**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** |
| 14/9/23 | V1.0 | 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 ***API Usage errors*** type defect identified in the following CIDs:  
***1520801***

# 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 libcfdp.c code segment within the cfdp/utils/bpcpd.c directory of the NASA ION DTN protocol.

Coverity Static Analysis Tool was able to detect **Medium Impact Quality** vulnerabilities that relate to an **API Usage errors** issue.

## Observations

The **bpcpd.c** code segment is the source code for the Bundle Protocol Control Protocol Daemon which runs on each node and is responsible for managing data transmission on the DTN network.

The issue in question occurs within the **dbgprintf** **f**unction. This function is used to print formatted data to a debug log, and on this occasion is used by the **cfdb\_get\_event** function. The **cfdb\_get\_event** function is used to retrieve the next event in the event queue and calls on the **dgbprintf** function to write to a log for debugging purposes.

The **uvast** function is a typedef of the unsigned long data type as shown in Fig1. **TID12** is declared as **uvast** which is data type unsigned long integer, as denoted by **%ul** in c language, and is passed to the **printf** function which is expecting data type unsigned integer. In Fig2, the error message states that format specifier **%u** is the format specifier for the **printf** function. It is used by the **printf** function to print unsigned decimal integers and when an argument with a format specifier **%ul** is passed to the function, there is potential for erroneous computations and ultimately problems with memory allocation and the functionality of the program.

Ideally variables of format specifier **%u** need to be passed to the **printf** function. Any format specifier which is different will be flagged by Coverity. **TID12** relates to ‘time-sensitive’ data such as video, telemetry and command data so caution should be exercised in relation to data types being used by this function. Other type identifier values (**TID11**, **TID21**, **TID22**) are also used within the code and relate to ‘non-time-sensitive’, ‘high-priority’ and ‘emergency’ data, and proper data type management practices should be implemented within the code for these essential functions to avoid possible issues.

A screenshot of a computer

Description automatically generated

Fig1 uvast specified as unsigned long integer

A screenshot of a computer program

Description automatically generated

Fig2 Invalid type in argument to printf

# Conclusions and Recommendations

Due to the fact that the type identifier **TID12** interacts with **printf**, it should be assigned the data type unsigned integer **(%u**). Given that the code segment in which the error occurs is used to generate logs by the **dgb\_printf** function, it is possible that a function could be inserted to convert the data type from unsigned long integer type to unsigned integer for logging purposes. An example of a conversion from data type unsigned long integer to data type unsigned integer is included in Fig3. In this example, the **main()** function defines a **TID** value, then calls the conversion function to convert the **TID** to an unsigned integer.

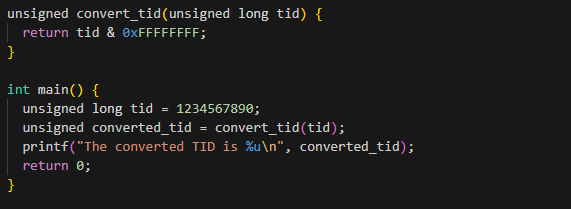


Fig3 Example function for conversion from unsigned long to unsigned integer

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.