# CWE Detail – CWE-782

## Description

The product implements an IOCTL with functionality that should be restricted, but it does not properly enforce access control for the IOCTL.

## Extended Description

When an IOCTL contains privileged functionality and is exposed unnecessarily, attackers may be able to access this functionality by invoking the IOCTL. Even if the functionality is benign, if the programmer has assumed that the IOCTL would only be accessed by a trusted process, there may be little or no validation of the incoming data, exposing weaknesses that would never be reachable if the attacker cannot call the IOCTL directly. The implementations of IOCTLs will differ between operating system types and versions, so the methods of attack and prevention may vary widely.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2009-2208: Operating system does not enforce permissions on an IOCTL that can be used to modify network settings.

**•** CVE-2008-3831: Device driver does not restrict ioctl calls to its direct rendering manager.

**•** CVE-2008-3525: ioctl does not check for a required capability before processing certain requests.

**•** CVE-2008-0322: Chain: insecure device permissions allows access to an IOCTL, allowing arbitrary memory to be overwritten.

**•** CVE-2007-4277: Chain: anti-virus product uses weak permissions for a device, leading to resultant buffer overflow in an exposed IOCTL.

**•** CVE-2007-1400: Chain: sandbox allows opening of a TTY device, enabling shell commands through an exposed ioctl.

**•** CVE-2006-4926: Anti-virus product uses insecure security descriptor for a device driver, allowing access to a privileged IOCTL.

**•** CVE-1999-0728: Unauthorized user can disable keyboard or mouse by directly invoking a privileged IOCTL.

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: REALIZATION: This weakness is caused during implementation of an architectural security tactic.

## Common Consequences

**•** Impact: Varies by Context — Notes: Attackers can invoke any functionality that the IOCTL offers. Depending on the functionality, the consequences may include code execution, denial-of-service, and theft of data.

## Potential Mitigations

**•** Architecture and Design: In Windows environments, 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

**•** Relationship: This can be primary to many other weaknesses when the programmer assumes that the IOCTL can only be accessed by trusted parties. For example, a program or driver might not validate incoming addresses in METHOD\_NEITHER IOCTLs in Windows environments (CWE-781), which could allow buffer overflow and similar attacks to take place, even when the attacker never should have been able to access the IOCTL at all.

**•** 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.