# CWE Detail – CWE-463

## Description

The accidental deletion of a data-structure sentinel can cause serious programming logic problems.

## Extended Description

Often times data-structure sentinels are used to mark structure of the data structure. A common example of this is the null character at the end of strings. Another common example is linked lists which may contain a sentinel to mark the end of the list. It is dangerous to allow this type of control data to be easily accessible. Therefore, it is important to protect from the deletion or modification outside of some wrapper interface which provides safety.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Other — Notes: Generally this error will cause the data structure to not work properly.

**•** Impact: Other — Notes: If a control character, such as NULL is removed, one may cause resource access control problems.

## Potential Mitigations

**•** Architecture and Design: Use an abstraction library to abstract away risky APIs. Not a complete solution. (Effectiveness: N/A)

**•** Build and Compilation: Run or compile the software using features or extensions that automatically provide a protection mechanism that mitigates or eliminates buffer overflows. For example, certain compilers and extensions provide automatic buffer overflow detection mechanisms that are built into the compiled code. Examples include the Microsoft Visual Studio /GS flag, Fedora/Red Hat FORTIFY\_SOURCE GCC flag, StackGuard, and ProPolice. (Effectiveness: Defense in Depth)

**•** Operation: Use OS-level preventative functionality. Not a complete solution. (Effectiveness: N/A)

## Applicable Platforms

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

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

## Demonstrative Examples

**•** The string foo has space for 9 characters and a null terminator, but 10 characters are written to it. As a result, the string foo is not null terminated and calling printf() on it will have unpredictable and possibly dangerous results.