# CWE Detail – CWE-831

## Description

The product defines a function that is used as a handler for more than one signal.

## Extended Description

While sometimes intentional and safe, when the same function is used to handle multiple signals, a race condition could occur if the function uses any state outside of its local declaration, such as global variables or non-reentrant functions, or has any side effects. An attacker could send one signal that invokes the handler function; in many OSes, this will typically prevent the same signal from invoking the handler again, at least until the handler function has completed execution. However, the attacker could then send a different signal that is associated with the same handler function. This could interrupt the original handler function while it is still executing. If there is shared state, then the state could be corrupted. This can lead to a variety of potential consequences depending on context, including denial of service and code execution. Another rarely-explored possibility arises when the signal handler is only designed to be executed once (if at all). By sending multiple signals, an attacker could invoke the function more than once. This may generate extra, unintended side effects. A race condition might not even be necessary; the attacker could send one signal, wait until it is handled, then send the other signal.

## Threat-Mapped Scoring

Score: 1.9

Priority: P3 - Important (Medium)

## Common Consequences

**•** Impact: DoS: Crash, Exit, or Restart, Execute Unauthorized Code or Commands, Read Application Data, Gain Privileges or Assume Identity, Bypass Protection Mechanism, Varies by Context — Notes: The most common consequence will be a corruption of the state of the product, possibly leading to a crash or exit. However, if the signal handler is operating on state variables for security relevant libraries or protection mechanisms, the consequences can be far more severe, including protection mechanism bypass, privilege escalation, or information exposure.

## Demonstrative Examples

**•** N/A

**•** The handler function uses global state (globalVar and logMessage), and it can be called by both the SIGHUP and SIGTERM signals. An attack scenario might follow these lines: