# CWE Detail – CWE-597

## Description

The product uses the wrong operator when comparing a string, such as using "==" when the .equals() method should be used instead.

## Extended Description

In Java, using == or != to compare two strings for equality actually compares two objects for equality rather than their string values for equality. Chances are good that the two references will never be equal. While this weakness often only affects program correctness, if the equality is used for a security decision, the unintended comparison result could be leveraged to affect program security.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Other — Notes:

## Potential Mitigations

**•** Implementation: Within Java, use .equals() to compare string values. Within JavaScript, use == to compare string values. Within PHP, use == to compare a numeric value to a string value. (PHP converts the string to a number.) (Effectiveness: High)

## Demonstrative Examples

**•** However, the if statement will not be executed as the strings are compared using the "==" operator. For Java objects, such as String objects, the "==" operator compares object references, not object values. While the two String objects above contain the same string values, they refer to different object references, so the System.out.println statement will not be executed. To compare object values, the previous code could be modified to use the equals method:

**•** However, the body of the if statement will not be executed, as the "===" compares both the type of the variable AND the value. As the types of the first comparison are number and string, it fails. The types in the second are int and reference, so this one fails as well. The types in the third are reference and string, so it also fails. While the variables above contain the same values, they are contained in different types, so the document.getElementById... statement will not be executed in any of the cases. To compare object values, the previous code is modified and shown below to use the "==" for value comparison so the comparison in this example executes the HTML statement:

**•** However, the body of the if statement will not be executed, as the "===" compares both the type of the variable AND the value. As the types of the first comparison are number and string, it fails. While the variables above contain the same values, they are contained in different types, so the TRUE portion of the if statement will not be executed. To compare object values, the previous code is modified and shown below to use the "==" for value comparison (string converted to number) so the comparison in this example executes the TRUE statement: