**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** |
| 21/04/2023 | V0.1 | Anthony Scantsonihas | Beginning of the Investigation |
| 22/04/2023 | V0.2 | Anthony Scantsonihas | Investigation |
| 24/04/2023 | V0.3 | Anthony Scantsonihas | Investigation |
| 25/04/2023 | V1.0 | Anthony Scantsonihas | Finalization |

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 ***Out-of-Bounds access*** type defect identified in the following CIDs:  
***CID1520684***

# 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 |
| CID | Coverity Issue Identification Number |
| CWE | Common Weakness Enumeration |

# Code Review and Analysis

## Overview

The Coverity report for the CID 1520684 has flagged a Out-of-bounds access issue found within the code base dtn2fw.c:124. The Out-of-bounds access issue is a type of error that occurs when code attempts to read and write memory out of the intended boundary of the buffer, this issue can also be found under CWE-119.

## Observations

The error appears to stem from the enqueueBundle bundle function which is used to enque a bundle for transmission to a destination node identified by the endpoint ID that can be found within the bundle.

Text, letter

Description automatically generated

Diving deeper into the code it can be seen that the source of the issue appears to come from this if statement where it can cause the code to try and access an offset of 719 bytes whilst the buffer is only 8 bytes in total. This means that the code is trying to access a unassigned position within the designated array length for the program.

**-Potential Vulnerabilities**

It is possible that this fault within the code could be used by an attacker to prevent the program from working correctly as it would be trying to access data from outside the given length of the array.

## Supporting Evidence

A screenshot of the Coverity error flag can be seen below. As can be seen the error message acknowledges that the program has attempted to access at byte offset 719 using the argument 720UL. This error could potentially be used by an attacker to deny access to the application by forcing the program to try and access an index that doesn’t exist within the array length, therefore causing the program not to function correctly.



# 

# Conclusions and Recommendations

My recommendation to solve this issue would be to implement a mechanism within the code that recognises when the program is trying to access a byte within the set boundaries and then prevents it from doing so. With a method like this implemented the application will be unable run into this error and crash, whether it may be caused intentionally by an attacker or unintentionally by a bug within the software.

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).

MITRE Corporation. (2023, January 31). CWE - CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer. Retrieved March 21, 2023, from <https://cwe.mitre.org/data/definitions/119.html>

Appendix

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