# CWE Detail – CWE-1323

## Description

Trace data collected from several sources on the
 System-on-Chip (SoC) is stored in unprotected locations or
 transported to untrusted agents.

## Extended Description

To facilitate verification of complex System-on-Chip
 (SoC) designs, SoC integrators add specific IP blocks that
 trace the SoC's internal signals in real-time. This
 infrastructure enables observability of the SoC's internal
 behavior, validation of its functional design,
 and detection of hardware and software bugs. Such tracing
 IP blocks collect traces from several sources on the SoC
 including the CPU, crypto coprocessors, and on-chip fabrics. Traces collected from these sources are then
 aggregated inside trace IP block and forwarded to trace
 sinks, such as debug-trace ports that facilitate debugging by
 external hardware and software debuggers. Since
 these traces are collected from several security-sensitive
 sources, they must be protected against untrusted
 debuggers. If they are stored in unprotected memory, an
 untrusted software debugger can access these traces and
 extract secret information. Additionally, if
 security-sensitive traces are not tagged as secure, an
 untrusted hardware debugger might access them to extract
 confidential information.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Related Attack Patterns (CAPEC)

* CAPEC-150
* CAPEC-167
* CAPEC-545

## 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)

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

## Modes of Introduction

**•** Architecture and Design: N/A

**•** Implementation: N/A

## Common Consequences

**•** Impact: Read Memory — Notes: An adversary can read secret values if they are captured in debug traces and stored unsafely.

## Potential Mitigations

**•** Implementation: Tag traces to indicate owner and debugging privilege level (designer, OEM, or end user) needed to access that trace. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A