**T2** 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** |
| 13/09/2023 | V0.1 | LIANG CHEN | Analyze Improper Input Validation |
|  |  |  |  |
|  |  |  |  |

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 5](#_Toc119848732)

[References 6](#_Toc119848733)

[Appendix 7](#_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 ***Improper Input Validation*** type defect identified in the following CIDs:  
***1520855***

# 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

## This document delves into the analysis of the identified issues within the bsl\_sdr\_bootstrap function located in the bpsec\_policy.c file, and its associated functions in ion.c and bpsec\_policy\_eventset.c. The primary concern revolves around the handling of untrusted or "tainted" data, which could potentially lead to security vulnerabilities.

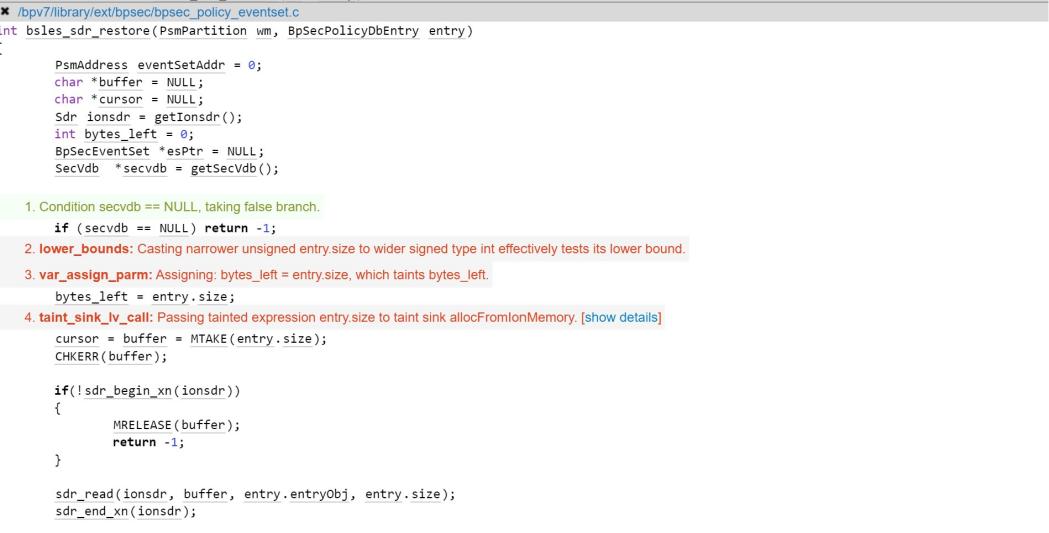
## Observations

## The main issues identified in the provided code snippets are related to the potential mishandling of tainted data. This tainted data, if not properly validated, could lead to unintended behaviors or security vulnerabilities within the application.

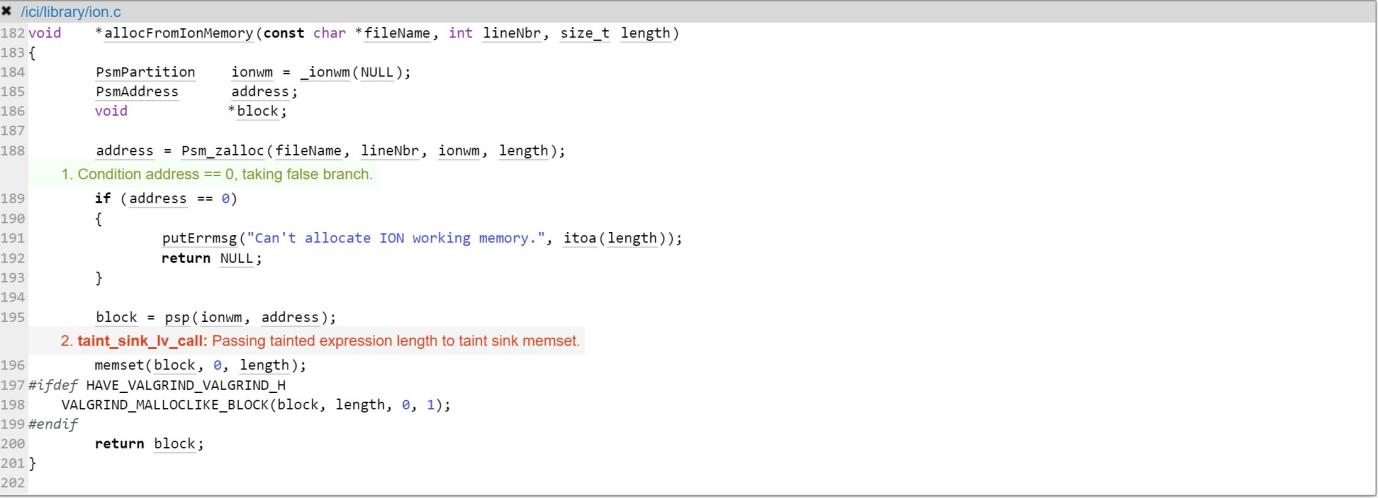
## Supporting Evidence



In the bsl\_sdr\_bootstrap function from bpsec\_policy.c, the entry variable receives data from an external source (possibly the SDR). This data is then passed to the bsles\_sdr\_restore and bslpol\_sdr\_rule\_restore functions without adequate validation.



Within the bsles\_sdr\_restore function in bpsec\_policy\_eventset.c, the entry.size attribute, which is derived from the aforementioned tainted entry, is used to allocate memory. This allocation is done using the allocFromIonMemory function found in ion.c.



The allocFromIonMemory function in ion.c uses the potentially tainted length parameter, which is directly related to the entry.size from the previous function, to allocate and initialize memory.

# Conclusions and Recommendations

The primary concern across the three files is the potential mishandling of untrusted data. This data, if not properly sanitized, could lead to vulnerabilities, especially if it influences memory operations or other critical functionalities.

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).

7 Pernicious Kingdoms (CWE Draft 3, 2006-07-19, Submitted on July 19, 2006).

<https://cwe.mitre.org/data/definitions/20.html>

Appendix

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