# CWE Detail – CWE-1331

## Description

The Network On Chip (NoC) does not isolate or incorrectly isolates its on-chip-fabric and internal resources such that they are shared between trusted and untrusted agents, creating timing channels.

## Extended Description

Typically, network on chips (NoC) have many internal resources that are shared between packets from different trust domains. These resources include internal buffers, crossbars and switches, individual ports, and channels. The sharing of resources causes contention and introduces interference between differently trusted domains, which poses a security threat via a timing channel, allowing attackers to infer data that belongs to a trusted agent. This may also result in introducing network interference, resulting in degraded throughput and latency.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Observed Examples (CVEs)

**•** CVE-2021-33096: Improper isolation of shared resource in a network-on-chip leads to denial of service

## Related Attack Patterns (CAPEC)

* CAPEC-124

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: DoS: Resource Consumption (Other), Varies by Context, Other — Notes: Attackers may infer data that belongs to a trusted agent. The methods used to perform this attack may result in noticeably increased resource consumption.

## Potential Mitigations

**•** Architecture and Design: Implement priority-based arbitration inside the NoC and have dedicated buffers or virtual channels for routing secret data from trusted agents. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** One of the masters to this NoC implements a cryptographic algorithm (RSA), and another master to the NoC is a core that can be exercised by an attacker. The RSA algorithm performs a modulo multiplication of two large numbers and depends on each bit of the secret key. The algorithm examines each bit in the secret key and only performs multiplication if the bit is 1. This algorithm is known to be prone to timing attacks. Whenever RSA performs multiplication, there is additional network traffic to the memory controller. One of the reasons for this is cache conflicts.