# CWE Detail – CWE-767

## Description

The product defines a public method that reads or modifies a private variable.

## Extended Description

If an attacker modifies the variable to contain unexpected values, this could violate assumptions from other parts of the code. Additionally, if an attacker can read the private variable, it may expose sensitive information or make it easier to launch further attacks.

## Threat-Mapped Scoring

Score: 3.0

Priority: P2 - Serious (High)

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Application Data, Other — Notes:

## Potential Mitigations

**•** Implementation: Use class accessor and mutator methods appropriately. Perform validation when accepting data from a public method that is intended to modify a critical private variable. Also be sure that appropriate access controls are being applied when a public method interfaces with critical data. (Effectiveness: N/A)

## Applicable Platforms

**•** C++ (Class: None, Prevalence: Undetermined)

**•** C# (Class: None, Prevalence: Undetermined)

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

## Demonstrative Examples

**•** N/A

**•** The programmer implemented setPID with the intention of modifying the PID variable, but due to a typo. accidentally specified the critical variable UID instead. If the program allows profile IDs to be between 1 and 10, but a UID of 1 means the user is treated as an admin, then a user could gain administrative privileges as a result of this typo.

## Notes

**•** Maintenance: This entry is closely associated with access control for public methods. If the public methods are restricted with proper access controls, then the information in the private variable will not be exposed to unexpected parties. There may be chaining or composite relationships between improper access controls and this weakness.