**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** |
| 6/04/2023 | V0.1 | Moe Khant Kyaw | Initial Document and analysis |
| 9/04/2023 | V0.2 | Moe Khant Kyaw | Finalizing Investigation and documentation |

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:  
***1520889***

# 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

CID 1520899 has been marked as high impact issue, Out-of-bound access type under Memory Corruption category by Coverity. Out-of-bound access error, which is a type of buffer overflow, occurs when accessing memory outside of an allocated buffer, i.e., when the program or function tries to read or write beyond the expected limit of the buffer.

## 

## Observations

This issue is found in “sdr\_reload\_profile” function under “/ici/sdr/sdrxn.c” filepath. At line 1473, sm\_list\_first function is called using elt, which is unassigned, as parameter.

Graphical user interface, text, application

Description automatically generated

*Figure 1: unassigned elt*

The function, sm\_list\_next, takes two parameters, partition and elt, and then returns the next element of the list. It checks if partition and elt are valid by using “CHKERR” macro. This function resulted in a large value of “18446744073709551615” being passed to elt. There’s a possibility that the value of elt may exceed the bounds of the linked list, which could lead to an out-of-bounds access.

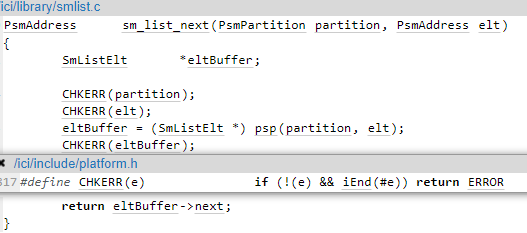


Figure 2: sm\_list\_next function

Because of a very large vale of elt, the program is trying to access memory outside of the allocated buffer, specifically in sm\_list\_data function, thus causing out-of-bound access error.

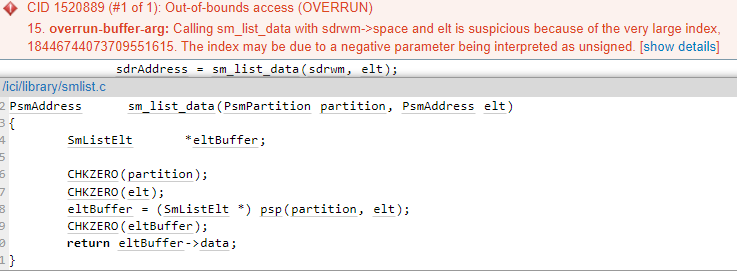
**

Figure 3: Out-of-bounds access

## Supporting Evidence

Please provide any supporting evidence, and feel free to make references to documents in the appendix.

# Conclusions and Recommendations

To fix this issue, it is recommended to initialize the value of elt to something within the boundary of the allocated buffer before using it. It would be better if sm\_list\_next and sm\_list\_data functions can handle parameters with unassigned value.

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.