# CWE Detail – CWE-395

## Description

Catching NullPointerException should not be used as an alternative to programmatic checks to prevent dereferencing a null pointer.

## Extended Description

Programmers typically catch NullPointerException under three circumstances: The program contains a null pointer dereference. Catching the resulting exception was easier than fixing the underlying problem. The program explicitly throws a NullPointerException to signal an error condition. The code is part of a test harness that supplies unexpected input to the classes under test. Of these three circumstances, only the last is acceptable.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: DoS: Resource Consumption (CPU) — Notes:

## Potential Mitigations

**•** Architecture and Design: Do not extensively rely on catching exceptions (especially for validating user input) to handle errors. Handling exceptions can decrease the performance of an application. (Effectiveness: N/A)

## Applicable Platforms

**•** Java (Class: None, Prevalence: Undetermined)

## Demonstrative Examples

**•** N/A