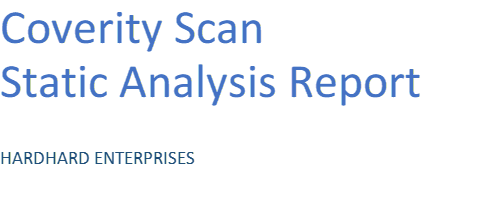
​



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** |
| 16/12/2022 | V0.1 | YIZHOU FENG/YONGXIANG HUANG | Initial document |
| 18/12/2022 | V0.2 | YIZHOU FENG/YONGXIANG HAUNG | Update information and conclusions |
| 18/12/2022 | V0.3 | YIZHOU FENG/YONGXIANG HUANG | Final fix |

Table of Content

Contents

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 ***USE OF 32-BIT TIME\_T*** type defect identified in the following CIDs: 1520853

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 |

Code Review and Analysis

Outcomes

When performing static code analysis using the ION Open Source 4.1.1 dashboard for CID 1520853, this vulnerability is flagged as a high severity vulnerability and we believe the presence of this vulnerability poses a high risk to all code. Also, the presence of the return\_constant flaw and the overrun-buffer-arg.

Observations

This code appears to be checking for congestion in a network and attempting to forecast the state of the network at some point in the future. The variables **forecastTime**, **iondb.horizon**, **change->time**, **netGrowth**, and **iondb.occupancyCeiling** are all used in this process.

The code begins by checking if the time of the current change in the network (stored in **change->time**) is less than the time at which the forecast is being made (**forecastTime**). If it is, an error message is printed and the code continues to the next iteration of the loop.

Next, the code checks if the **iondb.horizon** value is set and, if it is, whether it is less than **change->time**. If both of these conditions are true, the **secInEpoch** variable is set to the difference between **iondb.horizon** and **forecastTime**. Otherwise, **secInEpoch** is set to the difference between **change->time** and **forecastTime**.

The code then checks the value of **netGrowth**. If it is greater than 0, this indicates that the network is receiving more data than it is transmitting. In this case, the code calculates the number of seconds until the network's occupancy (the amount of data being stored in the network) reaches the ceiling defined by **iondb.occupancyCeiling**. This is done by dividing the amount of space remaining before the occupancy ceiling is reached by the net growth rate, and adding 0.5 to the result. If **secInEpoch** is less than this value, **secAdvanced** is set to **secInEpoch**, otherwise it is set to the calculated number of seconds until the occupancy ceiling is reached.

If **netGrowth** is less than or equal to 0, this means that the network is either transmitting more data than it is receiving, or that the amount of data being transmitted and received is equal. In this case, **secAdvanced** is simply set to **secInEpoch**.

For the Use of 32-bit time\_t (Y2K38\_SAFETY)

**Use of 32-bit time\_t (Y2K38\_SAFETY)** is a warning message generated by static analysis tools that are designed to check for potential issues in code. It indicates that the code is using a 32-bit integer data type to store a value of type **time\_t**, which is a data type used to represent time values in many programming languages.

One potential issue with using a 32-bit integer to store **time\_t** values is that it may not be large enough to accurately represent all possible time values. This can be particularly relevant when considering the year 2038 problem, also known as the Y2K38 issue. This problem arises due to the fact that many systems store time values as a signed 32-bit integer, which can represent values in the range of -2147483648 to 2147483647. This means that the largest possible value that can be represented using a 32-bit **time\_t** value is January 19, 2038 at 3:14:07 AM. After this point, the value will wrap around and become negative, causing problems for systems that rely on accurate time values.

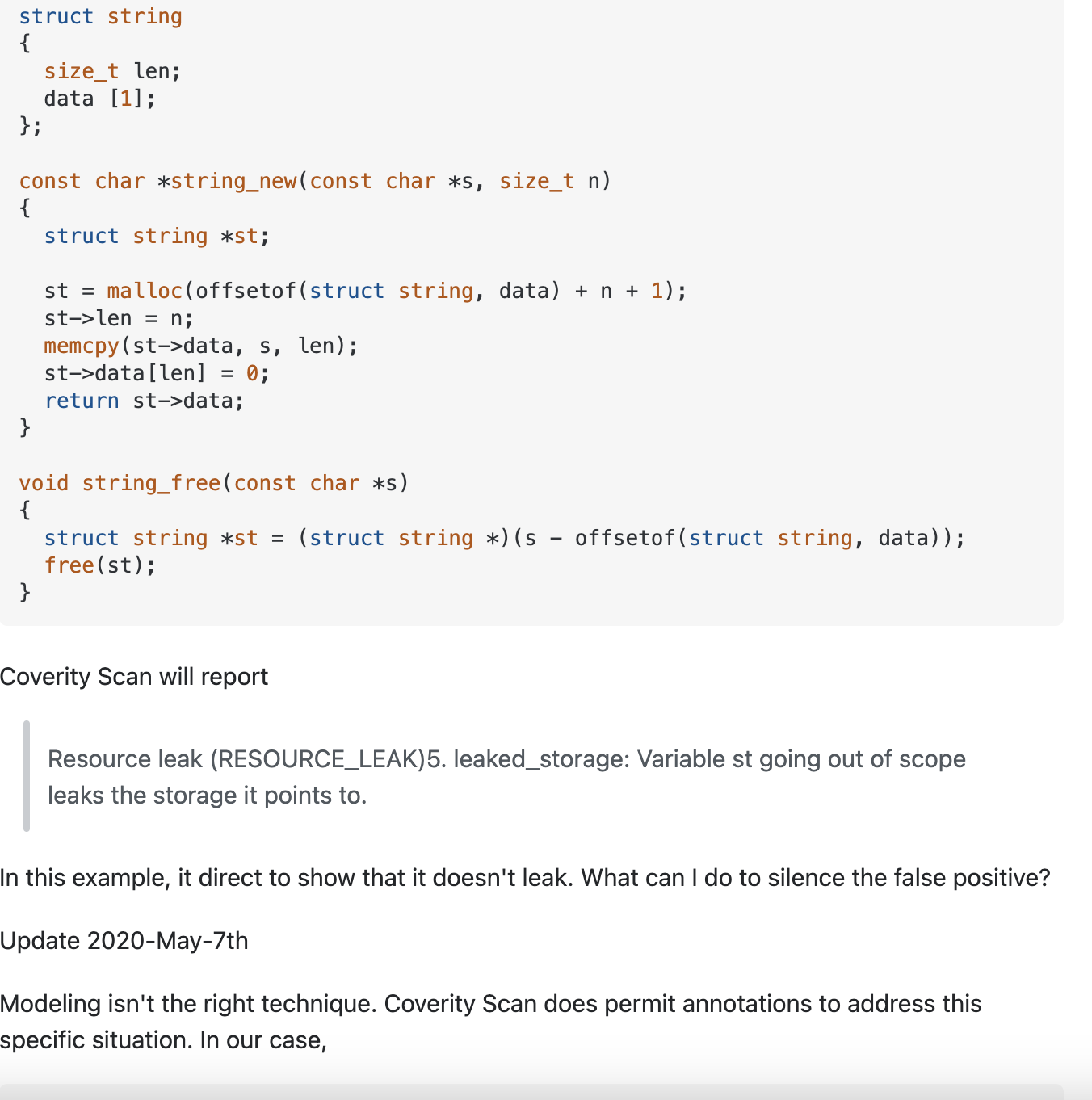
The **Use of 32-bit time\_t (Y2K38\_SAFETY)** warning is intended to alert developers to the potential for issues related to the year 2038 problem, and to encourage them to consider using a larger data type to store **time\_t** values in order to avoid these issues.

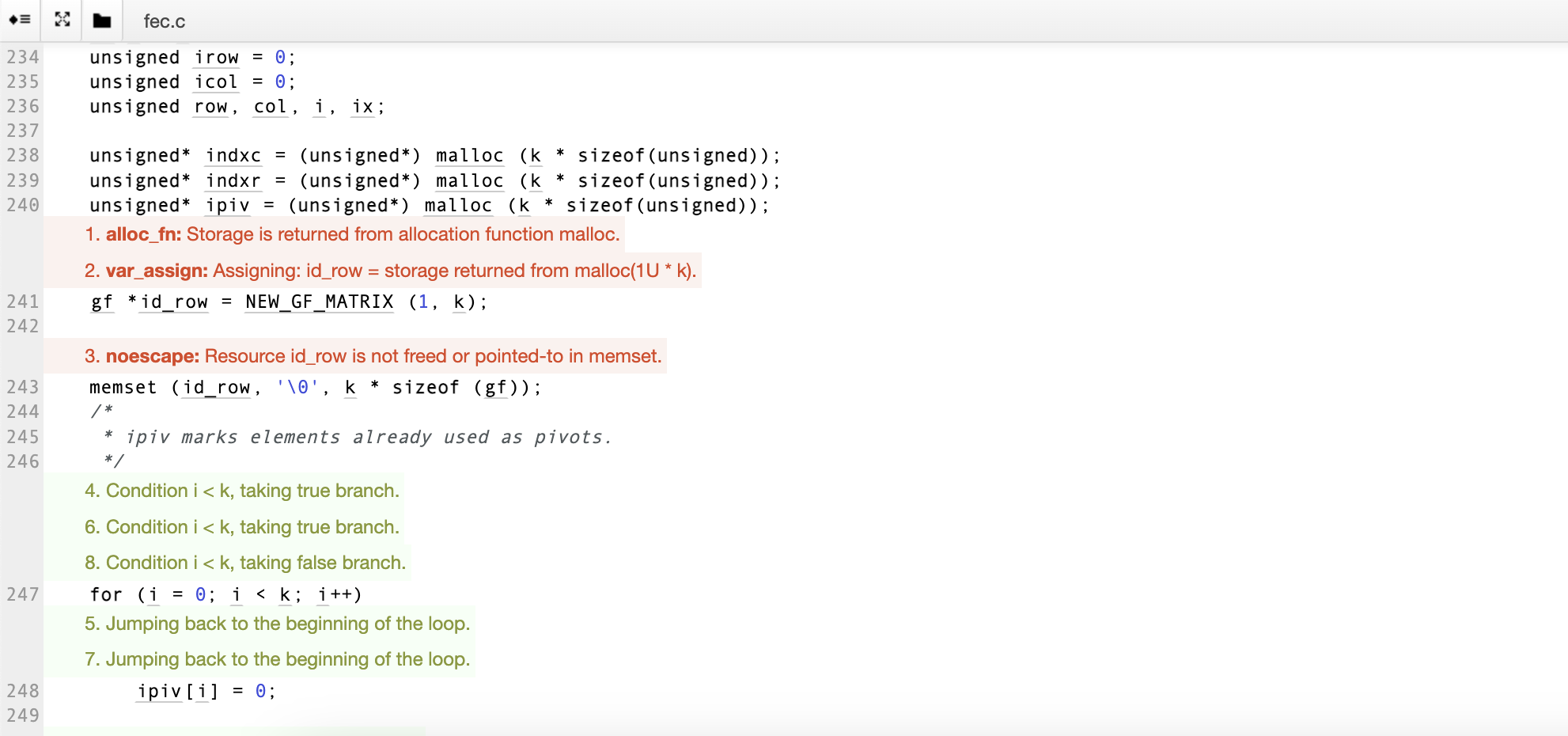
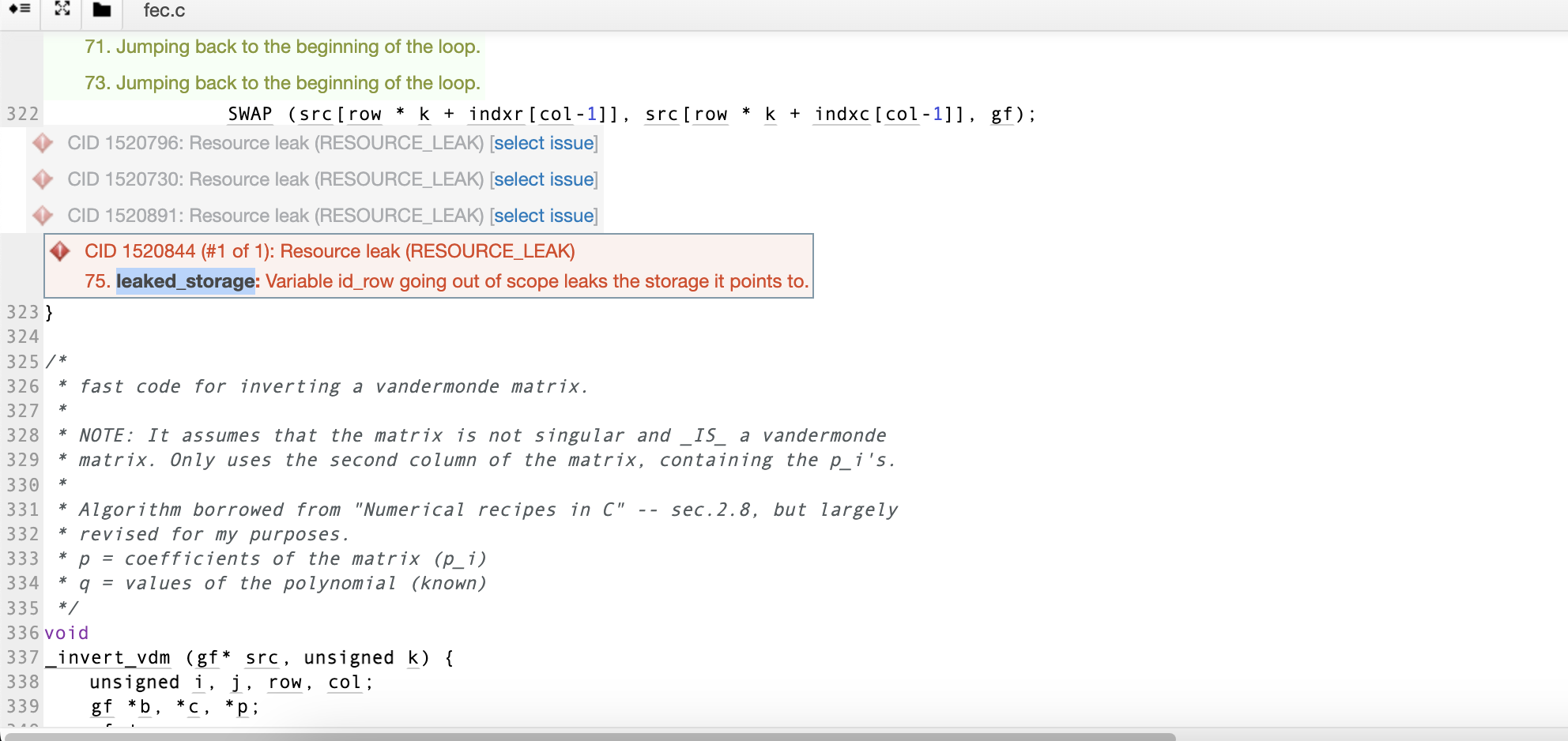
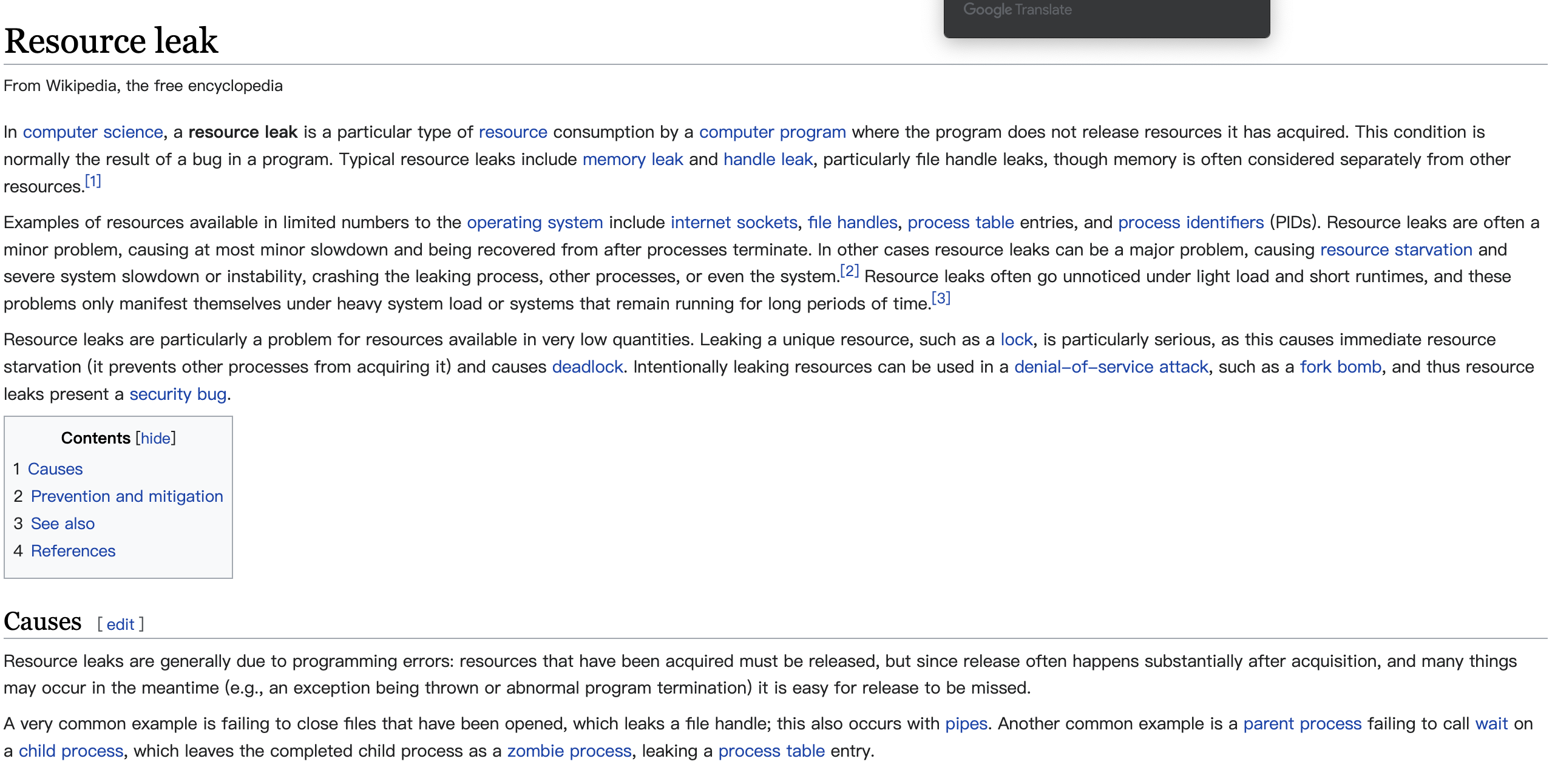
For the 1. **store\_truncates\_time\_t:** A time\_t value is stored in an integer with too few bits to accommodate it. The expression change->time is cast to unsigned int.

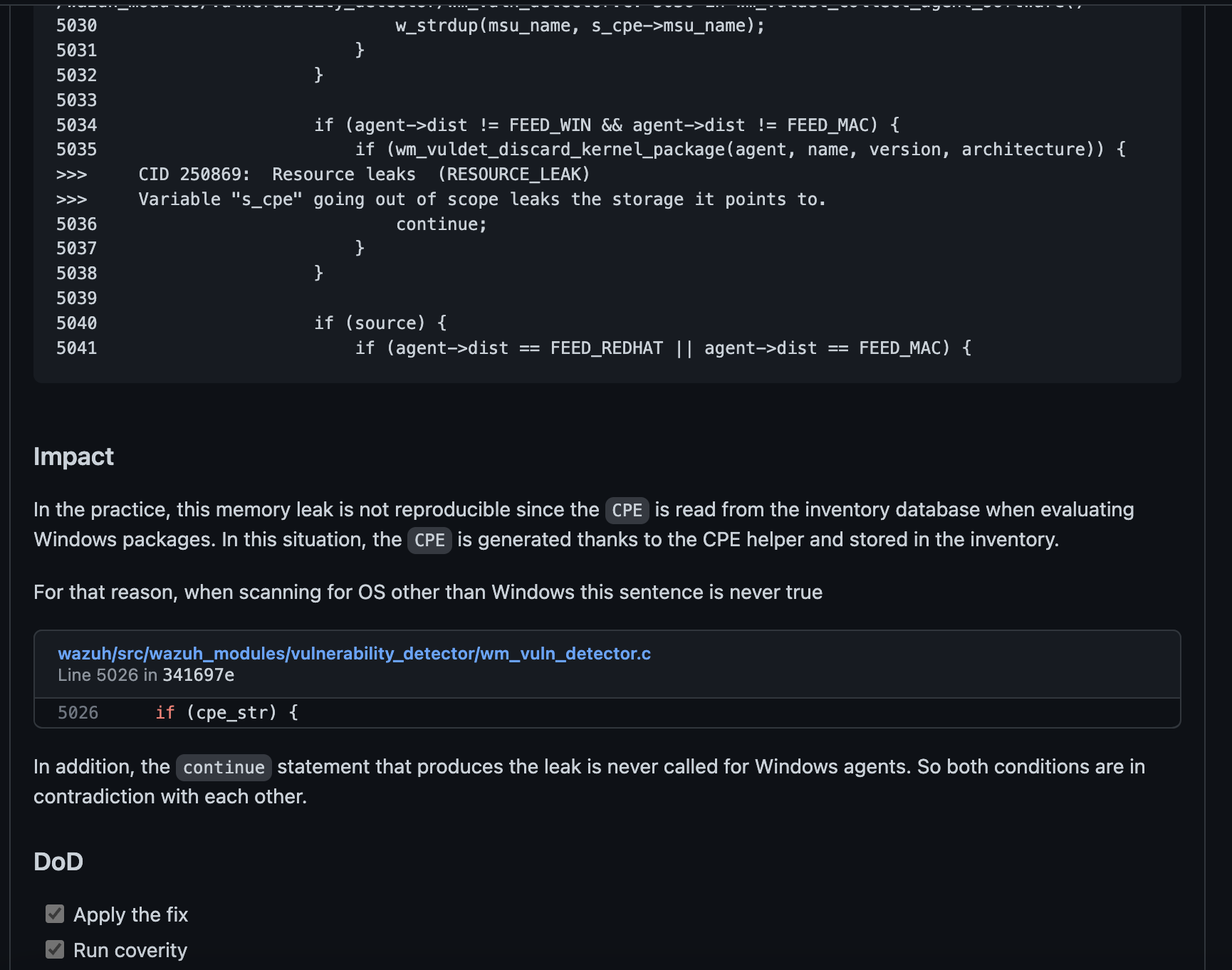
**store\_truncates\_time\_t** is a warning message generated by static analysis tools that are designed to check for potential issues in code. It indicates that a value of type **time\_t**, which is a data type used to represent time values in many programming languages, is being stored in a data type that is not large enough to accurately represent all possible values of **time\_t**. This can result in the value being truncated or lost when it is stored, potentially leading to issues or errors in the code.

It is important to ensure that **time\_t** values are stored in a data type that is large enough to accurately represent the full range of possible values. In many systems, this means using a 64-bit integer data type to store **time\_t** values.

Supporting Evidence



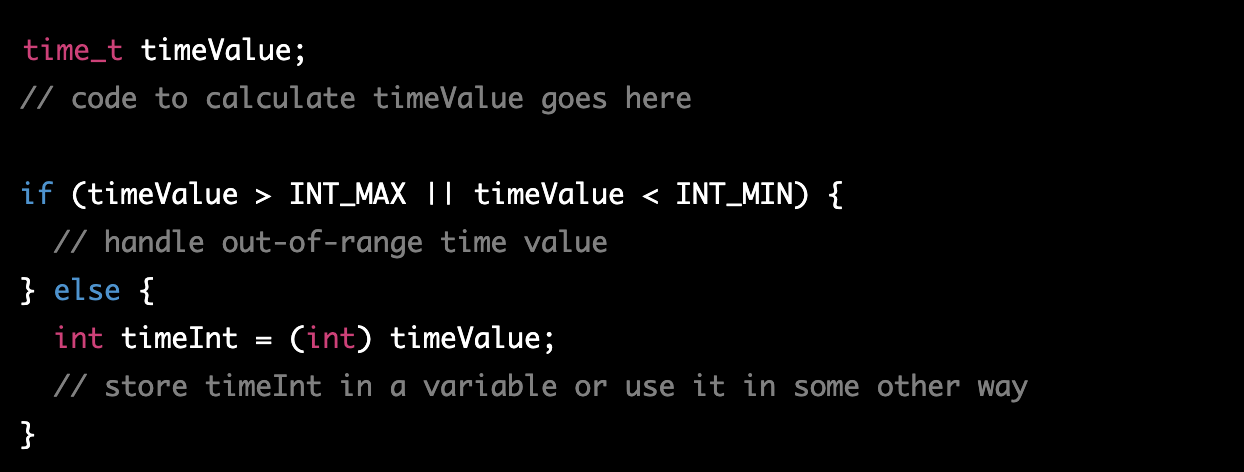
  



Conclusions and Recommendations

Here are a couple of options for fixing the **store\_truncates\_time\_t** warning:

1. Use a larger data type to store **time\_t** values: One way to fix this warning is to store **time\_t** values in a data type that is large enough to accurately represent the full range of possible values. In many systems, this means using a 64-bit integer data type to store **time\_t** values. For example, you could change the data type of the variable in which the **time\_t** value is being stored from **int** to **long long int** or **int64\_t**.
2. Check the range of **time\_t** values before storing them: If it is not possible or practical to use a larger data type to store **time\_t** values, you can add a check to ensure that the value being stored is within the range of the data type you are using. If the value is outside of this range, you can take appropriate action, such as returning an error or handling the value in some other way. For example:



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.*