# CWE Detail – CWE-124

## Description

The product writes to a buffer using an index or pointer that references a memory location prior to the beginning of the buffer.

## Extended Description

This typically occurs when a pointer or its index is decremented to a position before the buffer, when pointer arithmetic results in a position before the beginning of the valid memory location, or when a negative index is used.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2021-24018: buffer underwrite in firmware verification routine allows code execution via a crafted firmware image

**•** CVE-2002-2227: Unchecked length of SSLv2 challenge value leads to buffer underflow.

**•** CVE-2007-4580: Buffer underflow from a small size value with a large buffer (length parameter inconsistency, CWE-130)

**•** CVE-2007-1584: Buffer underflow from an all-whitespace string, which causes a counter to be decremented before the buffer while looking for a non-whitespace character.

**•** CVE-2007-0886: Buffer underflow resultant from encoded data that triggers an integer overflow.

**•** CVE-2006-6171: Product sets an incorrect buffer size limit, leading to "off-by-two" buffer underflow.

**•** CVE-2006-4024: Negative value is used in a memcpy() operation, leading to buffer underflow.

**•** CVE-2004-2620: Buffer underflow due to mishandled special characters

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Memory, DoS: Crash, Exit, or Restart — Notes: Out of bounds memory access will very likely result in the corruption of relevant memory, and perhaps instructions, possibly leading to a crash.

**•** Impact: Execute Unauthorized Code or Commands, Modify Memory, Bypass Protection Mechanism, Other — Notes: If the corrupted memory can be effectively controlled, it may be possible to execute arbitrary code. If the corrupted memory is data rather than instructions, the system will continue to function with improper changes, possibly in violation of an implicit or explicit policy. The consequences would only be limited by how the affected data is used, such as an adjacent memory location that is used to specify whether the user has special privileges.

**•** Impact: Bypass Protection Mechanism, Other — Notes: When the consequence is arbitrary code execution, this can often be used to subvert any other security service.

## Potential Mitigations

**•** Requirements: Choose a language that is not susceptible to these issues. (Effectiveness: N/A)

**•** Implementation: All calculated values that are used as index or for pointer arithmetic should be validated to ensure that they are within an expected range. (Effectiveness: N/A)

## Applicable Platforms

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

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

## Demonstrative Examples

**•** However, this function can cause a buffer underwrite if the input character string contains all whitespace. On some systems the while statement will move backwards past the beginning of a character string and will call the isspace() function on an address outside of the bounds of the local buffer.

**•** In the case where the substring is not found in destBuf, strstr() will return NULL, causing the pointer arithmetic to be undefined, potentially setting the value of idx to a negative number. If idx is negative, this will result in a buffer underwrite of destBuf.

## Notes

**•** Relationship: This could be resultant from several errors, including a bad offset or an array index that decrements before the beginning of the buffer (see CWE-129).