# CWE Detail – CWE-353

## Description

The product uses a transmission protocol that does not include a mechanism for verifying the integrity of the data during transmission, such as a checksum.

## Extended Description

If integrity check values or "checksums" are omitted from a protocol, there is no way of determining if data has been corrupted in transmission. The lack of checksum functionality in a protocol removes the first application-level check of data that can be used. The end-to-end philosophy of checks states that integrity checks should be performed at the lowest level that they can be completely implemented. Excluding further sanity checks and input validation performed by applications, the protocol's checksum is the most important level of checksum, since it can be performed more completely than at any previous level and takes into account entire messages, as opposed to single packets.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Related Attack Patterns (CAPEC)

* CAPEC-13
* CAPEC-14
* CAPEC-389
* CAPEC-39
* CAPEC-665
* CAPEC-74
* CAPEC-75

## Attack TTPs

**•** T1574.007: Path Interception by PATH Environment Variable (Tactics: persistence, privilege-escalation, defense-evasion)

**•** T1574.006: Dynamic Linker Hijacking (Tactics: persistence, privilege-escalation, defense-evasion)

**•** T1542.002: Component Firmware (Tactics: persistence, defense-evasion)

**•** T1562.003: Impair Command History Logging (Tactics: defense-evasion)

**•** T1556: Modify Authentication Process (Tactics: credential-access, defense-evasion, persistence)

**•** T1211: Exploitation for Defense Evasion (Tactics: defense-evasion)

## Modes of Introduction

**•** Architecture and Design: OMISSION: This weakness is caused by missing a security tactic during the architecture and design phase.

**•** Implementation: N/A

## Common Consequences

**•** Impact: Other — Notes: Data that is parsed and used may be corrupted.

**•** Impact: Hide Activities, Other — Notes: Without a checksum it is impossible to determine if any changes have been made to the data after it was sent.

## Potential Mitigations

**•** Architecture and Design: Add an appropriately sized checksum to the protocol, ensuring that data received may be simply validated before it is parsed and used. (Effectiveness: N/A)

**•** Implementation: Ensure that the checksums present in the protocol design are properly implemented and added to each message before it is sent. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** The response containing secret data has no integrity check associated with it, allowing an attacker to alter the message without detection.