# CWE Detail – CWE-1191

## Description

The chip does not implement or does not correctly perform access control to check whether users are authorized to access internal registers and test modes through the physical debug/test interface.

## Extended Description

A device's internal information may be accessed through a scan chain of interconnected internal registers, usually through a JTAG interface. The JTAG interface provides access to these registers in a serial fashion in the form of a scan chain for the purposes of debugging programs running on a device. Since almost all information contained within a device may be accessed over this interface, device manufacturers typically insert some form of authentication and authorization to prevent unintended use of this sensitive information. This mechanism is implemented in addition to on-chip protections that are already present. If authorization, authentication, or some other form of access control is not implemented or not implemented correctly, a user may be able to bypass on-chip protection mechanisms through the debug interface. Sometimes, designers choose not to expose the debug pins on the motherboard. Instead, they choose to hide these pins in the intermediate layers of the board. This is primarily done to work around the lack of debug authorization inside the chip. In such a scenario (without debug authorization), when the debug interface is exposed, chip internals are accessible to an attacker.

## Threat-Mapped Scoring

Score: 3.0

Priority: P2 - Serious (High)

## Observed Examples (CVEs)

**•** CVE-2019-18827: chain: JTAG interface is not disabled (CWE-1191) during ROM code execution, introducing a race condition (CWE-362) to extract encryption keys

## Related Attack Patterns (CAPEC)

* CAPEC-1
* CAPEC-180

## Attack TTPs

**•** T1574.010: Services File Permissions Weakness (Tactics: persistence, privilege-escalation, defense-evasion)

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Application Data — Notes:

**•** Impact: Read Memory — Notes:

**•** Impact: Execute Unauthorized Code or Commands — Notes:

**•** Impact: Modify Memory — Notes:

**•** Impact: Modify Application Data — Notes:

**•** Impact: Bypass Protection Mechanism — Notes:

## Potential Mitigations

**•** Architecture and Design: If feasible, the manufacturer should disable the JTAG interface or implement authentication and authorization for the JTAG interface. If authentication logic is added, it should be resistant to timing attacks. Security-sensitive data stored in registers, such as keys, etc. should be cleared when entering debug mode. (Effectiveness: High)

## Applicable Platforms

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

## Demonstrative Examples

**•** JTAG is useful to chip and device manufacturers during design, testing, and production and is included in nearly every product. Without proper authentication and authorization, the interface may allow tampering with a product.

**•** The developers employed a Finite State Machine (FSM) to implement this authentication. When a user intends to read from or write to the JTAG module, they must input a password.

**•** The vulnerable code shows an incorrect implementation of the HMAC authentication where it only uses the least significant 32 bits of the secret message for the authentication (the remaining 480 bits are hard coded as zeros). As a result, the system is susceptible to brute-force attacks on the access control mechanism of JTAG, where the attacker only needs to determine 32 bits of the secret message instead of 512 bits.

## Notes

**•** Relationship: CWE-1191 and CWE-1244 both involve physical debug access,
 but the weaknesses are different. CWE-1191 is effectively
 about missing authorization for a debug interface,
 i.e. JTAG. CWE-1244 is about providing internal assets with
 the wrong debug access level, exposing the asset to
 untrusted debug agents.