**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** |
| 9/08/2023 | V0.1 | Anthony Scantsonihas | Beginning of investigation |
| 10/08/2023 | V0.3 | Anthony Scantsonihas | Investigation |
| 12/08/2023 | V0.5 | Anthony Scantsonihas | Investigation |
| 13/08/2023 | 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 ***Ignoring number of bytes read*** type defect identified in the following

CIDs:  
***CID1520836***

# 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

Coverity has flagged an issue with the CID 1520836 which is found within the code base libbsP.c230. The error that has been flagged within this block of code is an Ignoring number of bytes error which is described under CWE-252. This issue is caused when the number of bytes copied into the buffer is potentially smaller than the requested number which in turn can cause the buffer to be potentially be accessed out of range.

## Observations

The flagged error appears to be caused by the “loadTblIndex” function. The overall purpose of the function is to facilitate the process of loading and interpreting data from the table index of a binary file to allow other components of the code to operate with the data contained within the file.

A screenshot of a computer code

Description automatically generated

The issue originates from an if statement located within the function as it returns the number of bytes read, however, it is ignored.

**-Potential Vulnerabilities**

The number bytes error with the “CHECKED\_RETURN” variable could potentially provide adversaries with vulnerabilities to exploit as ignoring the return value of functions like “lseek” and “read” has the potential to cause security implications. This issue could potentially be exploited to cause information leakage as well as controlled crashes to conduct a denial-of-service attack.

## 

## Supporting Evidence

Evidence of this error can be seen within the screenshot below as it displays the error flag. As can be seen the flag displays how “check\_return” returns the number of bytes read, and despite this it is still ignored by the program.

A close-up of a sign

Description automatically generated

# Conclusions and Recommendations

What I would recommend to resolve these vulnerabilities, would be to be to handle the return values of file operations by introducing proper error handling that checks the return values and handles any errors that may occur with the process. By doing this the chance of this threat occurring will be significantly reduced as the process will be far more difficult for any attackers to exploit.

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.