**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** |
| 5/05/23 | V1.0 | Damon Willmott | Write Up |
| 7/05/23 | V1.1 | Damon Willmott | Completed |

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:  
***CID 1520728***

# 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 |
| BPNM | Bundle Protocol Naming Module |
| PSM | Personal Space Memory |

# Code Review and Analysis

## Overview

Coverity identifies CID-1520728 as a **High Impact** quality vulnerability, and is labelled as an **Out of Bounds** access error, also known as a buffer overflow error. This implies the that the code is trying to access memory outside of an allocated buffer. This usually occurs when the data is somehow given the ability to read and write data from a memory location outside of its intended boundaries/buffer.

## Observations

The issue can be found in the *bpnm\_endpointNames\_get* function in the /bpv7/library/libbpnm.c file directory of Bundle Protocol 7. Whilst there is little to no documentation indicating what this function does, it can be deduced it might play a role within NASA ION’s Bundle Protocol Naming Module (BPNM), where it may retrieve the names of endpoints associated with the BPNM, which could be related to the network addresses or identifies used for communication and routing within the Bundle Protocol Network. As with many of the other out-of-bounds access errors in Bundle Protocol 7, a value is being passed without being properly validated, and in this case the value being passed (18446744073709551615) is larger than the specifications of the buffer and is too large to be handled by an element index, which is causing the overflow of the allocated memory.  
  
This CID error is very similar to other such Out of Bounds access errors identified within Bundle Protocol 7. Within the validation of the code, the valule elt (which is parsed to the element index schemeElt in this particular CID) refers to the PSMAddress, which specifies the personal space memory that is defined within the code, and elt is used to specify the specific memory that is being addressed. Within the function sm\_list\_first there is a value called eltbuffer, which utilises the parameters elt and partition, which is what validates whether the value is within the defined limits of the buffer. However, because the value of elt is so large (18446744073709551615), the validation error occurs because the program is trying to access memory outside the allocated buffer, which is specified in sm\_list\_data, causing the out-of-bounds access error

## Supporting Evidence

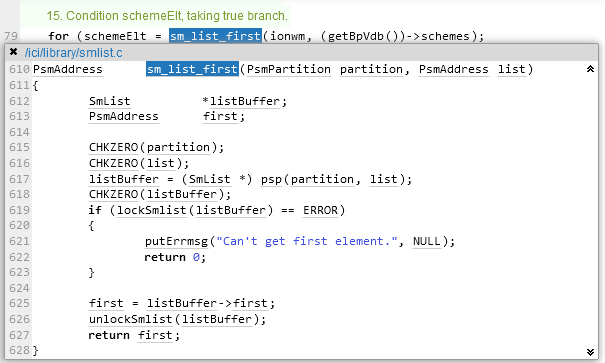
Fig. 1. Coverity static analysis results

Fig. 2. sm\_list\_first function

# Conclusions and Recommendations

See SAR\_1520702 by Damon Willmott

References

Common Weakness Enumeration (n.d.) *CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer*, <https://cwe.mitre.org/data/definitions/119.html>

Appendix

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