# CWE Detail – CWE-834

## Description

The product performs an iteration or loop without sufficiently limiting the number of times that the loop is executed.

## Extended Description

If the iteration can be influenced by an attacker, this weakness could allow attackers to consume excessive resources such as CPU or memory. In many cases, a loop does not need to be infinite in order to cause enough resource consumption to adversely affect the product or its host system; it depends on the amount of resources consumed per iteration.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

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

**•** 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]"

## Common Consequences

**•** Impact: DoS: Resource Consumption (CPU), DoS: Resource Consumption (Memory), DoS: Amplification, DoS: Crash, Exit, or Restart — Notes: Excessive looping will cause unexpected consumption of resources, such as CPU cycles or memory. The product's operation may slow down, or cause a long time to respond. If limited resources such as memory are consumed for each iteration, the loop may eventually cause a crash or program exit due to exhaustion of resources, such as an out-of-memory error.

## Demonstrative Examples

**•** Note that the only difference between the Good and Bad examples is that the recursion flag will change value and cause the recursive call to return.

**•** 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.