# CWE Detail – CWE-391

## Description

[PLANNED FOR DEPRECATION. SEE MAINTENANCE NOTES AND CONSIDER CWE-252, CWE-248, OR CWE-1069.] Ignoring exceptions and other error conditions may allow an attacker to induce unexpected behavior unnoticed.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

## Common Consequences

**•** Impact: Varies by Context, Unexpected State, Alter Execution Logic — Notes:

## Potential Mitigations

**•** Requirements: The choice between a language which has named or unnamed exceptions needs to be done. While unnamed exceptions exacerbate the chance of not properly dealing with an exception, named exceptions suffer from the up call version of the weak base class problem. (Effectiveness: N/A)

**•** Requirements: A language can be used which requires, at compile time, to catch all serious exceptions. However, one must make sure to use the most current version of the API as new exceptions could be added. (Effectiveness: N/A)

**•** Implementation: Catch all relevant exceptions. This is the recommended solution. Ensure that all exceptions are handled in such a way that you can be sure of the state of your system at any given moment. (Effectiveness: N/A)

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** If a RareException were to ever be thrown, the program would continue to execute as though nothing unusual had occurred. The program records no evidence indicating the special situation, potentially frustrating any later attempt to explain the program's behavior.

## Notes

**•** Maintenance: This entry is slated for deprecation; it has multiple widespread interpretations by CWE analysts. It currently combines information from three different taxonomies, but each taxonomy is talking about a slightly different issue. CWE analysts might map to this entry based on any of these issues. 7PK has "Empty Catch Block" which has an association with empty exception block (CWE-1069); in this case, the exception has performed the check, but does not handle. In PLOVER there is "Unchecked Return Value" which is CWE-252, but unlike "Empty Catch Block" there isn't even a check of the issue - and "Unchecked Error Condition" implies lack of a check. For CLASP, "Uncaught Exception" (CWE-248) is associated with incorrect error propagation - uncovered in CWE 3.2 and earlier, at least. There are other issues related to error handling and checks.

**•** Other: When a programmer ignores an exception, they implicitly state that they are operating under one of two assumptions: This method call can never fail. It doesn't matter if this call fails.