# CWE Detail – CWE-566

## Description

The product uses a database table that includes records that should not be accessible to an actor, but it executes a SQL statement with a primary key that can be controlled by that actor.

## Extended Description

When a user can set a primary key to any value, then the user can modify the key to point to unauthorized records. Database access control errors occur when: Data enters a program from an untrusted source. The data is used to specify the value of a primary key in a SQL query. The untrusted source does not have the permissions to be able to access all rows in the associated table.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Architecture and Design: COMMISSION: This weakness refers to an incorrect design related to an architectural security tactic.

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Application Data, Modify Application Data, Bypass Protection Mechanism — Notes:

## Potential Mitigations

**•** Implementation: Assume all input is malicious. Use a standard input validation mechanism to validate all input for length, type, syntax, and business rules before accepting the data. Use an "accept known good" validation strategy. (Effectiveness: N/A)

**•** Implementation: Use a parameterized query AND make sure that the accepted values conform to the business rules. Construct your SQL statement accordingly. (Effectiveness: N/A)

## Applicable Platforms

**•** SQL (Class: None, Prevalence: Often)

## Demonstrative Examples

**•** The problem is that the developer has not considered all of the possible values of id. Although the interface generates a list of invoice identifiers that belong to the current user, an attacker can bypass this interface to request any desired invoice. Because the code in this example does not check to ensure that the user has permission to access the requested invoice, it will display any invoice, even if it does not belong to the current user.