# Encrypted Keylogger with GUI and CLI

## Introduction

In the evolving landscape of cybersecurity, understanding the behavior and mechanics of threat tools is essential for building robust defense mechanisms. This project introduces a proof-of-concept encrypted keylogger with both Graphical User Interface (GUI) and Command-Line Interface (CLI) capabilities. Designed strictly for ethical and educational purposes, the keylogger captures user keystrokes, encrypts the data in real-time, and provides flexible methods for managing and viewing logs. Through this project, students and security professionals can gain hands-on experience with secure logging techniques and the structure of threat simulations.

## Abstract

The aim of this project is to develop a Python-based keylogger that securely captures and encrypts keystrokes using the Fernet symmetric encryption method from the Cryptography library. The tool features a responsive GUI, allowing users to start or stop the logger and view decrypted logs in real time. Additionally, a CLI mode is provided for advanced users or headless operation. All logs are timestamped and saved in an encrypted format to ensure confidentiality. This project serves as a practical educational tool that demonstrates the core mechanics of keylogging in a controlled, ethical, and responsible manner.

## Tools Used

- **Python 3**: Core programming language used for building the application.
- **pynput**: Library used to capture and monitor keyboard input.
- **cryptography (Fernet)**: Provides secure symmetric encryption for storing logs.
- **tkinter**: Enables the creation of a simple and effective graphical interface.
- **threading**: Facilitates non-blocking execution for GUI responsiveness.
- **datetime and os**: Used for generating timestamps and handling file operations.

# Steps Involved in Building the Project

1. Project Structure
   Designed a clean structure with folders for logs/ and key.key.

2. Keystroke Logging
   Implemented pynput.keyboard.Listener to record all keystrokes. Defined a kill switch using the ESC key.

3. Symmetric Encryption
   Generated and securely stored an encryption key using Fernet. Encrypted each keystroke immediately before writing to file.

4. Encrypted Log Storage
   Wrote logs to logs/encrypted_keys.log using binary mode. Included timestamps for auditability.

5. Graphical Interface (GUI)
   Built using tkinter. Included buttons to start, stop, and view decrypted logs. Used ScrolledText for clean output display.

6. Command-Line Interface (CLI)
   Included flags such as --start, --stop, and --view. CLI runs automatically if arguments are passed; GUI launches by default otherwise.

# Conclusion

This project successfully demonstrates a practical approach to understanding the working of keyloggers in a safe, ethical, and controlled environment. The integration of both GUI and CLI interfaces provides users with flexibility and a deeper understanding of real-world cybersecurity threats. By incorporating encryption, the project also emphasizes the importance of data confidentiality and security. It serves not only as a technical exercise but also as an educational tool to enhance awareness of the methods attackers use—and how to defend against them.