Smart Behavior-Based Keylogger

Author: YUVARAJ S

1. Abstract

This project implements a Python-based keylogging system that goes beyond simple keystroke collection. It captures input, measures user typing behavior, and classifies keystrokes using rule-based heuristics. The system stores logs in both plain text and encrypted form using Fernet symmetric encryption, ensuring confidentiality even if the log files are accessed. The project provides a practical foundation for understanding offensive security techniques and secure data handling, while enabling future integration of machine-learning-based behavior analysis.

2. Introduction

Keystroke monitoring is a core technique used in both offensive security and digital forensics. Understanding how keyloggers function is essential for building effective detection and defense strategies.

This project demonstrates how a behavior-aware keylogger works internally: capturing keystrokes, tracking timing patterns, classifying user intent (URL, command, password-like input, or general text), and storing the captured data securely.

Instead of focusing only on raw logging, the project emphasizes data security and behavioral analysis — two areas critical for modern threat detection.

3. Tools & Technologies Used

Python 3 – Core programming language

pynput – Captures keyboard events at OS level

cryptography (Fernet) – Encrypts log files using strong symmetric encryption

datetime – Adds precise timestamps to each entry

re (Regex) – Classifies keystrokes into behavior categories

virtualenv – Isolated project environment

Git & GitHub – Version control and project hosting

4. Steps Involved in Building the Project

Environment Setup

- Created a virtual environment for dependency isolation

- Installed necessary packages (pynput, cryptography)

- Initialized Git repository for version control

Project Structure

keylogger_uv/

core.py → Key capture & logging logic

clic.py → CLI handling

__init__.py → Package initializer

setup.py → Packaging script

Core Keylogging Features

- Records every key press and release

- Measures time delay between consecutive keystrokes

- Uses heuristics to classify input:

• URL → contains http, .com, .in, etc.

• Command → starts with /, -, sudo, etc.

• Password-like → long continuous alphanumeric patterns

• General text → everything else

Encryption & Logging

- Generates a Fernet encryption key

- Saves logs in:

• Plain text → Easy for analysis

• Encrypted binary → Secure storage

- Ensures encrypted logs can only be accessed with the correct key

Command-Line Interface (CLI)

Provides simple commands to:

- Start logging

- View/analyze plain logs

- Decrypt encrypted logs

Packaging & Execution

- Configured setup.py for setuptools packaging

- Added entry_points to create a globally accessible command (keyvi)

- Enabled smooth tool execution from terminal

5. Conclusion

This project delivers more than a basic keylogger. It introduces behavior-aware logging, secure encrypted storage, and modular design suitable for both research and education.

The system serves as a practical base for cybersecurity learners while providing enough structure to integrate advanced features such as anomaly detection, user profiling, or AI-driven behavior analysis in the future.