# CWE Detail – CWE-911

## Description

The product uses a reference count to manage a resource, but it does not update or incorrectly updates the reference count.

## Extended Description

Reference counts can be used when tracking how many objects contain a reference to a particular resource, such as in memory management or garbage collection. When the reference count reaches zero, the resource can be de-allocated or reused because there are no more objects that use it. If the reference count accidentally reaches zero, then the resource might be released too soon, even though it is still in use. If all objects no longer use the resource, but the reference count is not zero, then the resource might not ever be released.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2002-0574: chain: reference count is not decremented, leading to memory leak in OS by sending ICMP packets.

**•** CVE-2004-0114: Reference count for shared memory not decremented when a function fails, potentially allowing unprivileged users to read kernel memory.

**•** CVE-2006-3741: chain: improper reference count tracking leads to file descriptor consumption

**•** CVE-2007-1383: chain: integer overflow in reference counter causes the same variable to be destroyed twice.

**•** CVE-2007-1700: Incorrect reference count calculation leads to improper object destruction and code execution.

**•** CVE-2008-2136: chain: incorrect update of reference count leads to memory leak.

**•** CVE-2008-2785: chain/composite: use of incorrect data type for a reference counter allows an overflow of the counter, leading to a free of memory that is still in use.

**•** CVE-2008-5410: Improper reference counting leads to failure of cryptographic operations.

**•** CVE-2009-1709: chain: improper reference counting in a garbage collection routine leads to use-after-free

**•** CVE-2009-3553: chain: reference count not correctly maintained when client disconnects during a large operation, leading to a use-after-free.

**•** CVE-2009-3624: Reference count not always incremented, leading to crash or code execution.

**•** CVE-2010-0176: improper reference counting leads to expired pointer dereference.

**•** CVE-2010-0623: OS kernel increments reference count twice but only decrements once, leading to resource consumption and crash.

**•** CVE-2010-2549: OS kernel driver allows code execution

**•** CVE-2010-4593: improper reference counting leads to exhaustion of IP addresses

**•** CVE-2011-0695: Race condition causes reference counter to be decremented prematurely, leading to the destruction of still-active object and an invalid pointer dereference.

**•** CVE-2012-4787: improper reference counting leads to use-after-free

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: DoS: Resource Consumption (Memory), DoS: Resource Consumption (Other) — Notes: An adversary that can cause a resource counter to become inaccurate may be able to create situations where resources are not accounted for and not released, thus causing resources to become scarce for future needs.

**•** Impact: DoS: Crash, Exit, or Restart — Notes: An adversary that can cause a resource counter to become inaccurate may be able to force an error that causes the product to crash or exit out of its current operation.

## Applicable Platforms

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

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

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