**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** |
| 19/09/2023 | V1.0 | Damon Willmott | Initial document |
| 24/09/2023 | V1.1 | Damon Willmott | Finalisation |
|  |  |  |  |

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 ***Unchecked return value*** type defect identified in the following CIDs:  
***CID 1520777***

# 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

Static analysis by Coverity categorieses CID-1520777 as a **Medium** impactvulnerability, and depicts it as an **Unchecked return value** error. This issue is described in CWE-252 as one where the product does not check the return value from a method/function, preventing it from detecting unexpected states and conditions. In this particular case it is indicating that a value that has been returned from a function is not being checked for any errors before being used. The purpose of this code is to send a bundle status report with the provided reason code.

## Observations

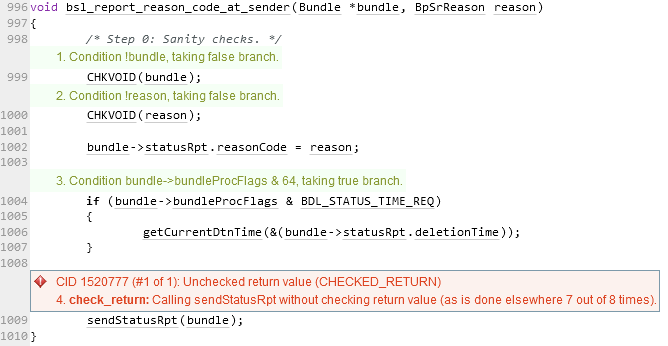
The issue can be found in the bsl\_report\_reason\_code\_at\_sender function within the /bpv7/library/ext/bpsec/bpsec\_policy.c file directory. On line 1009, Coverity highlights a warning within the sendStatusRpt function, which suggests that the return value of this function has not been checked for errors.

It appears that there is no validation included to check or validation on whether the value returned from sendStatusRpt has any errors, which could result in unexpected behaviours and outcomes from the code.

The error flagged from Coverity also indicates that there are 7 out of 8 similar instances of this particular function where correct handling of the return value has occured, meaning that this is the first instance of this particular function where it has not been correctly validate or checked.

As a result of not checking the value returned from sendStatusRpt, unexpected behaviours and outcomes can potentially occur for other areas of the porgram

## Supporting Evidence

The following highlights the issue within, the code. On line 1009, sendStatusRpt is called, however the value that is receives is not validated, hence the error is not properly handled by this instance of code, and the outcome could cause potential problems in the program.

*\* See also SAR\_1520671 and SAR\_1520831 by Damon Willmott*

# Conclusions and Recommendations

This issue is a simple bug in the code, and as such the likelihood of exploitation and severity of the issue is low. However, having an unexpected return value could possibly cause the system/program to crash or cause other unintended consequences, such as exposing the system to denial-of-service (DOS) attacks, so it is an important defect to remedy.

One possible recommendation to resolve this issue would be to implement measures to ensure that there are checks in place for all functions that return a value that ensure that the value received is verified and expected. Another recommendation could be to create an exception case that occurs in the event that the value returned is unexpected to ensure the program acts in an expected manner, even if the values received are not typical or expected.

References  
Common Weakness Enumeration (n.d.) *CWE-252:* *Unchecked Return Value*, <https://cwe.mitre.org/data/definitions/252.html>

Appendix

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