# CWE Detail – CWE-1293

## Description

The product relies on one source of data, preventing the ability to detect if an adversary has compromised a data source.

## Extended Description

To operate successfully, a product sometimes has to implicitly trust the integrity of an information source. When information is implicitly signed, one can ensure that the data was not tampered in transit. This does not ensure that the information source was not compromised when responding to a request. By requesting information from multiple sources, one can check if all of the data is the same. If they are not, the system should report the information sources that respond with a different or minority value as potentially compromised. If there are not enough answers to provide a majority or plurality of responses, the system should report all of the sources as potentially compromised. As the seriousness of the impact of incorrect integrity increases, so should the number of independent information sources that would need to be queried.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Architecture and Design: This flaw could be introduced during the design of the application or misconfiguration at run time by only specifying a single point of validation.

**•** Implementation: Such issues could be introduced during hardware implementation, then identified later during Testing or System Configuration phases.

**•** Operation: This weakness could be introduced by intentionally failing all but one of the devices used to retrieve the data or by failing the devices that validate the data.

## Common Consequences

**•** Impact: Read Application Data, Modify Application Data, Gain Privileges or Assume Identity — Notes: An attacker that may be able to execute a single Person-in-the-Middle attack can subvert a check of an external oracle (e.g. the ACME protocol check for a file on a website), and thus inject an arbitrary reply to the single perspective request to the external oracle.

## Potential Mitigations

**•** Requirements: Design system to use a Practical Byzantine fault method, to request information from multiple sources to verify the data and report on potentially compromised information sources. (Effectiveness: N/A)

**•** Implementation: Failure to use a Practical Byzantine fault method when requesting data. Lack of place to report potentially compromised information sources. Relying on non-independent information sources for integrity checking. Failure to report information sources that respond in the minority to incident response procedures. (Effectiveness: N/A)

## Applicable Platforms

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