# CWE Detail – CWE-781

## Description

The product defines an IOCTL that uses METHOD\_NEITHER for I/O, but it does not validate or incorrectly validates the addresses that are provided.

## Extended Description

When an IOCTL uses the METHOD\_NEITHER option for I/O control, it is the responsibility of the IOCTL to validate the addresses that have been supplied to it. If validation is missing or incorrect, attackers can supply arbitrary memory addresses, leading to code execution or a denial of service.

## Threat-Mapped Scoring

Score: 1.9

Priority: P3 - Important (Medium)

## Observed Examples (CVEs)

**•** CVE-2006-2373: Driver for file-sharing and messaging protocol allows attackers to execute arbitrary code.

**•** CVE-2009-0686: Anti-virus product does not validate addresses, allowing attackers to gain SYSTEM privileges.

**•** CVE-2009-0824: DVD software allows attackers to cause a crash.

**•** CVE-2008-5724: Personal firewall allows attackers to gain SYSTEM privileges.

**•** CVE-2007-5756: chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Memory, Read Memory, Execute Unauthorized Code or Commands, DoS: Crash, Exit, or Restart — Notes: An attacker may be able to access memory that belongs to another process or user. If the attacker can control the contents that the IOCTL writes, it may lead to code execution at high privilege levels. At the least, a crash can occur.

## Potential Mitigations

**•** Implementation: If METHOD\_NEITHER is required for the IOCTL, then ensure that all user-space addresses are properly validated before they are first accessed. The ProbeForRead and ProbeForWrite routines are available for this task. Also properly protect and manage the user-supplied buffers, since the I/O Manager does not do this when METHOD\_NEITHER is being used. See References. (Effectiveness: N/A)

**•** Architecture and Design: If possible, avoid using METHOD\_NEITHER in the IOCTL and select methods that effectively control the buffer size, such as METHOD\_BUFFERED, METHOD\_IN\_DIRECT, or METHOD\_OUT\_DIRECT. (Effectiveness: N/A)

**•** Architecture and Design: If the IOCTL is part of a driver that is only intended to be accessed by trusted users, then use proper access control for the associated device or device namespace. See References. (Effectiveness: N/A)

## Applicable Platforms

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

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

## Notes

**•** Applicable Platform: Because IOCTL functionality is typically performing low-level actions and closely interacts with the operating system, this weakness may only appear in code that is written in low-level languages.

**•** Research Gap: While this type of issue has been known since 2006, it is probably still under-studied and under-reported. Most of the focus has been on high-profile software and security products, but other kinds of system software also use drivers. Since exploitation requires the development of custom code, it requires some skill to find this weakness. Because exploitation typically requires local privileges, it might not be a priority for active attackers. However, remote exploitation may be possible for software such as device drivers. Even when remote vectors are not available, it may be useful as the final privilege-escalation step in multi-stage remote attacks against application-layer software, or as the primary attack by a local user on a multi-user system.