**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** |
| 8/8/2023 | V0.1 | Tahlea Grant | Initial report start |
| 9/8/2023 | V0.1 | Tahlea Grant | Report completion |
|  |  |  |  |

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 ***Control Flow Issue*** *type* defect identified in the following CIDs:

***CID-1520870***

# 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

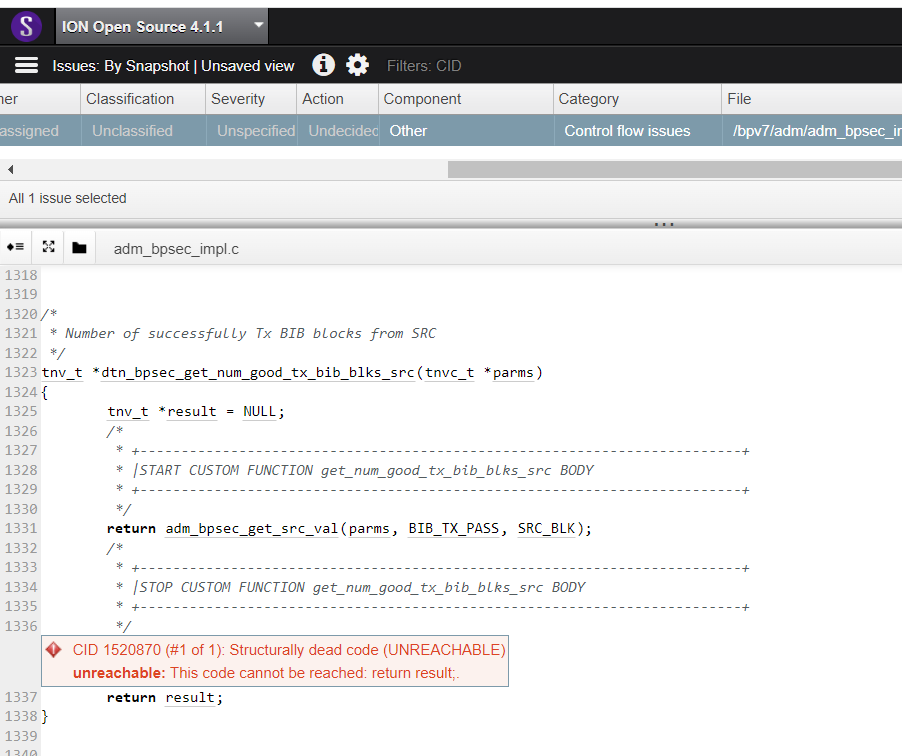
Coverity identifies CID-1520870 as a **Medium Impact** quality vulnerability, and is labelled as an **Structurally dead code** error. This means that the code is not accessible by the compliler after a certain point due to an incorrect statement which impacts the flow of code. This usually occurs when the code has a return, if, else, while statement in a position which prevents the code below it from ever being accessed (eg the if statement variable is always true but there is still code after it).

## Observations

## This issue can be found at /bpv7/adm/adm\_bpsec\_impl.c and within the function dtn\_bpsec\_get\_num\_good\_tx\_bib\_blks\_src. Within this function a value is to be returned (in this case the value of adm\_bpsec\_get\_src\_val(), and then is also calling for a undefined value ‘result’ to be returned, however this is unreachable. This is an issue as it uses processor memory unnecessarily.

## The error begins in line 1331 when there is a return of a function and then after in line 1337 another return for result which is unreachable due to the earlier return.

## Supporting Evidence



# 

# Conclusions and Recommendations

In conclusion, this could have negative effects on the whole program if result is called upon else where in the code. As a result, I recommend that line 1331 is modified to read “result = adm\_bpsec\_get\_src\_val(parms, BIB\_TX\_PASS, SRC\_BLK) ” and line 1337 stands as is. Because this would improve the program’s consistency, continuity and speed as well as increase the readability of the code and rectify the error.

References  
Bauch, A. (2021). Debugging poor performing or unreachable code. Embedded. <https://www.embedded.com/debugging-poor-performing-or-unreachable-code/#:~:text=Unreachable%20code%20is%20generally%20considered,code%20that%20is%20never%20used>

Appendix

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