KEYLOGGER – PROJECT DOCUMENTATION

Jagadeesh Kommineni

Keylogger – Project Documentation

Project Title

GUI-Based Educational Keylogger Using Python and Tkinter

Table of Contents

---

1. Overview

This project is a **keyboard activity logger** built with Python and Tkinter, intended **strictly for educational and authorized use only**. It captures keystrokes and stores them in a text file (keylog.txt). The application features a GUI interface that allows users to start or stop logging and view recorded logs.

This project demonstrates knowledge of:

- Keyboard event handling with Python
- Real-time logging mechanisms
- Multithreading in GUI applications
- Ethical tool development in cybersecurity education

---

2. Objective

The goal of this project is to:

- Develop a functional and safe educational keylogger
- Implement keyboard event listening using pynput
- Provide a user-friendly interface with status updates
- Emphasize ethical and legal boundaries of monitoring tools
- Enable log viewing through a secure in-app viewer

---

3. Ethical Disclaimer

This software is developed solely for:

- **Educational purposes**
- **Ethical demonstrations**
- **Authorized environments only**

Unauthorized use of this application on systems without explicit consent may violate privacy laws and is strictly discouraged.

The application includes an inbuilt disclaimer that is shown on startup.

---

4. Technologies and Libraries Used

Technology	Description
Python	Core programming language
Tkinter	GUI creation
pynput	Keyboard event handling
threading	Background listener execution
os	File existence checks

---

5. Key Features

- Captures all keystrokes in real time
- Start and stop logging with the click of a button
- View captured logs in a scrollable, read-only text window
- GUI-based design with status indicators

- Ethical use prompt on application launch

---

6. System Architecture

text
CopyEdit
[Start Application]
↓
[Ethical Disclaimer Display]
↓
[User Controls Logging]
↓
[Keystrokes Captured → Stored in keylog.txt]
↓
[Logs Viewed via GUI Window]

- All key presses are captured using the pynput.keyboard.Listener.
- Logs are stored in plain text.
- GUI operates asynchronously to prevent freezing using threading.

---

7. Code Walkthrough

a. on_press(key)

Captures each keypress and appends it to keylog.txt. Special keys are recorded with bracket notation.

```
python
CopyEdit
try:
   f.write(f"{key.char}")
except AttributeError:
   f.write(f"[{key}]")
```

b. start_keylogger()

Initializes the pynput listener and sets the is_listening flag. Prevents multiple instances from running concurrently.

c. stop_keylogger()

Safely stops the listener and resets flags. Updates the GUI with a stopped status.

d. view_logs()

Opens a new Toplevel window and displays the contents of keylog.txt in a scrollable, read-only text area.

e. show_disclaimer()

Displays a startup warning informing users of ethical use constraints.

---

8. User Interface Design

The application is designed with:

- A black-themed interface for minimal eye strain
- Clear, bold titles for purpose transparency
- Status label for live feedback
- Three main action buttons:
    - Start Logging
    - Stop Logging
    - View Logs

All actions are executed safely in the background, keeping the GUI responsive.

---

9. Security Considerations

- This project does **not** transmit data to any server or network.
- Logging occurs locally within the same directory.
- The user is explicitly warned about **ethical use** via a popup window.
- No stealth mode or startup persistence is implemented, avoiding unethical implications.

---

10. Future Enhancements

Feature	Description
Timestamp logging	Include date/time per keypress
Log encryption	Secure the output file with basic encryption
Export to CSV	Convert log to structured CSV format

Feature	Description
Stealth mode toggle	(Only for educational research with permissions)
Keystroke visualization	Visual replay of captured input in simulation

---

11. Conclusion

This keylogger application is an educational tool to demonstrate:

- How keystrokes can be captured using Python
- The responsibilities and ethical concerns of such software
- Basic GUI development for interactive security tools

It serves as a foundational project for students, researchers, or cybersecurity interns interested in learning about surveillance technologies, keylogging mechanisms, and how to responsibly develop tools in a controlled environment.