Keylogger with Encrypted Data Exfiltration

Project Report

---

1. Introduction

The increasing reliance on digital communication and data processing has heightened concerns around information security. A keylogger is a program that records the keys struck on a keyboard.

In cybersecurity, understanding how keyloggers function is essential for both offense (ethical hacking, red teaming) and defense (detection, prevention). This project presents a **proof-of-concept keylogger** that encrypts keystrokes and simulates secure data exfiltration to a local server.

---

2. Abstract

The purpose of this project is to develop a functional keylogger that:

- Captures keystrokes using Python,
- Encrypts the log using `cryptography.fernet` for secure storage,
- Simulates the exfiltration of logs to a remote (localhost) server via TCP sockets,
- Implements a kill switch and Windows startup persistence.

This project is intended for educational and ethical use, demonstrating how such tools operate and can be analyzed to enhance cybersecurity awareness and prevention strategies.

---

3. Tools & Technologies Used

Tool / Library	Purpose
Python	Programming language
`pynput`	Keystroke capturing
`cryptography`	AES-based encryption (Fernet)
`socket`	Exfiltration using TCP
`threading`	Multithreaded socket server
`winreg`	Add to Windows startup
`datetime`, `os`	Timestamp logs, manage paths/files

---

4. Steps Involved in Building the Project

1. **Environment Setup**
   Installed necessary libraries using pip:

   ```
   nginx
   CopyEdit
   pip install pynput cryptography
   ```

2. **Key Generation & Encryption Setup**
   Generated and stored an AES-based Fernet key. All logs were encrypted before storage.
3. **Keystroke Logging**
   Captured keyboard events using `pynput.Listener`. Every 10 characters were timestamped, encrypted, and appended to a local log file.
4. **Simulated Data Exfiltration**
   A local TCP server (`localhost:4444`) was implemented using Python sockets. It read the encrypted log and sent it line-by-line to any connected client.
5. **Kill Switch**
   The logger stops execution if the user types a special phrase: `q!exit`.
6. **Startup Persistence (Windows)**
   Used the Windows registry to add the script to startup automatically under `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`.

---

5. Conclusion

This keylogger project provided hands-on experience in Python programming, encryption, socket communication, and system persistence mechanisms. By ethically building and analyzing such tools, cybersecurity learners gain better insight into how attackers may exploit systems—empowering them to build better defenses. This knowledge is vital for cybersecurity professionals engaged in penetration testing, malware analysis, and secure system design.