



## **Software and Computer Security**

**SOFE 4848U**

### **Phase 3 - Final Report**

**Date: March 27, 2023**

**Project Group: Group 17**

| <b>Student Name</b> | <b>Student Number</b> |
|---------------------|-----------------------|
| Carson McClelland   | 100725653             |
| Shwan Majeed        | 100749077             |
| Walid Ayub          | 100695612             |
| Kalid Ajibade       | 100660188             |

# Abstract

In recent years, it has become evident that USBs have become an important tool for the transfer of data and information storage, but they can also pose a high-security risk to their users. The main focus for its security would be to have a USB scanning application. In today's age of technology, the importance of this application would be in its features to identify and prevent any potential threats, which could include any sort of malware, viruses, and corrupted files that can breach the user's information or any other party that is involved. It should work like any scanning program would, by having it identify all infected files and provide the details of its scan to the user. The methodology behind it would require the use of various tools such as python and ClamAV engine. There is a heavy emphasis on the importance of this USB scanner to protect the data from any security risks along with why software and computer security is critical in ensuring the safety of a user's information.

# Table of Contents

|                                       |           |
|---------------------------------------|-----------|
| <b>Abstract</b>                       | <b>2</b>  |
| <b>Table of Contents</b>              | <b>3</b>  |
| <b>List of Illustrations</b>          | <b>4</b>  |
| <b>Introduction and Background</b>    | <b>5</b>  |
| <b>Project Objective</b>              | <b>6</b>  |
| <b>Project Methodology</b>            | <b>7</b>  |
| <b>Results</b>                        | <b>11</b> |
| <b>Conclusion and Recommendations</b> | <b>14</b> |
| <b>References</b>                     | <b>15</b> |
| <b>Contribution Matrix</b>            | <b>16</b> |

# List of Illustrations

|  |    |
|--|----|
| Figure 1: UML Sequence diagram               | 7  |
| Figure 2: Waterfall Diagram                  | 8  |
| Figure 3: UML Class diagram                  | 9  |
| Figure 4: ClamAV daemon                      | 11 |
| Figure 5: ClamAv daemon status               | 11 |
| Figure 6: Pyclamd initialization             | 12 |
| Figure 7: Scan directory function            | 12 |
| Figure 8: Subdirectory scanning              | 13 |
| Figure 9: Running program and showing output | 13 |

# Introduction and Background

In this age of technology, USBs have become an essential tool for data transfer and storage. Even though this is such a crucial tool for us to use with what it offers, USB keys can also be a problem with the number of security risks to a user. This is where our USB scanner can be an essential application. It can be a powerful tool that allows a user to check for any potential security threats on the USBs. Users can use a USB scanner application to verify USB keys for any potential security risks. As long as it is still linked to the same wifi network after being connected to a computer, the application can scan for and identify any problems. The app ought to have the ability to search for any potential malicious applications. By having a user interface that enables users to know what they are doing or makes it easy for them to figure it out by providing instructions, this application will be both aesthetically pleasing for complete customer satisfaction and simple to use for all users. The scanner app ought to have features like scanning USB keys, rearranging data from the scan, having a settings menu, and possibly sharing the data. This app will help with the issues many users have with potential hazards to their USBs and can eliminate these hazards.

# Project Objective

An efficient way of storing data and transporting files is to use a USB key. These keys often contain sensitive information where users expect the data to be kept safely. The objective of this project is to create an application that will ensure the safety of the data and reduce the chance of possible risks that can corrupt the files or breach the user's information. By creating an application using python and an antivirus engine, ClamAV, all of the contents of the USB will be scanned and assessed for potential malware.

There are a few criterias that the application must have in order to process the project objective. The application must effectively detect all files within the USB key that are infected and move them to a quarantine directory to isolate the files from additional harm. This ensures that all of the corrupted files are known by the user and they are now isolated from the rest of the files so that they can't further harm any additional files within the USB. The application should also be able to scan and detect all of the files within the USB plugged into the machine within a reasonable amount of time. This is because the user doesn't want to wait a long time to access the files, and the application should be able to detect the files quickly to eliminate the possibility of the malware transferring from the USB to the user's machine. Additionally, the application will need to provide a detailed report to the user regarding the USB health, available space, space used, number of corrupted files, and any potential risks. This detailed report will help the user understand the contents of the USB that they inserted into their machine.

With the help of our application, the objective of creating a USB scanner that can identify malware, protect against data theft, and prevent system damage will be accomplished. Upon the completion of this objective, the user will be able to safely use the contents within the USB key.

# Project Methodology

The way we went about approaching this problem is by carefully selecting our tools and then implementing those tools congruently to reach our final result of a safe and secure usb drive. The tools involved in this project are Linux, Python, Clamav- daemon, Pyclamd and EICAR test. We chose these tools because we felt that each of them served a demand that we have and have properties that when paired together complement each other.

Figure 1: UML Sequence diagram

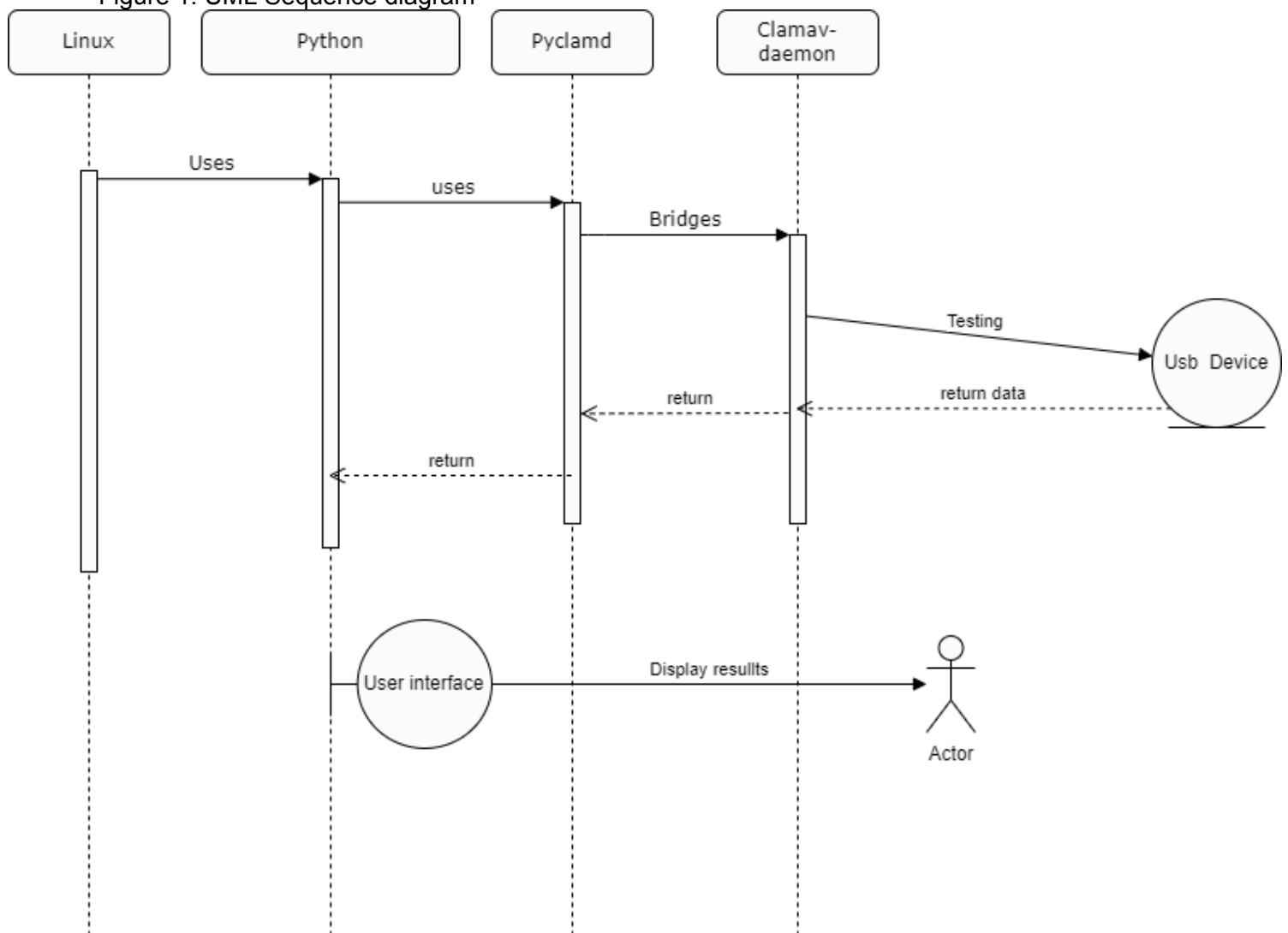
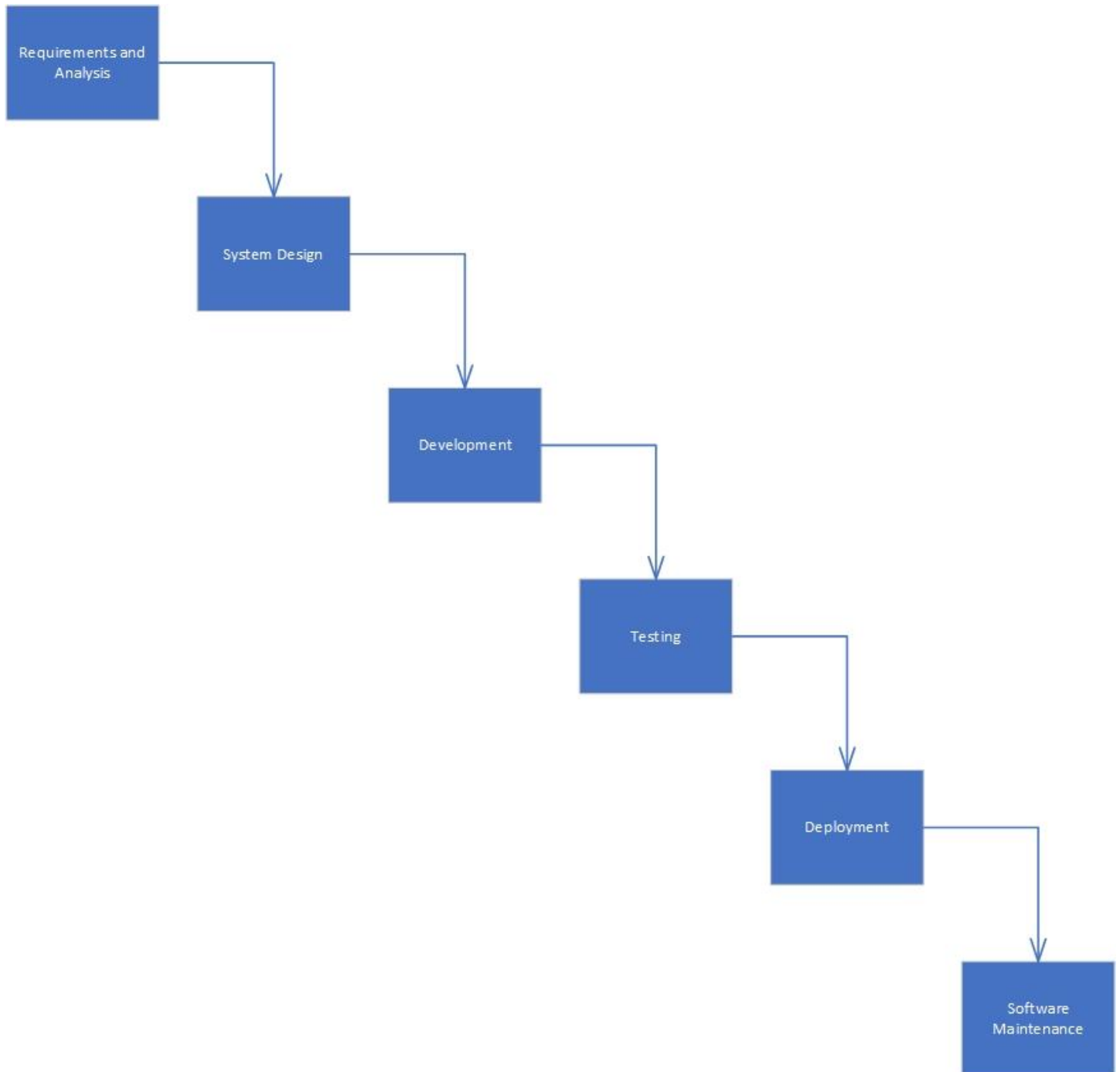


Figure 2: Waterfall Diagram

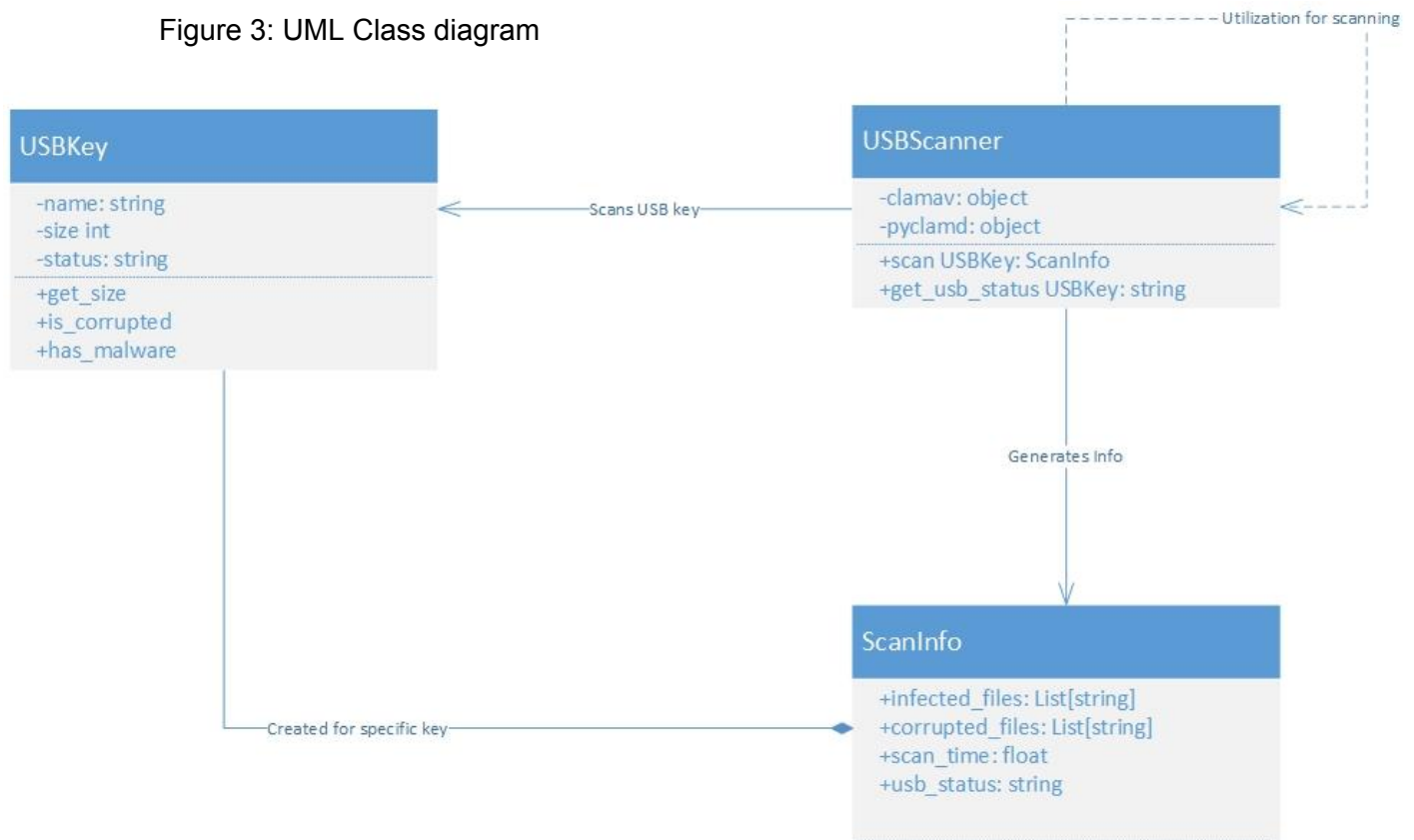




## 6 Steps to this design process:

1. Outline the requirements - which are, to be able to scan the USB keys for any potential risks and provide all the details of what was found.
2. Design of the architecture - the web app needs to be able to know whether a USB key has been connected, so that the ClamAV can scan the data for any corrupted files and provide all its findings
3. Developing the application - this involves all the code to be completed within the system so that the ClamAV can run its scans on the USB
4. Testing the application - will require multiple tests under certain conditions to determine if the required application features are running successfully
5. Deploying the application - if the application is running properly with all requirements met, the application can be deployed for users to try out
6. Maintenance (optional) - to fix any small bugs that come over time and to make sure it is always running smoothly along with extra features to be added if required.

Figure 3: UML Class diagram



This class diagram shows the design for a web app that scans USB keys and notifies users with all required information determining whether it has any potential risks or not along with other information given.

3 Classes:

1. USBKey - which represents the USB key that is used to check for properties such as name, size and status for its health whether it has viruses or not
2. USBScanner - which represents the web app that scans the USB keys using ClamAV and Pyclamd for scanning and determining its status
3. ScanInfo - is the information given to the user of what the USB scanner was able to retrieve such as a list of infected or corrupted files, along with how long it took and the current status of the USB's health.

# Results

ClamAv antivirus daemon is an open-source antivirus software to detect and remove viruses and malware. It operates as a background process and can be used with Python by calling its command-line interface using subprocess, allowing scripts to scan files for malware. The first result in the project involved the installation and activation of the ClamAV antivirus daemon.

Figure 4: ClamAV daemon

```
Executing: /lib/systemd/systemd-sysv-install enable clamav-daemon
carson@carson-VirtualBox:~$ sudo systemctl is-enabled clamav-daemon.service
enabled
```

systemctl is a command-line utility in Linux-based operating systems that is used to control and manage the systemd system and service manager.

Figure 5: ClamAv daemon status

```
carson@carson-VirtualBox:~$ sudo systemctl start clamav-daemon.service
carson@carson-VirtualBox:~$ sudo systemctl status clamav-daemon.service
● clamav-daemon.service - Clam AntiVirus userspace daemon
   Loaded: loaded (/lib/systemd/system/clamav-daemon.service; enabled; vendor
   Drop-In: /etc/systemd/system/clamav-daemon.service.d
            └─extend.conf
   Active: active (running) since Tue 2023-03-14 19:26:43 EDT; 4s ago
     Docs: man:clamd(8)
            man:clamd.conf(5)
            https://docs.clamav.net/
   Process: 30915 ExecStartPre=/bin/mkdir -p /run/clamav (code=exited, status=
   Process: 30916 ExecStartPre=/bin/chown clamav /run/clamav (code=exited, sta
   Main PID: 30917 (clamd)
      Tasks: 1 (limit: 3873)
     Memory: 99.5M
    CGroup: /system.slice/clamav-daemon.service
            └─30917 /usr/sbin/clamd --foreground=true

Mar 14 19:26:43 carson-VirtualBox systemd[1]: Starting Clam AntiVirus userspace
Mar 14 19:26:43 carson-VirtualBox systemd[1]: Started Clam AntiVirus userspace
lines 1-18/18 (END)
```

The daemon runs continuously in the background of the computer system and can be called and accessed from the code without any user interaction.

Figure 6: pyclamd initialization

```
1 import os
2 import pyclamd
3
4 usb_path = r'/media/carson/505F-5872'
5
6 clamav = pyclamd.ClamdAgnostic()
7
8 infected_files = []
9 cleaned_files = []
0 quarantine_dir = os.path.join(usb_path, "quarantine")
1
```

Next we begin the script by importing the necessary libraries; os and pyclamd, set a variable up holding the path directory to the desired USB. Next we initialize an instance of the class ClamdAgnostic from the pycamd library and assign it to the variable clamav. The ClamAgnostic class provides an interface between python and the ClamAV antivirus engine.

Figure 7: scan directory function

```
13 if not os.path.exists(quarantine_dir):
14     os.makedirs(quarantine_dir)
15
16 def scan_directory(directory):
17     for dirpath, _, filenames in os.walk(directory):
18         for filename in filenames:
19             file_path = os.path.join(dirpath, filename)
20             scan_result = clamav.scan_file(file_path)
21             if scan_result:
22                 infected_files.append(file_path)
23                 # move infected file to quarantine directory
24                 new_path = os.path.join(quarantine_dir, filename)
25                 os.rename(file_path, new_path)
26                 cleaned_files.append(new_path)
27
```

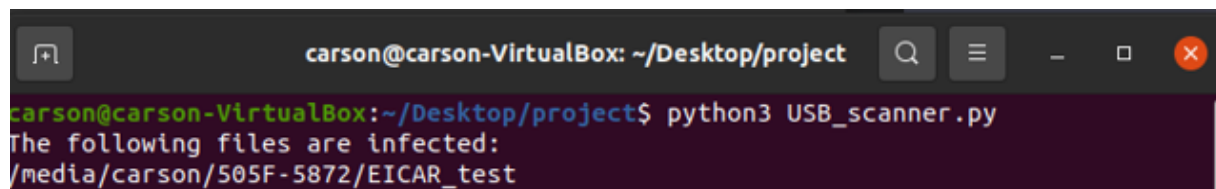
Next we create the function scan\_directory which loops through all files on the USB and passes them through the ClamAV scan\_file method. The method takes a file object as an argument and returns a dictionary with information about the scan results. It uses a combination of signature-based detection and other techniques to identify and isolate potentially dangerous files. If any of the files are found to be corrupted or malicious they will be moved into a quarantine directory created on the USB.

Figure 8: Subdirectory scanning

```
32 for root, dirs, _ in os.walk(usb_path):
33     for directory in dirs:
34         scan_directory(os.path.join(root, directory))
```

Scanning subdirectories is important as malware can hide in these locations. Many modern malware strains are polymorphic, causing them to change their code to evade detection by antivirus software. Scanning subdirectories and using advanced heuristics and behavioral analysis techniques can help ClamAV detect threats.

Figure 9: Running program and showing output

A screenshot of a terminal window titled 'carson@carson-VirtualBox: ~/Desktop/project'. The terminal shows the command 'python3 USB\_scanner.py' being executed. The output of the script is 'The following files are infected:' followed by the path '/media/carson/505F-5872/EICAR\_test' on the next line.

```
carson@carson-VirtualBox: ~/Desktop/project
carson@carson-VirtualBox:~/Desktop/project$ python3 USB_scanner.py
The following files are infected:
/media/carson/505F-5872/EICAR_test
```

To test the program's ability to check for corrupted and infected files we used an EICAR test. The EICAR test file is a commonly used standard file for verifying the proper functioning of antivirus software. It comprises a single line of text that is intentionally created to be detected as a virus signature by antivirus scanners.

The string of characters used is:

X5O!P%@AP[4PZX54(P^)7CC)7}\$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!\$H+H\*

# Conclusion and Recommendations

When it comes to our USB scanning application, its efficiency and easy-to-use UI allow for the ability to detect infected files and add for extra security. This virus-scanning program known as ClamAV is the heart of our application as it is a tool used by many to scan USB keys for any potential risks. With the help of this scanning program, the antivirus engine makes it possible for it to be reliable. Implementing the necessary guide to detecting potential malware and protection against data theft or system damage. By doing this, the application should be able to detect infected files and move them to a separate directory for quarantine, then proceed to scan all files within the USB in an efficient manner of time and provide a detailed report. With this, the users of this application will have a clear understanding of the contents of the USB and can figure out what to do afterward to make sure certain risks don't occur.

As a recommendation, the improvement for this code could be to add error handling for cases where the USB drive or quarantine folder is not accessible and to continuously update and improve the scanner application. Another feature could be to provide data, ensuring they are educated on the best way to deal with USB security along with “what to do and what not to do” when it comes to using their USB.

This project highlights how valuable cyber and computer security is for safeguarding against any threats or breaches to a person's data. The importance of implementing security concepts against cyber threats is crucial for any user or an organization as a whole.

# References

1. "ClamAV documentation," *Introduction - ClamAV Documentation*. [Online]. Available: <https://docs.clamav.net/>. [Accessed: 27-Mar-2023].
2. J. Ellingwood, "How to use Systemctl to manage systemd services and Units," *DigitalOcean*, 24-Jan-2022. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units>. [Accessed: 27-Mar-2023].
3. "OS - miscellaneous operating system interfaces," *Python documentation*. [Online]. Available: <https://docs.python.org/3/library/os.html>. [Accessed: 27-Mar-2023].
4. "PyClamd," *PyPI*. [Online]. Available: <https://pypi.org/project/pyClamd/>. [Accessed: 27-Mar-2023].
5. "Ubuntu documentation," *ClamAV - Community Help Wiki*. [Online]. Available: <https://help.ubuntu.com/community/ClamAV>. [Accessed: 27-Mar-2023].

# Contribution Matrix

| Tasks                                     | Carson<br>McClelland | Shwan Majeed | Walid Ayub | Kalid Ajibade |
|---|----------------------|--------------|------------|---------------|
| <b>Abstract</b>                           |                      |              | X          |               |
| <b>Introduction and<br/>Background</b>    |                      |              | X          |               |
| <b>Project Objective</b>                  |                      | X            |            |               |
| <b>Project<br/>Methodology</b>            |                      |              |            | X             |
| <b>Results</b>                            | X                    |              |            |               |
| <b>Conclusion and<br/>Recommendations</b> |                      |              | X          |               |
| <b>Editing and<br/>Formatting</b>         | X                    | X            | X          | X             |
| <b>Presentation</b>                       | X                    | X            | X          | X             |
| <b>Coding</b>                             | X                    |              | X          |               |
| <b>Testing and<br/>Debugging</b>          | X                    | X            |            | X             |