# CWE Detail – CWE-1395

## Description

The product has a dependency on a third-party component that contains one or more known vulnerabilities.

## Extended Description

Many products are large enough or complex enough that part of their functionality uses libraries, modules, or other intellectual property developed by third parties who are not the product creator. For example, even an entire operating system might be from a third-party supplier in some hardware products. Whether open or closed source, these components may contain publicly known vulnerabilities that could be exploited by adversaries to compromise the product.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Architecture and Design: The product architect or designer might choose a component that is already known to contain vulnerabilities or has a high likelihood of containing vulnerabilities in the future.

**•** Implementation: For reasons of compatibility or stability, developers might choose a third-party component, such as a library, that is already known to contain vulnerabilities.

**•** Patching and Maintenance: Since all products contain vulnerabilities, over time, a third-party component will be discovered to have a vulnerability.

## Common Consequences

**•** Impact: Varies by Context — Notes: The consequences vary widely, depending on the vulnerabilities that exist in the component; how those vulnerabilities can be "reached" by adversaries, as the exploitation paths and attack surface will vary depending on how the component is used; and the criticality of the privilege levels and features for which the product relies on the component.

## Potential Mitigations

**•** Requirements: In some industries such as healthcare [REF-1320] [REF-1322] or technologies such as the cloud [REF-1321], it might be unclear about who is responsible for applying patches for third-party vulnerabilities: the vendor, the operator/customer, or a separate service. Clarifying roles and responsibilities can be important to minimize confusion or unnecessary delay when third-party vulnerabilities are disclosed. (Effectiveness: N/A)

**•** Requirements: Require a Bill of Materials for all components and sub-components of the product. For software, require a Software Bill of Materials (SBOM) [REF-1247] [REF-1311]. (Effectiveness: N/A)

**•** Architecture and Design: Maintain a Bill of Materials for all components and sub-components of the product. For software, maintain a Software Bill of Materials (SBOM). According to [REF-1247], "An SBOM is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships." (Effectiveness: N/A)

**•** Operation: Actively monitor when a third-party component vendor announces vulnerability patches; fix the third-party component as soon as possible; and make it easy for operators/customers to obtain and apply the patch. (Effectiveness: N/A)

**•** Operation: Continuously monitor changes in each of the product's components, especially when the changes indicate new vulnerabilities, end-of-life (EOL) plans, etc. (Effectiveness: N/A)

## Applicable Platforms

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

## Demonstrative Examples

**•** N/A

**•** N/A