algorithm Field (C#) |
| CWE:330 | Use of Insufficiently Random Values | Hardcoded Random Seed (C#)<br>Insecure Random Number Generator (C#) |
| CWE:338 | Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG) | Weak Cryptographic Value (C#) |
| CWE:390 | Detection of Error Condition Without Action | Empty Exception Handler (C#) |
| CWE:395 | Use of NullPointerException Catch to Detect NULL Pointer Dereference | Inappropriate Exception Handler (C#) |
| CWE:396 | Declaration of Catch for Generic Exception | Generic Exception Handler (C#) |
| CWE:397 | Declaration of Throws for Generic Exception | Broad Throws Clause (C#) |
| CWE:398 | 7PK - Code Quality | Empty Branch Statement (C#)<br>Redundant Call for String Argument (C#)<br>Static Field Assigned Non-Static (C#)<br>Unused Class (C#)<br>Unused Field (C#)<br>Useless Class Cast (C#)<br>Useless null Test (C#)<br>Useless null Test of Field (C#)<br>Useless null Test of Parameter (C#) |
| CWE:400 | Uncontrolled Resource Consumption | Closeable Not Stored (C#) |
| CWE:412 | Unrestricted Externally Accessible Lock | Synchronization on Interned String (C#) |
| CWE:413 | Improper Resource Locking | Impossible Client Side Locking (C#)<br>Synchronization on static (C#) |
| CWE:440 | Expected Behavior Violation | toString on Array (C#) |
| CWE:456 | Missing Initialization of a Variable | Field Never Written (C#)<br>Lambda Parameter may be null (C#)<br>Lambda Parameter may be null (Java)<br>Null Pointer Dereference (C#) |
| CWE:470 | Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') | Reflection Injection (C#) |
| CWE:476 | NULL Pointer Dereference | Actual Parameter Element may be null (C#)<br>Field Element may be null (deep) (C#)<br>Field may be null (deep) (C#)<br>Null Parameter Dereference (C#)<br>Null Pointer Dereference (C#)<br>Null Pointer Dereference (deep) (C#)<br>Return Value may Contain null Element (C#)<br>Return Value may be null (C#)<br>Return null Array (C#)<br>Return null Boolean (C#)<br>Return null Optional (C#)<br>Unchecked Parameter Dereference (C#)<br>Unchecked Parameter Dereference (deep) (C#)<br>Unchecked Parameter Element Dereference (deep) (C#)<br>null Passed to Method (deep) (C#) |

| | | |
|---|---|---|
| CWE:480 | Use of Incorrect Operator | Bitwise AND on Boolean Constant (C#)<br>Bitwise OR on Boolean Constant (C#)<br>Inefficient Bitwise AND (C#)<br>Inefficient Bitwise OR (C#)<br>Should Use == Instead of equals() (C#) |
| CWE:481 | Assigning instead of Comparing | Assignment in Conditional (C#) |
| CWE:485 | 7PK - Encapsulation | Field Too Visible (C#)<br>Method Should be final (C#)<br>Method Should be private (C#)<br>Shadowed Identifier (C#)<br>Static Field Too Visible (C#) |
| CWE:489 | Active Debug Code | Class Enables Debug Features (C#)<br>Debug Call (C#)<br>Method Enables Debug Features (C#) |
| CWE:491 | Public cloneable() Method Without Final ('Object Hijack') | clone Non-cloneable (C#)<br>clone Subclass of Non-clonable (C#)<br>clone not final (C#) |
| CWE:501 | Trust Boundary Violation | Tainted Bundle (C#)<br>Tainted Session (C#) |
| CWE:502 | Deserialization of Untrusted Data | Serialization Not Disabled (C#) |
| CWE:522 | Insufficiently Protected Credentials | Password in Property File (C#) |
| CWE:547 | "Use of Hard-coded, Security-relevant Constants" | Hardcoded Filename (C#)<br>Hardcoded IP Address (C#) |
| CWE:550 | Server-generated Error Message Containing Sensitive Information | Exception Information Disclosure (C#) |
| CWE:561 | Dead Code | Unreachable Instruction (C#)<br>Unused Class (C#)<br>Unused Method (C#) |
| CWE:563 | Assignment to Variable without Use | Unnecessary Field (C#)<br>Unused Value: Actual Parameter (C#)<br>Unused Value: Variable (C#)<br>Unused Value: Write to Parameter (C#) |
| CWE:567 | Unsynchronized Access to Shared Data in a Multithreaded Context | Missing synchronized Statement (C#)<br>Unguarded Field (C#)<br>Unguarded Parameter (C#)<br>Useless volatile Modifier (C#) |
| CWE:570 | Expression is Always False | Impossible reference comparison (Java)<br>Instanceof Always False (C#)<br>Redundant Condition (C#)<br>equals Always Fails (C#) |
| CWE:571 | Expression is Always True | Instanceof Always True (C#) |
| CWE:572 | Call to Thread run() instead of start() | Synchronous Call to Thread Body (C#) |
| CWE:573 | Improper Following of Specification by Caller | Missing Call to super (C#) |
| CWE:581 | Object Model Violation: Just One of Equals and Hashcode Defined | Defines equals but not hashCode (C#)<br>Defines hashCode but not equals (C#) |
| CWE:585 | Empty Synchronized Block | Useless Synchronization (C#) |
| CWE:595 | Comparison of Object References Instead of Object Contents | equals on Array (C#) |
| CWE:601 | URL Redirection to Untrusted Site ('Open Redirect') | Tainted URL (C#) |
| CWE:607 | Public Static Final Field References Mutable Object | Mutable Constant Field (C#)<br>Mutable Enumeration (C#) |
| CWE:609 | Double-Checked Locking | Double-Checked Locking (C#) |
| CWE:611 | Improper Restriction of XML External Entity Reference | Insecure XSLT Execution (C#)<br>Possible XML External Entity Reference (C#) |
| CWE:614 | Sensitive Cookie in HTTPS Session Without 'Secure' Attribute | Insecure Cookie (C#) |
| CWE:624 | Executable Regular Expression Error | Tainted Regular Expression (C#) |
| CWE:628 | Function Call with Incorrectly Specified Arguments | Method Names Differ Only in Case (C#) |
| CWE:643 | Improper Neutralization of Data within XPath Expressions ('XPath Injection') | Tainted Xpath (C#) |
| CWE:662 | Improper Synchronization | Useless volatile Modifier (C#) |

| | | |
|---|---|---|
| CWE:665 | Improper Initialization | Useless Assignment (C#)<br>Useless Assignment to Default (C#) |
| CWE:674 | Uncontrolled Recursion | Potential Infinite Recursion (C#) |
| CWE:676 | Use of Potentially Dangerous Function | Method Disables Security Setting (C#) |
| CWE:682 | Incorrect Calculation | Abs on random (C#) |
| CWE:686 | Function Call With Incorrect Argument Type | Non-overriding Method Signature (C#) |
| CWE:697 | Incorrect Comparison | Asymmetric compareTo (C#)<br>compareTo without equals (C#)<br>compareTo/equals mismatch (C#) |
| CWE:704 | Incorrect Type Conversion or Cast | Risky Class Cast (C#)<br>Risky array store (C#) |
| CWE:710 | Improper Adherence to Coding Standards | Naming Style Violation (C#) |
| CWE:749 | Exposed Dangerous Method or Function | Security Annotation Conflict (C#) |
| CWE:757 | Selection of Less-Secure Algorithm During Negotiation ('Algorithm Downgrade') | Deprecated Transfer Protocol (C#) |
| CWE:768 | Incorrect Short Circuit Evaluation | Bitwise AND on Boolean (C#)<br>Bitwise OR on Boolean (C#) |
| CWE:820 | Missing Synchronization | Unguarded Method (C#) |
| CWE:833 | Deadlock | Blocking in Critical Section (C#) |
| CWE:909 | Missing Initialization of Resource | Empty jar File Archived (C#)<br>Empty zip File Archived (C#) |
| CWE:913 | Improper Control of Dynamically-Managed Code Resources | Deserializable Class (C#)<br>Deserializing Non-Serializable Class (C#)<br>Missing Serial Version Field (C#)<br>Nonserializable Field (C#)<br>Nonserializable Field Element (C#)<br>Nonserializable Outer Class (C#)<br>Unexpected Serial Version Field (C#) |
| CWE:915 | Improperly Controlled Modification of Dynamically-Determined Object Attributes | Reflection Modifies Member Accessibility (C#) |
| CWE:916 | Use of Password Hash With Insufficient Computational Effort | Inadequate Salt (C#) |
| CWE:922 | Insecure Storage of Sensitive Information | Certificate Added to Root Store (C#) |
| CWE:1023 | Incomplete Comparison with Missing Factors | Missing Equals Override (C#) |
| CWE:1024 | Comparison of Incompatible Type | == Always Fails Because Types Always Different (C#) |
| CWE:1077 | Floating Point Comparison with Incorrect Operator | Floating Point Equality (C#) |
| CWE:1126 | Declaration of Variable with Unnecessarily Wide Scope | Unnecessary Field (C#) |
| CWE:1164 | Irrelevant Code | Unused Object (C#) |
| | | Field Never Read (C#) |
| CWE:1176 | Inefficient CPU Computation | Single-use Random Number Generator (C#) |