# CWE Detail – CWE-1303

## Description

Hardware structures shared across execution contexts (e.g., caches and branch predictors) can violate the expected architecture isolation between contexts.

## Extended Description

Modern processors use techniques such as out-of-order execution, speculation, prefetching, data forwarding, and caching to increase performance. Details about the implementation of these techniques are hidden from the programmer's view. This is problematic when the hardware implementation of these techniques results in resources being shared across supposedly isolated contexts. Contention for shared resources between different contexts opens covert channels that allow malicious programs executing in one context to recover information from another context. Some examples of shared micro-architectural resources that have been used to leak information between contexts are caches, branch prediction logic, and load or store buffers. Speculative and out-of-order execution provides an attacker with increased control over which data is leaked through the covert channel. If the extent of resource sharing between contexts in the design microarchitecture is undocumented, it is extremely difficult to ensure system assets are protected against disclosure.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Related Attack Patterns (CAPEC)

* CAPEC-663

## Modes of Introduction

**•** Architecture and Design: Such issues could be introduced during hardware architecture and design and identified later during Testing or System Configuration phases.

**•** Implementation: Such issues could be introduced during implementation and identified later during Testing or System Configuration phases.

## Common Consequences

**•** Impact: Read Application Data, Read Memory — Notes: Microarchitectural side-channels have been used to leak specific information such as cryptographic keys, and Address Space Layout Randomization (ALSR) offsets as well as arbitrary memory.

## Potential Mitigations

**•** Architecture and Design: Microarchitectural covert channels can be addressed using a mixture of hardware and software mitigation techniques. These include partitioned caches, new barrier and flush instructions, and disabling high resolution performance counters and timers. (Effectiveness: N/A)

**•** Requirements: Microarchitectural covert channels can be addressed using a mixture of hardware and software mitigation techniques. These include partitioned caches, new barrier and flush instructions, and disabling high resolution performance counters and timers. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

## Notes

**•** Maintenance: As of CWE 4.9, members of the CWE Hardware SIG are closely analyzing this entry and others to improve CWE's coverage of transient execution weaknesses, which include issues related to Spectre, Meltdown, and other attacks. Additional investigation may include other weaknesses related to microarchitectural state. Finally, this entry's demonstrative example might not be appropriate. As a result, this entry might change significantly in CWE 4.10.