# CWE Detail – CWE-676

## Description

The product invokes a potentially dangerous function that could introduce a vulnerability if it is used incorrectly, but the function can also be used safely.

## Extended Description

N/A

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2007-1470: Library has multiple buffer overflows using sprintf() and strcpy()

**•** CVE-2009-3849: Buffer overflow using strcat()

**•** CVE-2006-2114: Buffer overflow using strcpy()

**•** CVE-2006-0963: Buffer overflow using strcpy()

**•** CVE-2011-0712: Vulnerable use of strcpy() changed to use safer strlcpy()

**•** CVE-2008-5005: Buffer overflow using strcpy()

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Varies by Context, Quality Degradation, Unexpected State — Notes: If the function is used incorrectly, then it could result in security problems.

## Potential Mitigations

**•** Build and Compilation: Identify a list of prohibited API functions and prohibit developers from using these functions, providing safer alternatives. In some cases, automatic code analysis tools or the compiler can be instructed to spot use of prohibited functions, such as the "banned.h" include file from Microsoft's SDL. [REF-554] [REF-7] (Effectiveness: N/A)

## Applicable Platforms

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

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

## Demonstrative Examples

**•** However, the programmer does not ensure that the size of the data pointed to by string will fit in the local buffer and copies the data with the potentially dangerous strcpy() function. This may result in a buffer overflow condition if an attacker can influence the contents of the string parameter.

## Notes

**•** Relationship: This weakness is different than CWE-242 (Use of Inherently Dangerous Function). CWE-242 covers functions with such significant security problems that they can never be guaranteed to be safe. Some functions, if used properly, do not directly pose a security risk, but can introduce a weakness if not called correctly. These are regarded as potentially dangerous. A well-known example is the strcpy() function. When provided with a destination buffer that is larger than its source, strcpy() will not overflow. However, it is so often misused that some developers prohibit strcpy() entirely.