# CWE Detail – CWE-1290

## Description

The product implements a decoding mechanism to decode certain bus-transaction signals to security identifiers. If the decoding is implemented incorrectly, then untrusted agents can now gain unauthorized access to the asset.

## Extended Description

In a System-On-Chip (SoC), various integrated circuits and hardware engines generate transactions such as to access (reads/writes) assets or perform certain actions (e.g., reset, fetch, compute, etc.). Among various types of message information, a typical transaction is comprised of source identity (to identify the originator of the transaction) and a destination identity (to route the transaction to the respective entity). Sometimes the transactions are qualified with a security identifier. The security identifier helps the destination agent decide on the set of allowed actions (e.g., access an asset for read and writes). A decoder decodes the bus transactions to map security identifiers into necessary access-controls/protections. A common weakness that can exist in this scenario is incorrect decoding because an untrusted agent's security identifier is decoded into a trusted agent's security identifier. Thus, an untrusted agent previously without access to an asset can now gain access to the asset.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Implementation: N/A

**•** Architecture and Design: N/A

## Common Consequences

**•** Impact: Modify Memory, Read Memory, DoS: Resource Consumption (Other), Execute Unauthorized Code or Commands, Gain Privileges or Assume Identity, Quality Degradation — Notes:

## Potential Mitigations

**•** Architecture and Design: Security identifier decoders must be reviewed for design consistency and common weaknesses. (Effectiveness: N/A)

**•** Implementation: Access and programming flows must be tested in pre-silicon and post-silicon testing in order to check for this weakness. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** The following Pseudo code outlines the process of checking the value of the Security Identifier within the AES\_KEY\_ACCESS\_POLICY register: