Keylogger with Encrypted Data Exfiltration

Introduction

A **keylogger** is a program that records keyboard inputs. In cybersecurity, building a keylogger as a **proof-of-concept** helps understand data security, encryption, and ethical hacking practices. This project focuses on creating a keylogger that captures keystrokes, encrypts them, and simulates exfiltration to a local server, ensuring all testing is performed ethically on the user's own machine.

Abstract

This project demonstrates how sensitive data can be captured, encrypted, and safely managed. Using Python libraries like **pynput** and **cryptography**, the keylogger captures all keystrokes with timestamps, encrypts the logs using **Fernet symmetric encryption**, and stores them securely. A **Flask web interface** is provided to start, stop, and decrypt logs for monitoring. The project also includes a **kill switch** (ESC key) and **startup persistence**, making it a comprehensive educational tool for cybersecurity students.

Tools Used

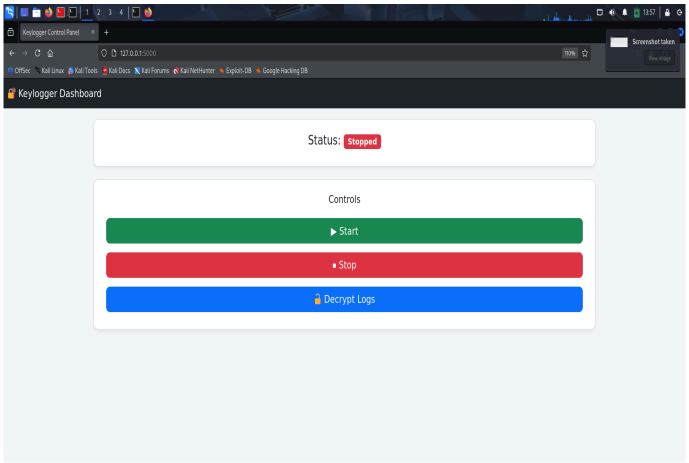
- **Python 3** – Programming language
- **pynput** – Captures keyboard inputs in real-time
- **cryptography (Fernet)** – Encrypts keystroke logs securely
- **Flask** – Creates a web interface/dashboard for controlling the keylogger
- **VirtualBox Shared Folders / Localhost** – For simulating log exfiltration
- **datetime** – Adds timestamps to keystroke logs

Steps Involved in Building the Project

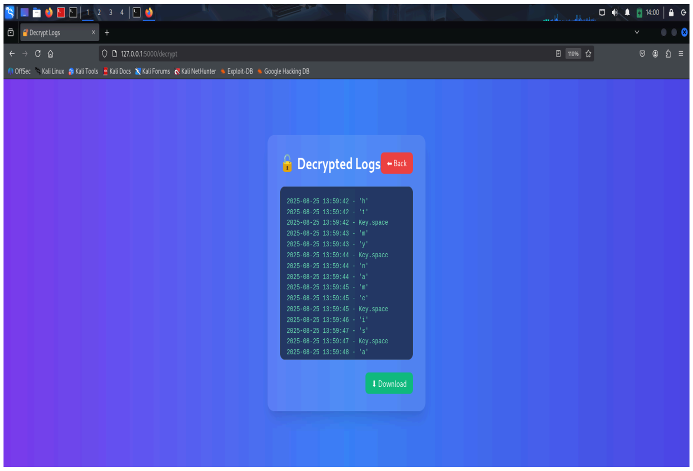
1. **Set up environment**
    - Install Python 3 and required libraries (pynput, cryptography, flask).

   - Optionally, set up a virtual environment.
2. **Capture keystrokes**
   - Use pynput.keyboard.Listener to monitor keypress events.
   - Append each keystroke to a local log file with a timestamp.
3. **Encrypt logs**
   - Generate or load a symmetric key (Fernet).
   - Encrypt the plaintext log file and save as keystrokes.log.enc.
   - Delete plaintext logs to maintain security.
4. **Simulate exfiltration**
   - Move or copy encrypted logs to a shared folder or localhost directory.
5. **Add startup persistence and kill switch**
   - Add a cron job to run the keylogger automatically on Linux startup.
   - Implement an **ESC key kill switch** to stop the keylogger.
6. **Build web interface**
   - Use **Flask** to create a dashboard to start/stop the keylogger.
   - Display decrypted logs safely in the browser using decrypt.py logic.

Screenshot



Status: Stopped



```
2025-08-25 13:59:42 - 'h'
2025-08-25 13:59:42 - 'i'
2025-08-25 13:59:42 - Key.space
2025-08-25 13:59:43 - 'm'
2025-08-25 13:59:43 - 'y'
2025-08-25 13:59:44 - Key.space
2025-08-25 13:59:44 - 'n'
2025-08-25 13:59:44 - 'a'
2025-08-25 13:59:45 - 'm'
2025-08-25 13:59:45 - 'e'
2025-08-25 13:59:45 - Key.space
2025-08-25 13:59:46 - 'i'
2025-08-25 13:59:47 - 's'
2025-08-25 13:59:47 - Key.space
2025-08-25 13:59:48 - 'a'
```

Conclusion

This project provides practical exposure to **ethical keylogging, encryption, and web interface development** in cybersecurity. By building this keylogger, students learn how to handle sensitive data responsibly, implement encryption for data security, and simulate exfiltration without violating ethical guidelines. The project demonstrates a complete workflow from **data capture → encryption → secure access via a dashboard**, making it an excellent educational tool for cybersecurity internships.