# CWE Detail – CWE-835

## Description

The product contains an iteration or loop with an exit condition that cannot be reached, i.e., an infinite loop.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2022-22224: Chain: an operating system does not properly process malformed Open Shortest Path First (OSPF) Type/Length/Value Identifiers (TLV) (CWE-703), which can cause the process to enter an infinite loop (CWE-835)

**•** CVE-2022-25304: A Python machine communication platform did not account for receiving a malformed packet with a null size, causing the receiving function to never update the message buffer and be caught in an infinite loop.

**•** CVE-2011-1027: Chain: off-by-one error (CWE-193) leads to infinite loop (CWE-835) using invalid hex-encoded characters.

**•** CVE-2011-1142: Chain: self-referential values in recursive definitions lead to infinite loop.

**•** CVE-2011-1002: NULL UDP packet is never cleared from a queue, leading to infinite loop.

**•** CVE-2006-6499: Chain: web browser crashes due to infinite loop - "bad
 looping logic [that relies on] floating point math [CWE-1339] to exit
 the loop [CWE-835]"

**•** CVE-2010-4476: Floating point conversion routine cycles back and forth between two different values.

**•** CVE-2010-4645: Floating point conversion routine cycles back and forth between two different values.

**•** CVE-2010-2534: Chain: improperly clearing a pointer in a linked list leads to infinite loop.

**•** CVE-2013-1591: Chain: an integer overflow (CWE-190) in the image size calculation causes an infinite loop (CWE-835) which sequentially allocates buffers without limits (CWE-1325) until the stack is full.

**•** CVE-2008-3688: Chain: A denial of service may be caused by an uninitialized variable (CWE-457) allowing an infinite loop (CWE-835) resulting from a connection to an unresponsive server.

## Common Consequences

**•** Impact: DoS: Resource Consumption (CPU), DoS: Resource Consumption (Memory), DoS: Amplification — Notes: An infinite loop will cause unexpected consumption of resources, such as CPU cycles or memory. The software's operation may slow down, or cause a long time to respond.

## Applicable Platforms

**•** None (Class: Not Language-Specific, Prevalence: Undetermined)

## Demonstrative Examples

**•** However, this will create an infinite loop if the server does not respond. This infinite loop will consume system resources and can be used to create a denial of service attack. To resolve this a counter should be used to limit the number of attempts to establish a connection to the server, as in the following code.

**•** However, the while loop will become an infinite loop if the rateSold input parameter has a value of zero since the inventoryCount will never fall below the minimumCount. In this case the input parameter should be validated to ensure that a value of zero does not cause an infinite loop, as in the following code.