**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** |
| 01 Apr 2023 | V0.1 | Connie Cox | Initial draft. |
|  |  |  |  |
|  |  |  |  |

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 ***Memory – corruptions (Out of bounds access)*** type defect identified in the following CIDs:

**1520873**

**1520883**

# 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 |
| PSM | Personal Space Memory |

# Code Review and Analysis

## Overview

Please provide an overview on the analysis performed here.

The Coverity report identified several Out-of-bounds access issues with the bpsec\_policy\_rule.c within the Bundle Protocol v7 library extensions for bundle protocol security [/bpv7/library/ext/bpsec] directory. The following analysis outlines the issues identified in the bslpol\_rule\_get\_addr and bslpol\_rule\_get\_all\_match functions.

Both functions use doubly linked lists as a buffer to hold the memory address of an element. Each element represents a personal space memory addresses (PSM). In order to retrieve the next element after having already retrieved the first element, a function call to sm\_list\_next() is required.

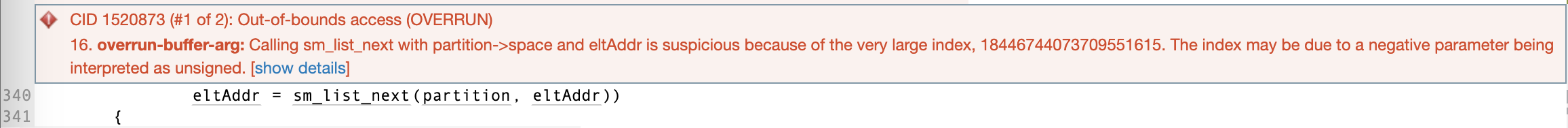
## Observations

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

The bpsec\_policy\_rule file implements processing specific to BPSec policy rules which are used by the policy engine to associate security actions with security events. The two functions underpinning the CIDs are bslpol\_rule\_get\_add() and bslpol\_rule\_get\_all\_match().

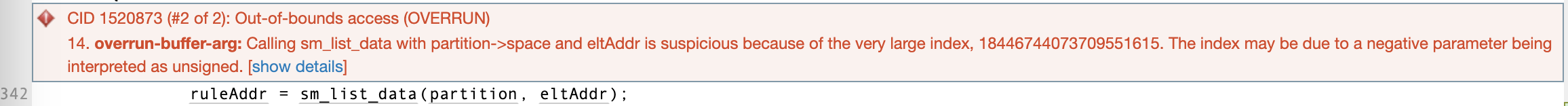
**FUNCTION: bslpol\_rule\_get\_add()**

The bslpol\_rule\_get\_add() function returns the policy rule address associated with a unique user id. The rules only exist in shared memory, therefore the pointer returned by this function is only valid in the context of the process calling this function. There are two issues being highlighted in this function. The first is attributed to the following code.



In the above code (line 340), the element address denoted by eltAddr is of data type PsmAddress and is expected to be returned on assignment to sm\_list\_next(). However, on closer inspection, if the partition that is passed into the bslpol\_rule\_get\_addr() is null, and/or the pointer to the bpsecPolicyRules is null, the function is likely to be querying an invalid list and thus a return a large number as an error.

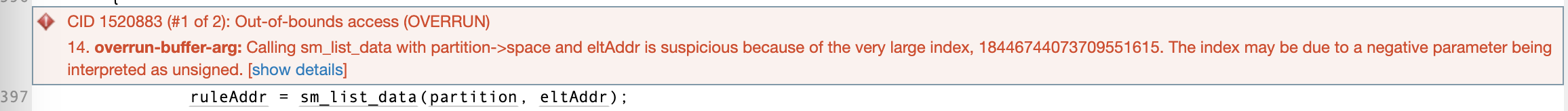
The second issue is attributed to the return value of sm\_list\_data().



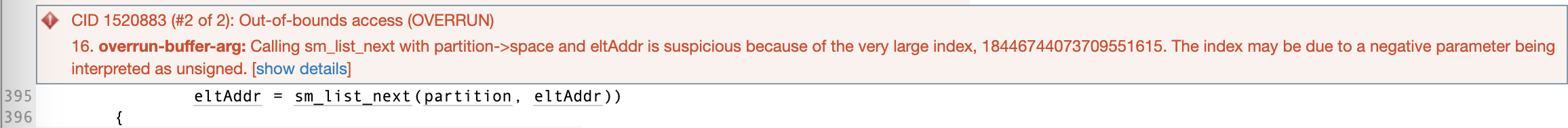
As with the sm\_list\_next(), the sm\_list\_data() (line342) performs a similar function but rather than returning the pointer to the element address, it returns the pointer of the rule address. As with sm\_list\_next(), if the partition passed into sm\_list\_data() is null, and/or the value assigned to eltAddr is invalid or null, sm\_list\_data() is likely to be an invalid list and thus, return a large number as an error.

**FUNCTION: bslpol\_rule\_get\_all\_match()**

The bslpol\_rule\_get\_all\_match() function returns a list of all the rules known by the policy engine matching the set of provided criteria. The list returned contains pointers to rules in shared memory and only valid in the context of the calling process. There are two issues being highlighted with this function. The first is regarding the following code (line 397):



The second relates to this code (line 395):



In both sets of code, there is an issue regarding the value of eltAddr being of large number. Whilst eltAddr was initialized to zero, it was assigned a new value line 393 to be the PsmAddress of the first element in the linked list. Unfortunately, when a linked list is invalid, or when a parameter to retrieve the first element is negative, the value returned by sm\_list\_first() is a large number which could be a representation of an error.

## Supporting Evidence

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

# Conclusions and Recommendations

Please provide any conclusions and recommendations here. Include references to supporting evidence and any other required information in the appendix.

When elements are used with linked lists, it would be prudent to ensure that the element exists before they are added to the linked list. This would ensure that the linked list is valid and that there isn’t a buffer overrun on assignment to PsmAddresses.

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.