# CWE Detail – CWE-126

## Description

The product reads from a buffer using buffer access mechanisms such as indexes or pointers that reference memory locations after the targeted buffer.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2022-1733: Text editor has out-of-bounds read past end of line while indenting C code

**•** CVE-2014-0160: Chain: "Heartbleed" bug receives an inconsistent length parameter (CWE-130) enabling an out-of-bounds read (CWE-126), returning memory that could include private cryptographic keys and other sensitive data. (KEV)

**•** CVE-2009-2523: Chain: product does not handle when an input string is not NULL terminated, leading to buffer over-read or heap-based buffer overflow.

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Memory — Notes:

**•** Impact: Bypass Protection Mechanism — Notes: By reading out-of-bounds memory, an attacker might be able to get secret values, such as memory addresses, which can be bypass protection mechanisms such as ASLR in order to improve the reliability and likelihood of exploiting a separate weakness to achieve code execution instead of just denial of service.

**•** Impact: DoS: Crash, Exit, or Restart — Notes: An attacker might be able to cause a crash or other denial of service by causing the product to read a memory location that is not allowed (such as a segmentation fault), or to cause other conditions in which the read operation returns more data than is expected.

## Applicable Platforms

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

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

## Demonstrative Examples

**•** However, the message length variable from the structure is used as the condition for ending the for loop without validating that the message length variable accurately reflects the length of the message body (CWE-606). This can result in a buffer over-read (CWE-125) by reading from memory beyond the bounds of the buffer if the message length variable indicates a length that is longer than the size of a message body (CWE-130).

**•** However, the code do not take into account that strncpy() will not add a NULL terminator when the source buffer is equal in length of longer than that provide size attribute. Therefore if a user enters a filename or pattern that are the same size as (or larger than) their respective character arrays, a NULL terminator will not be added (CWE-170) which leads to the printf() read beyond the expected end of the Filename and Pattern buffers.

## Notes

**•** Relationship: These problems may be resultant from missing sentinel values (CWE-463) or trusting a user-influenced input length variable.

**•** Other: A buffer over-read typically occurs when the pointer or its index is incremented to a position past the end of the buffer or when pointer arithmetic results in a position after the valid memory location.