# CWE Detail – CWE-242

## Description

The product calls a function that can never be guaranteed to work safely.

## Extended Description

Certain functions behave in dangerous ways regardless of how they are used. Functions in this category were often implemented without taking security concerns into account. The gets() function is unsafe because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to gets() and overflow the destination buffer. Similarly, the >> operator is unsafe to use when reading into a statically-allocated character array because it does not perform bounds checking on the size of its input. An attacker can easily send arbitrarily-sized input to the >> operator and overflow the destination buffer.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2007-4004: FTP client uses inherently insecure gets() function and is setuid root on some systems, allowing buffer overflow

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Varies by Context — Notes:

## Potential Mitigations

**•** Implementation: Ban the use of dangerous functions. Use their safe equivalent. (Effectiveness: N/A)

**•** Testing: Use grep or static analysis tools to spot usage of dangerous functions. (Effectiveness: N/A)

## Applicable Platforms

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

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

## Demonstrative Examples

**•** The gets() function in C is inherently unsafe.

**•** However, gets() is inherently unsafe, because it copies all input from STDIN to the buffer without checking size. This allows the user to provide a string that is larger than the buffer size, resulting in an overflow condition.