# CWE Detail – CWE-1272

## Description

The product performs a power or debug state transition, but it does not clear sensitive information that should no longer be accessible due to changes to information access restrictions.

## Extended Description

A device or system frequently employs many power and sleep states during its normal operation (e.g., normal power, additional power, low power, hibernate, deep sleep, etc.). A device also may be operating within a debug condition. State transitions can happen from one power or debug state to another. If there is information available in the previous state which should not be available in the next state and is not properly removed before the transition into the next state, sensitive information may leak from the system.

## Threat-Mapped Scoring

Score: 3.0

Priority: P2 - Serious (High)

## Observed Examples (CVEs)

**•** CVE-2020-12926: Product software does not set a flag as per TPM specifications, thereby preventing a failed authorization attempt from being recorded after a loss of power.

## Related Attack Patterns (CAPEC)

* CAPEC-150
* CAPEC-37
* CAPEC-545
* CAPEC-546

## Attack TTPs

**•** T1003: OS Credential Dumping (Tactics: credential-access)

**•** T1602: Data from Configuration Repository (Tactics: collection)

**•** T1555.001: Keychain (Tactics: credential-access)

**•** T1119: Automated Collection (Tactics: collection)

**•** T1530: Data from Cloud Storage (Tactics: collection)

**•** T1005: Data from Local System (Tactics: collection)

**•** T1555: Credentials from Password Stores (Tactics: credential-access)

**•** T1552.004: Private Keys (Tactics: credential-access)

**•** T1213: Data from Information Repositories (Tactics: collection)

## Modes of Introduction

**•** Architecture and Design: N/A

## Common Consequences

**•** Impact: Read Memory, Read Application Data — Notes: Sensitive information may be used to unlock additional capabilities of the device and take advantage of hidden functionalities which could be used to compromise device security.

## Potential Mitigations

**•** Architecture and Design: During state transitions, information not needed in the next state should be removed before the transition to the next state. (Effectiveness: N/A)

## Applicable Platforms

**•** VHDL (Class: None, Prevalence: Undetermined)

**•** Verilog (Class: None, Prevalence: Undetermined)

**•** None (Class: Hardware Description Language, Prevalence: Undetermined)

## Demonstrative Examples

**•** Suppose a device is transitioning from state A to state B. During state A, it can read certain private keys from the hidden fuses that are only accessible in state A but not in state B. The device reads the keys, performs operations using those keys, then transitions to state B, where those private keys should no longer be accessible.