# CWE Detail – CWE-766

## Description

The product declares a critical variable, field, or member to be public when intended security policy requires it to be private.

## Extended Description

This issue makes it more difficult to maintain the product, which indirectly affects security by making it more difficult or time-consuming to find and/or fix vulnerabilities. It also might make it easier to introduce vulnerabilities.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2010-3860: variables declared public allow remote read of system properties such as user name and home directory.

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Application Data, Modify Application Data — Notes: Making a critical variable public allows anyone with access to the object in which the variable is contained to alter or read the value.

**•** Impact: Reduce Maintainability — Notes:

## Potential Mitigations

**•** Implementation: Data should be private, static, and final whenever possible. This will assure that your code is protected by instantiating early, preventing access, and preventing tampering. (Effectiveness: N/A)

## Applicable Platforms

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

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

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

## Demonstrative Examples

**•** Instead, the critical data should be declared private.

**•** However, the member variables username and password are declared public and therefore will allow access and changes to the member variables to anyone with access to the object. These member variables should be declared private as shown below to prevent unauthorized access and changes.