# CWE Detail – CWE-477

## Description

The code uses deprecated or obsolete functions, which suggests that the code has not been actively reviewed or maintained.

## Extended Description

As programming languages evolve, functions occasionally become obsolete due to: Advances in the language Improved understanding of how operations should be performed effectively and securely Changes in the conventions that govern certain operations Functions that are removed are usually replaced by newer counterparts that perform the same task in some different and hopefully improved way.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Quality Degradation — Notes:

## Potential Mitigations

**•** Implementation: Refer to the documentation for the obsolete function in order to determine why it is deprecated or obsolete and to learn about alternative ways to achieve the same functionality. (Effectiveness: N/A)

**•** Requirements: Consider seriously the security implications of using an obsolete function. Consider using alternate functions. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** Although the code often behaves correctly, using the getpw() function can be problematic from a security standpoint, because it can overflow the buffer passed to its second parameter. Because of this vulnerability, getpw() has been supplanted by getpwuid(), which performs the same lookup as getpw() but returns a pointer to a statically-allocated structure to mitigate the risk. Not all functions are deprecated or replaced because they pose a security risk. However, the presence of an obsolete function often indicates that the surrounding code has been neglected and may be in a state of disrepair. Software security has not been a priority, or even a consideration, for very long. If the program uses deprecated or obsolete functions, it raises the probability that there are security problems lurking nearby.

**•** N/A

**•** In this example, the constructor may not correctly convert bytes to characters depending upon which charset is used to encode the string represented by nameBytes. Due to the evolution of the charsets used to encode strings, this constructor was deprecated and replaced by a constructor that accepts as one of its parameters the name of the charset used to encode the bytes for conversion.