# CWE Detail – CWE-478

## Description

The code does not have a default case in an expression with multiple conditions, such as a switch statement.

## Extended Description

If a multiple-condition expression (such as a switch in C) omits the default case but does not consider or handle all possible values that could occur, then this might lead to complex logical errors and resultant weaknesses. Because of this, further decisions are made based on poor information, and cascading failure results. This cascading failure may result in any number of security issues, and constitutes a significant failure in the system.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Varies by Context, Alter Execution Logic — Notes: Depending on the logical circumstances involved, any consequences may result: e.g., issues of confidentiality, authentication, authorization, availability, integrity, accountability, or non-repudiation.

## Potential Mitigations

**•** Implementation: Ensure that there are no cases unaccounted for when adjusting program flow or values based on the value of a given variable. In the case of switch style statements, the very simple act of creating a default case can, if done correctly, mitigate this situation. Often however, the default case is used simply to represent an assumed option, as opposed to working as a check for invalid input. This is poor practice and in some cases is as bad as omitting a default case entirely. (Effectiveness: N/A)

## Applicable Platforms

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

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

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

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

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

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

## Demonstrative Examples

**•** Instead a default label should be used for unaccounted conditions:

**•** However, this code assumes that the value of the points input parameter will always be 0, 1 or 2 and does not check for other incorrect values passed to the method. This can be easily accomplished by providing a default label in the switch statement that outputs an error message indicating an invalid value for the points input parameter and returning a null value.

**•** The recommended approach is to add a default case that captures any unexpected result conditions, regardless of how improbable these unexpected conditions might be, and properly handles them.

**•** The recommended approach is to add a default case that captures any unexpected result conditions and properly handles them.

**•** The case statement does not include a default to handle the scenario when the user provides inputs of 3'h6 and 3'h7. Those inputs push the system to an undefined state and might cause a crash (denial of service) or any other unanticipated outcome. Adding a default statement to handle undefined inputs mitigates this issue. This is shown in the "Good" code snippet below. The default statement is in bold.