# CWE Detail – CWE-648

## Description

The product does not conform to the API requirements for a function call that requires extra privileges. This could allow attackers to gain privileges by causing the function to be called incorrectly.

## Extended Description

When a product contains certain functions that perform operations requiring an elevated level of privilege, the caller of a privileged API must be careful to: ensure that assumptions made by the APIs are valid, such as validity of arguments account for known weaknesses in the design/implementation of the API call the API from a safe context If the caller of the API does not follow these requirements, then it may allow a malicious user or process to elevate their privilege, hijack the process, or steal sensitive data. For instance, it is important to know if privileged APIs do not shed their privileges before returning to the caller or if the privileged function might make certain assumptions about the data, context or state information passed to it by the caller. It is important to always know when and how privileged APIs can be called in order to ensure that their elevated level of privilege cannot be exploited.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Observed Examples (CVEs)

**•** CVE-2003-0645: A Unix utility that displays online help files, if installed setuid, could allow a local attacker to gain privileges when a particular file-opening function is called.

## Related Attack Patterns (CAPEC)

* CAPEC-107
* CAPEC-234

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

**•** Operation: N/A

## Common Consequences

**•** Impact: Gain Privileges or Assume Identity — Notes: An attacker may be able to elevate privileges.

**•** Impact: Read Application Data — Notes: An attacker may be able to obtain sensitive information.

**•** Impact: Execute Unauthorized Code or Commands — Notes: An attacker may be able to execute code.

## Potential Mitigations

**•** Implementation: Before calling privileged APIs, always ensure that the assumptions made by the privileged code hold true prior to making the call. (Effectiveness: N/A)

**•** Architecture and Design: Know architecture and implementation weaknesses of the privileged APIs and make sure to account for these weaknesses before calling the privileged APIs to ensure that they can be called safely. (Effectiveness: N/A)

**•** Implementation: If privileged APIs make certain assumptions about data, context or state validity that are passed by the caller, the calling code must ensure that these assumptions have been validated prior to making the call. (Effectiveness: N/A)

**•** Implementation: If privileged APIs do not shed their privilege prior to returning to the calling code, then calling code needs to shed these privileges immediately and safely right after the call to the privileged APIs. In particular, the calling code needs to ensure that a privileged thread of execution will never be returned to the user or made available to user-controlled processes. (Effectiveness: N/A)

**•** Implementation: Only call privileged APIs from safe, consistent and expected state. (Effectiveness: N/A)

**•** Implementation: Ensure that a failure or an error will not leave a system in a state where privileges are not properly shed and privilege escalation is possible (i.e. fail securely with regards to handling of privileges). (Effectiveness: N/A)

## Applicable Platforms

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