USB Scanner Application

Group #17: Carson McClelland, Shwan Majeed, Walid Ayub, Kalid Ajibade

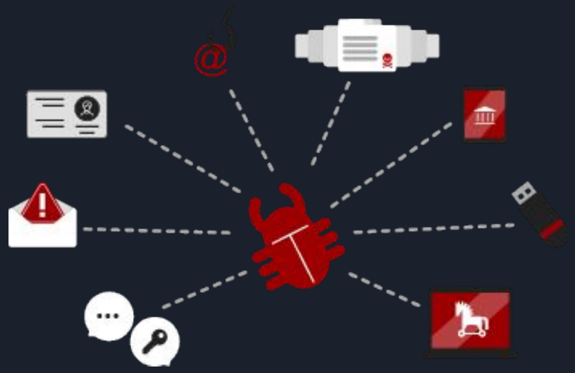
Outline

Project Aim
Project Features
Project Implementation Details
Physical Overview Of The Project
Testing and Results
Project Limitations
Conclusion

Project Aim

Project Aim

- To create a USB scanner application that can identify malware, protect against data theft, and prevent against system damage
- Scan and detect all files within the USB to quickly eliminate the possibility of any threats of transferring to the users device
- Detect all infected files and move them to a separate directory to isolate them from the rest of the files
- The user should be able to safely use the USB



Project Features

Project Features

- Detects that a USB has been plugged into the machine
- Using ClamAV, the application effectively detects and sorts corrupted files
- Feedback report of results
- Quick scanning process

Project Implementation

Project Implementation

To implement our project we utilized various tools such as:

- Linux :

  Is open source software, which offer more flexibility and has high security

- Python:

  Works very well in cyber security because of it analysis capabilities, and has a vast library

- Clamav-daemon:

  Is an antivirus engine capable of discovering, viruses, malware, trojans and many other threats to our devices

- Pyclamd:

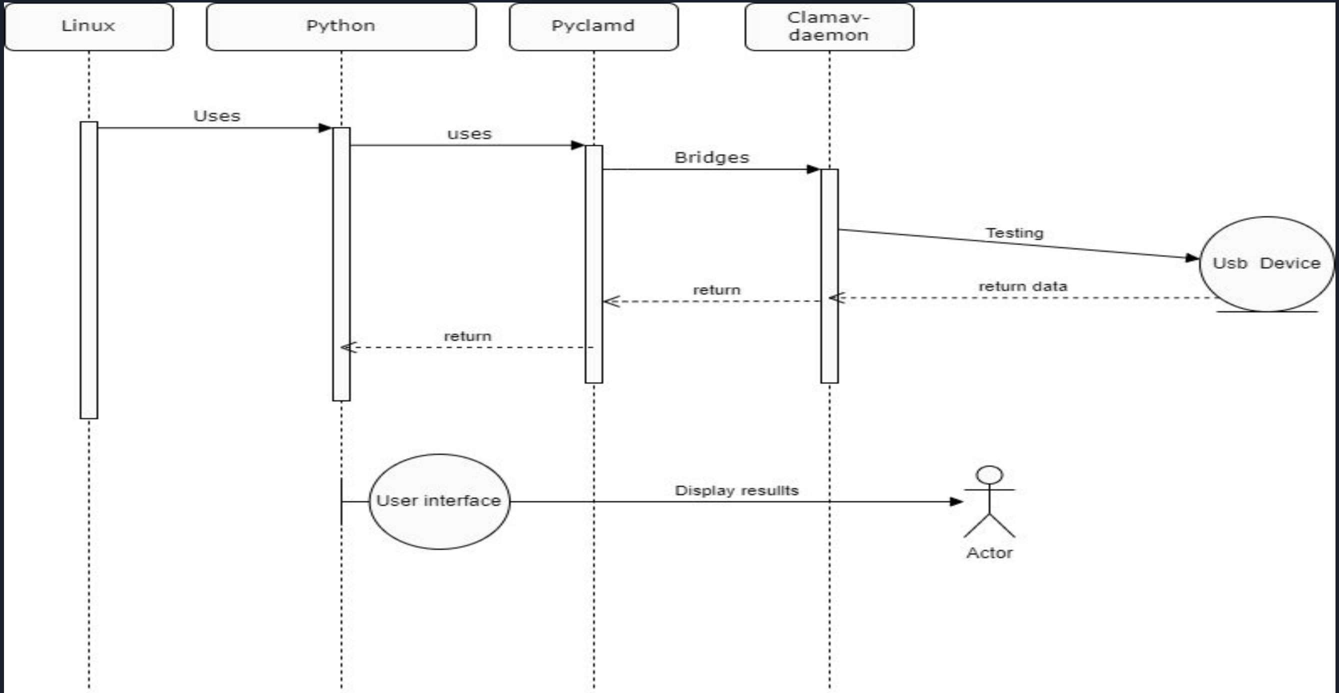
  Acts as a bridge between python and clamav-daemon, increase the overall efficiency.

- EICAR test:

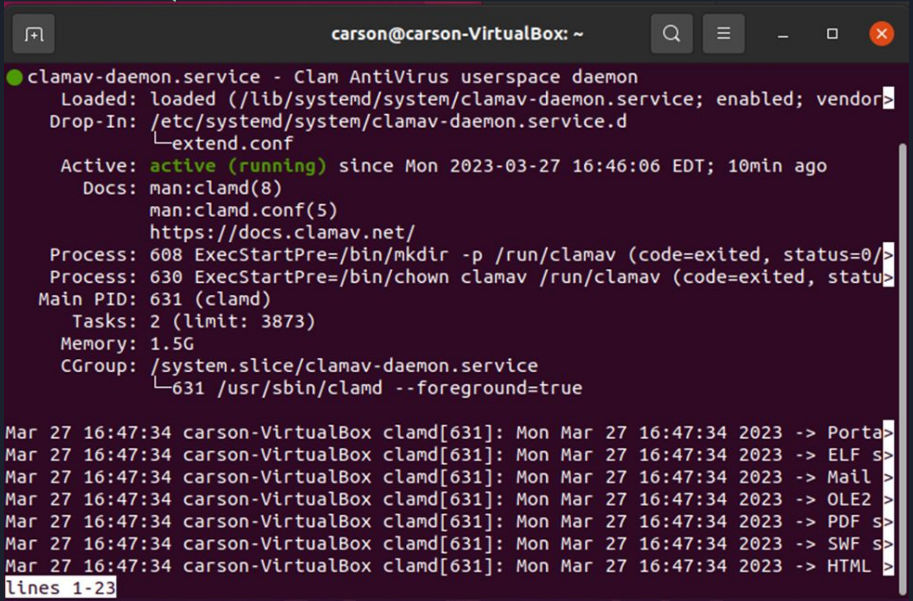
  A standard test file used to verify if antivirus software is working properly

Physical Overview Of The Project

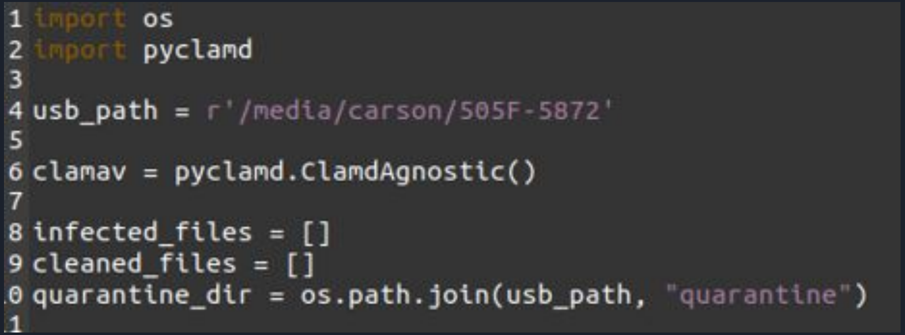
Physical Overview Of The Project



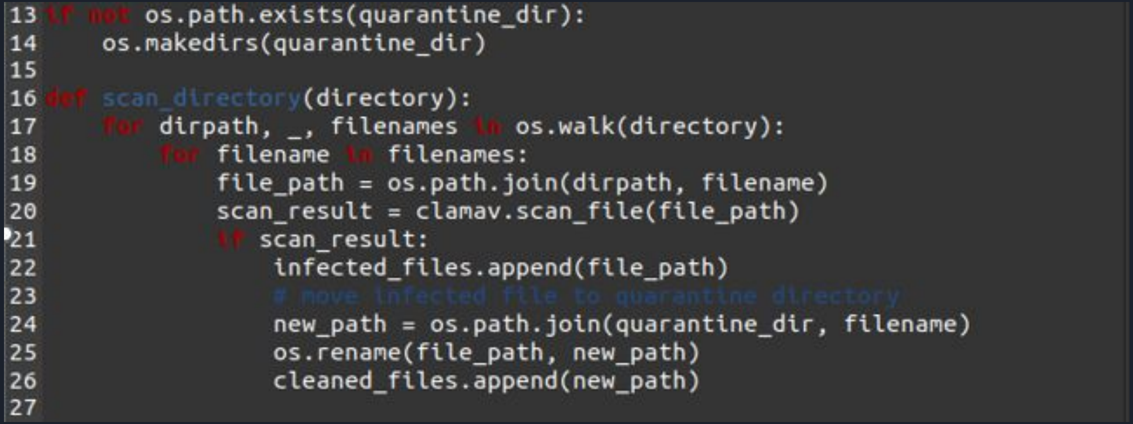
Testing and Results



```
● 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 Mon 2023-03-27 16:46:06 EDT; 10min ago
       Docs: man:clamd(8)
             man:clamd.conf(5)
             https://docs.clamav.net/
    Process: 608 ExecStartPre=/bin/mkdir -p /run/clamav (code=exited, status=0/>
    Process: 630 ExecStartPre=/bin/chown clamav /run/clamav (code=exited, statu>
   Main PID: 631 (clamd)
      Tasks: 2 (limit: 3873)
     Memory: 1.5G
     CGroup: /system.slice/clamav-daemon.service
             └─631 /usr/sbin/clamd --foreground=true

Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> Porta>
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> ELF s>
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> Mail >
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> OLE2 >
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> PDF s>
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> SWF s>
Mar 27 16:47:34 carson-VirtualBox clamd[631]: Mon Mar 27 16:47:34 2023 -> HTML >
lines 1-23
```



```python
import os
import pyclamd

usb_path = r'/media/carson/505F-5872'

clamav = pyclamd.ClamdAgnostic()

infected_files = []
cleaned_files = []
quarantine_dir = os.path.join(usb_path, "quarantine")

```



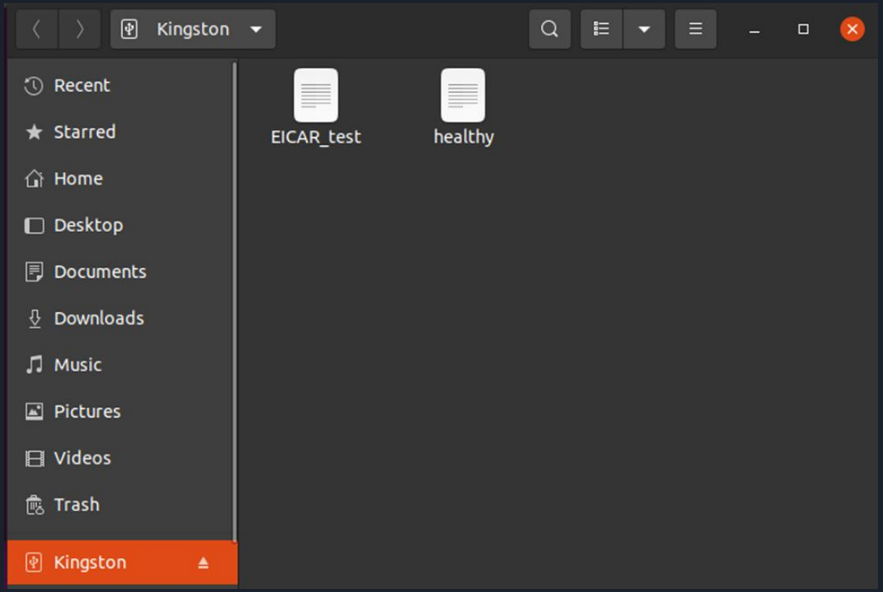
```python
if not os.path.exists(quarantine_dir):
    os.makedirs(quarantine_dir)

def scan_directory(directory):
    for dirpath, _, filenames in os.walk(directory):
        for filename in filenames:
            file_path = os.path.join(dirpath, filename)
            scan_result = clamav.scan_file(file_path)
            if scan_result:
                infected_files.append(file_path)
                # move infected file to quarantine directory
                new_path = os.path.join(quarantine_dir, filename)
                os.rename(file_path, new_path)
                cleaned_files.append(new_path)

```

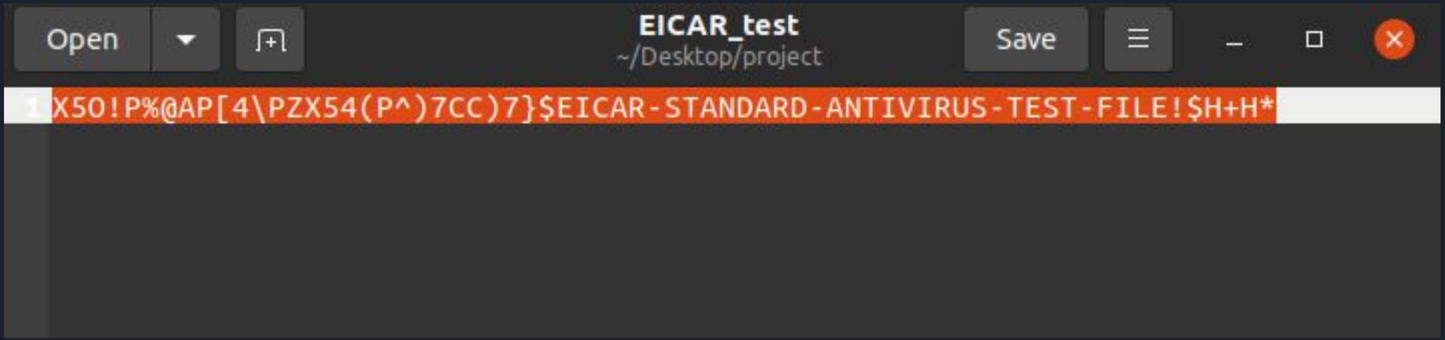
1. Signature-based Detection

   - comparing a file's digital signature to a database of known malware signatures.

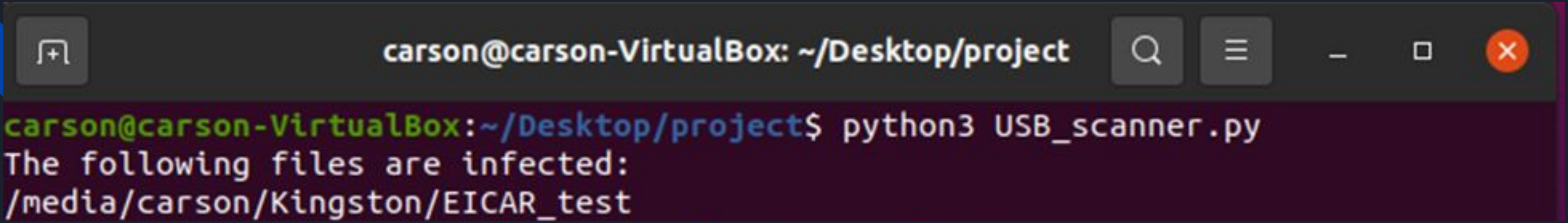
2. Heuristic Analysis

   - detect malware based on the behavior of a file, rather than relying on a known signature.

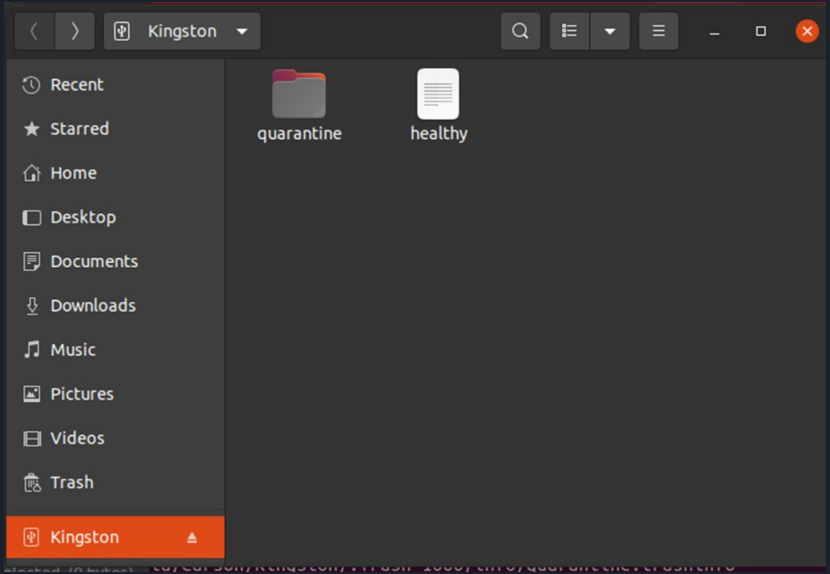


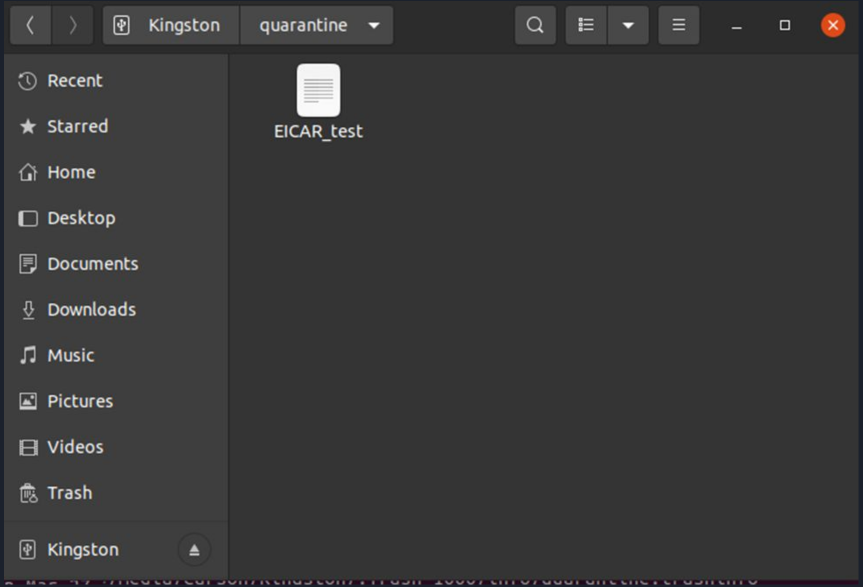


```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```



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





Project Limitations

Project Limitations

- **Malicious bypass attacks on the application**

  - Update the application regularly when in use

- **File Format limitations**

  - Increase scalability in file formats

- USB security risks

  - Encryption and other security measure can be used to limit risks

Conclusion

Conclusion

- Implement the necessary guide to detecting potential malware and protection against data theft or system damage.
- The code can be improved by adding error handling and continuously updating the scanner application
- Another feature could be added to provide education on USB security best practices
- The project highlights the importance of cybersecurity and implementing security concepts to protect against threats and breaches to personal data