**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** |
| 28/08/2023 | V0.1 | Anthony Scantsonihas | Beginning of investigation |
| 30/08/2023 | V0.2 | Anthony Scantsonihas | Researched the flagged code |
| 01/09/2023 | V0.3 | Anthony Scantsonihas | Began report write up |
| 08/09/2023 | V1.0 | Anthony Scantsonihas | Finished report write up |

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 ***Strucurally dead code*** type defect identified in the following CIDs:  
***CID 1520626***

# 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 system has flagged issues within the code base adm\_bpsec\_impl.c:1442. In the case of CID 1520626 the error that has been flagged is a Structurally dead code error which can be seen described under CWE-561. Essentially, the issue indicates that the code being flagged contains ‘dead code’ which can never be executed therefore rendering the code as irrelevant to the overall program.

## Observations

The Coverity flag has identified a multitude of errors all stemming from different functions within the snippet of code. The purpose of these functions is to retrieve statistics derived from “BPsec” operations, these statistics include the number of successfully transmitted and received blocks, the number of failed blocks as well as additional metrics that are typically used for monitoring and debugging the BPsec system’s performance. Several lines of the code snippet are flagged with “structurally dead code” meaning that those lines contain unreachable statements and will never be executed. The majority of these issues originate from ‘return’ and ‘result’ statements at the end of the different functions.

A screenshot of a computer program

Description automatically generated

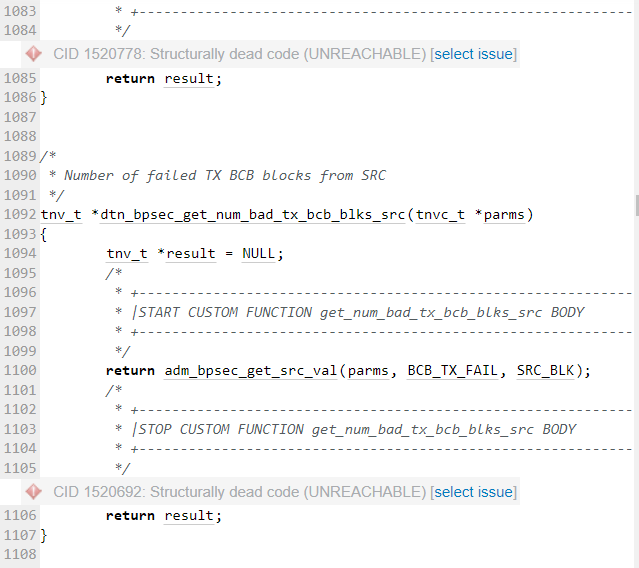
**-Potential Vulnerabilities**

Overall, from what can be seen from the flagged code snippet it is unlikely these errors alone will provide adversaries with the potential vulnerabilities to attack and exploit the larger program.

## 

## Supporting Evidence

Evidence of these issues can be seen within the screenshot below as there are multiple functions flagged with the structurally dead code error.



# Conclusions and Recommendations

In conclusion of this investigation, the recommendation I would provide to resolve this issue would to remove all the unreachable code found within this code snippet. As this component of the program appears to be redundant and doing so should not effect the overall function of the application.

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. (2006, July, 19). CWE - CWE-561: Dead Code. from

<https://cwe.mitre.org/data/definitions/561.html>

Appendix

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