**T2** 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** |
| 04/08/2023 | V0.1 | LIANG CHEN | Analyze Improper Input Validation |
|  |  |  |  |
|  |  |  |  |

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 5](#_Toc119848732)

[References 6](#_Toc119848733)

[Appendix 7](#_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 ***Improper Input Validation*** type defect identified in the following CIDs:  
***1520812***

# 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

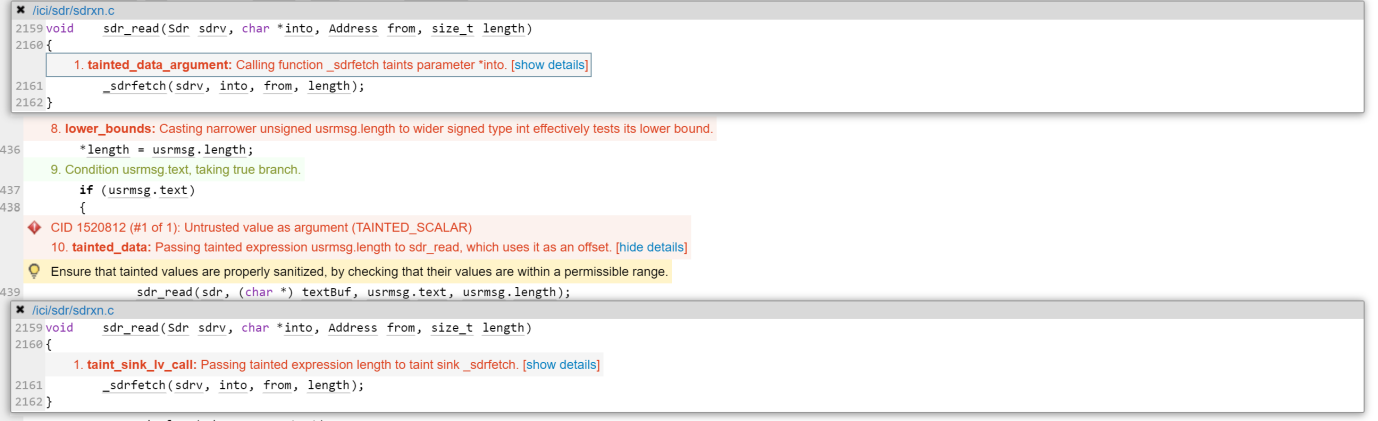
## This report is based on the error analysis of the “cfdp\_get\_usrmsg” code segment in the "/cfdp/library/libcfdp.c" file. The content includes an analysis of the data processing inside the function and its source. Three questions, warnings 7, 8, 10. They mainly deal with possible problems of data pollution and data type conversion.

## Observations

## error

Warnings 7 and 10 stem from data pollution issues, which can lead to security vulnerabilities as the program may use untrusted data without proper validation and sanitization. In this case, data read from files or other data sources may be untrusted, and if not properly handled, may have adverse effects on the program.Warning 8 involves the conversion of unsigned integers to signed integers. While such conversions are safe in many cases, they can lead to problems if not properly handled, especially when dealing with data that could potentially be controlled by an attacker.

## Supporting Evidence



warning 7 suggests that the read function may introduce data pollution by reading untrusted data into the program. Similarly, warning 10 points out that passing an untrusted length value to the memset function could lead to issues such as buffer overflow.The situation with warning 8 is a bit more complex. Although converting unsigned integers to signed integers is safe in most cases, if the input value could potentially be controlled by an attacker, extra caution is needed.

# Conclusions and Recommendations

Proper validation and sanitization should be applied to all data read from external sources such as files or networks. Only data that has been properly sanitized and validated can be safely used for further computation and processing. For example, we can add some code to check usrmsg The data is within the expected range. If not, we clean or adjust the data. When converting unsigned integers to signed integers, ensure that the converted value remains within the valid range. A possible strategy is to perform boundary checks on the unsigned integer to ensure that the converted value does not cause an overflow. A simple solution is to change The type of length is some other appropriate unsigned type.

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).

7 Pernicious Kingdoms (CWE Draft 3, 2006-07-19, Submitted on July 19, 2006).

<https://cwe.mitre.org/data/definitions/20.html>

Appendix

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