# CWE Detail – CWE-1260

## Description

The product allows address regions to overlap, which can result in the bypassing of intended memory protection.

## Extended Description

Isolated memory regions and access control (read/write) policies are used by hardware to protect privileged software. Software components are often allowed to change or remap memory region definitions in order to enable flexible and dynamically changeable memory management by system software. If a software component running at lower privilege can program a memory address region to overlap with other memory regions used by software running at higher privilege, privilege escalation may be available to attackers. The memory protection unit (MPU) logic can incorrectly handle such an address overlap and allow the lower-privilege software to read or write into the protected memory region, resulting in privilege escalation attack. An address overlap weakness can also be used to launch a denial of service attack on the higher-privilege software memory regions.

## Threat-Mapped Scoring

Score: 1.9

Priority: P3 - Important (Medium)

## Observed Examples (CVEs)

**•** CVE-2008-7096: virtualization product allows compromise of hardware product by accessing certain remapping registers.

**•** [REF-1100]: processor design flaw allows ring 0 code to access more privileged rings by causing a register window to overlap a range of protected system RAM [REF-1100]

## Related Attack Patterns (CAPEC)

* CAPEC-456
* CAPEC-679

## Modes of Introduction

**•** Architecture and Design: Such issues could be introduced during hardware architecture and design or implementation and identified later during the Testing phase.

**•** Implementation: N/A

## Common Consequences

**•** Impact: Modify Memory, Read Memory, DoS: Instability — Notes:

## Potential Mitigations

**•** Architecture and Design: Ensure that memory regions are isolated as intended and that access control (read/write) policies are used by hardware to protect privileged software. (Effectiveness: N/A)

**•** Implementation: For all of the programmable memory protection regions, the memory protection unit (MPU) design can define a priority scheme. For example: if three memory regions can be programmed (Region\_0, Region\_1, and Region\_2), the design can enforce a priority scheme, such that, if a system address is within multiple regions, then the region with the lowest ID takes priority and the access-control policy of that region will be applied. In some MPU designs, the priority scheme can also be programmed by trusted software. Hardware logic or trusted firmware can also check for region definitions and block programming of memory regions with overlapping addresses. The memory-access-control-check filter can also be designed to apply a policy filter to all of the overlapping ranges, i.e., if an address is within Region\_0 and Region\_1, then access to this address is only granted if both Region\_0 and Region\_1 policies allow the access. (Effectiveness: High)

## Applicable Platforms

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

## Demonstrative Examples

**•** This design could be improved in several ways.

**•** In this example, we have 12 IP agents, and only 4 of them are called out for illustration purposes in the code snippets. Access to the AES IP MMIO region is considered privileged as it provides access to AES secret key, internal states, or decrypted data.

## Notes

**•** Maintenance: As of CWE 4.6, CWE-1260 and CWE-1316 are siblings under view 1000, but CWE-1260 might be a parent of CWE-1316. More analysis is warranted.