# CWE Detail – CWE-672

## Description

The product uses, accesses, or otherwise operates on a resource after that resource has been expired, released, or revoked.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2009-3547: Chain: race condition (CWE-362) might allow resource to be released before operating on it, leading to NULL dereference (CWE-476)

## Modes of Introduction

**•** Implementation: N/A

**•** Operation: N/A

## Common Consequences

**•** Impact: Modify Application Data, Read Application Data — Notes: If a released resource is subsequently reused or reallocated, then an attempt to use the original resource might allow access to sensitive data that is associated with a different user or entity.

**•** Impact: Other, DoS: Crash, Exit, or Restart — Notes: When a resource is released it might not be in an expected state, later attempts to access the resource may lead to resultant errors that may lead to a crash.

## Applicable Platforms

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

## Demonstrative Examples

**•** When an error occurs, the pointer is immediately freed. However, this pointer is later incorrectly used in the logError function.

**•** Double free vulnerabilities have two common (and sometimes overlapping) causes:

**•** However, the call to the method logError includes the messageBody after the memory for messageBody has been released using the free method. This can cause unexpected results and may lead to system crashes. A variable should never be used after its memory resources have been released.