**T1** 2023

Coverity Scan Static Analysis Report

Hardhard Enterprises

Statement of Intent

Overview

This document aims to provide a record of static code analysis performed on a specific issue from the Coverity SAST scan for the NASA ION Open-Source code 4.1.1 project.

The primary purpose of this document is to validate the issue identified via the automated detection process to eliminate false positives.

Depending on findings, secondary purposes can include but are not limited to listing/providing recommended fixes alongside a list of attack vectors and potential exploits for consideration.

Reporting Best Practices

Please ensure best practices are kept when completing the document via regularly updating the Acronyms and Abbreviations table alongside any iterations made to the Document History table. This will allow other members to identify any updates and progress made across trimesters easily.

When using code snippets, please use screenshots that are clear and easy to read, alternatively, use words built-in code formatter found [here](https://appsource.microsoft.com/en-us/product/office/WA104382008?tab=Overview).

Document Naming Conventions

Naming conventions for this file are as follow; SAR\_{CID}. For example, when investigating issue 123456 the file name would be SAR\_123456.docx

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Dates** | **Version** | **Author** | **Comments** |
| 8/9/23 | 1.0 | Dean Scanlon | Initial document |
|  |  |  |  |
|  |  |  |  |

Table of Content

Contents

[Introduction 3](#_Toc119848724)

[Objective 3](#_Toc119848725)

[Scope 3](#_Toc119848726)

[Acronyms and Abbreviations 3](#_Toc119848727)

[Code Review and Analysis 4](#_Toc119848728)

[Outcomes 4](#_Toc119848729)

[Observations 4](#_Toc119848730)

[Supporting Evidence 4](#_Toc119848731)

[Conclusions and Recommendations 4](#_Toc119848732)

[References 5](#_Toc119848733)

[Appendix 6](#_Toc119848734)

# Introduction

## Objective

The primary objective of this analysis is to determine whether the defects identified in the Coverity Report for the ION Open Source 4.1.1 project are:

* Indeed, defects.
* Potentially exploitable.

The secondary objective of this analysis, where applicable, is to provide the following:

* Recommendation(s) to fix.
* Any exploit for consideration.

## Scope

This static code analysis is limited to the ***Insecure data handling*** type defect identified in the following CIDs:  
***CID\_1520874***

# Acronyms and Abbreviations

Please keep an updated list of acronyms and abbreviations used throughout the report.

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| DTN | Delay/Disruption Tolerant Network |
| ION | Interplanetary Overlay Network |
|  |  |

# Code Review and Analysis

## Overview

This issue occurs in the libltpP.c code segment within the ltp/library/ directory of the NASA ION DTN protocol.

Coverity Static Analysis Tool was able to detect **Medium Impact Quality** vulnerabilities that relate to an **Untrusted value as argument** error.

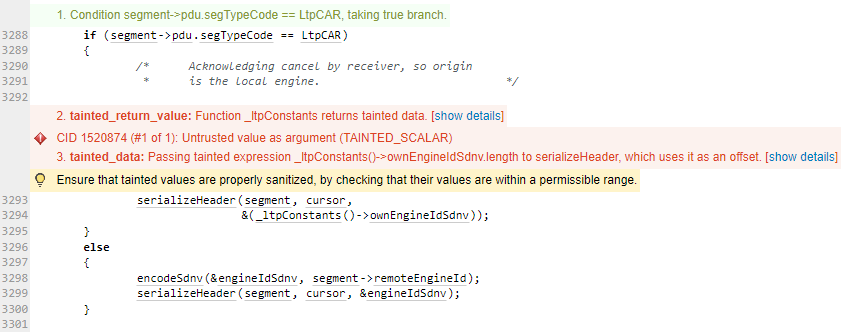
## Observations

The libltpP.c code segment is used as part of the Licklider Transport Protocol within the NASA ION DTN code. The serializeHeader function is passed an argument when the length function points to a position within the structure EngineIdSdnv. This value is then used by serializeHeader function as an offset. The value which is passed to the serializeHeader function is not validated prior and hence has the potential to be of a size or type which is not within the permissible range of the function using it as shown in Fig1.

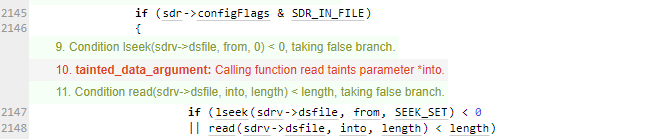
This can be considered an ‘Improper Input Validation’ error (CWE-20) and is flagged by Coverity, as are a number of other errors within NASA ION DTN, because of the lack of validation of values being passed between functions.

Investigation of this issue reveals that the read function taints the into parameter when this code library interacts with /ici/sdr/sdrxn.c code segment, as shown in Fig2. The value of the into function is then called by \_ltpConstants function and returns tainted data. This data is considered tainted by Coverity Static Analysis tool because it is not validated. The TAINTED\_SCALAR error message occurs in Coverity when it is required that the value of a variable is verified before use.

The memcpy function is ultimately affected by the lack of validation on the arguments it uses (Fig3), which could potentially calculate erroneous memory locations and cause problems for the functionality of the protocol.



**Fig1 Coverity error CID 1520874**



**Fig2 into function being tainted**

A close-up of a computer code

Description automatically generated

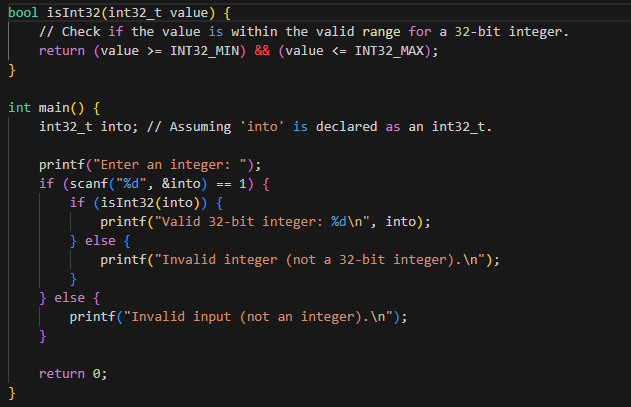
**Fig3 memcpy function being affected by lack of validation function in code**

# Conclusions and Recommendations

Broadly within the NASA ION DTN protocol, variables are not validated which can result in problems with the processing and allocation of memory within the code. NASA ION DTN should be altered to include such validation functions to ensure program correctness, security and stability.

Without proper validation, programs can access unallocated memory which can lead to improper computations of values and to undefined behaviour within the program.

Fig4 shows a basic example of a validation function which, in this case, checks that the variable **into** is an integer of size 4 bytes. If included in NASA ION DTN, this would alleviate flags from Coverity and ensure that each variable is of the correct size and/or type to be handled by functions within the code.



**Fig4 Example of validation function**

References  
Please keep an updated references list in APA7; The Deakin referencing guide can be found [here](https://www.deakin.edu.au/__data/assets/pdf_file/0009/2236752/Deakin-guide-to-APA7.pdf).

Appendix

Include additional information/documentation here to help the readers understand complex information.