# CWE Detail – CWE-785

## Description

The product invokes a function for normalizing paths or file names, but it provides an output buffer that is smaller than the maximum possible size, such as PATH\_MAX.

## Extended Description

Passing an inadequately-sized output buffer to a path manipulation function can result in a buffer overflow. Such functions include realpath(), readlink(), PathAppend(), and others.

## Threat-Mapped Scoring

Score: 1.5

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Memory, Execute Unauthorized Code or Commands, DoS: Crash, Exit, or Restart — Notes:

## Potential Mitigations

**•** Implementation: Always specify output buffers large enough to handle the maximum-size possible result from path manipulation functions. (Effectiveness: N/A)

## Applicable Platforms

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

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

## Demonstrative Examples

**•** For most values of the current directory and the name parameter, this function will work properly. However, if the name parameter is particularly long, then the second call to PathAppend() could overflow the outputDirectoryName buffer, which is smaller than MAX\_PATH bytes.

## Notes

**•** Maintenance: This entry is at a much lower level of abstraction than most entries because it is function-specific. It also has significant overlap with other entries that can vary depending on the perspective. For example, incorrect usage could trigger either a stack-based overflow (CWE-121) or a heap-based overflow (CWE-122). The CWE team has not decided how to handle such entries.