**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** |
| 4/8/2023 | V0.1 | Anthony Scantsonihas | Beginning of investigation |
| 5/8/2023 | V0.5 | Anthony Scantsonihas | Investigation |
| 6/8/2023 | V0.8 | Anthony Scantsonihas | Investigation |
| 7/8/2023 | 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:  
***CID1520880***

# 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 flag for the CID 1520880 has discovered errors within the code base Itpadmin.c:723. The issue for this segment of code is an Out-of-bounds access error which is described under CWE-119. This issue indicates that a function within the code is attempting to read and write data that is outside the intended buffer.

## Observations

The error flagged by Coverity can be traced back to the function “manageMaxBER”. This function is used for managing the maximum bit rate for the Licklider Transmission Protocol (LTP) used within the ION program.

A screenshot of a chat

Description automatically generated

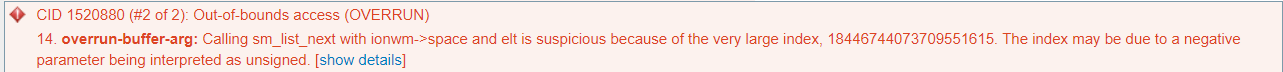
The error appears to be caused by a for loop within the function as the variable “elt” is assigned the value of the variable “sm\_list\_next” which has the potential to return the value 18446744073709551615 which is well out of the range of the programs intended buffer.

**-Potential Vulnerabilities**

There is a possibility that the error flagged by Coverity is a vulnerability that attackers could potentially exploit. In this case the attacker may try to exploit the “sm\_list\_next” variable to try to induce a buffer overflow error within the application. By doing this they could potentially prevent the program for functioning correctly and crash.

## Supporting Evidence

Evidence of this error can be seen within the error code flag below. As can be seen the variables ”sm\_list\_next” and “elt” have been flagged by Coverity as suspicious as they both may return a value that is not within the programs bounds.



# Conclusions and Recommendations

What I would recommend to resolve the issue flagged within this report would be to implement a check for the “sm\_list\_next” variable to ensure that it remains a safe value that is within the programs intended buffer. Doing this would implement a safety net for the variable to ensure that it does not reach a value that would cause the program to crash whether the value is achieved accidently by a bug or intentionally by an adversary.

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.