Post-Breach Black Box Logger - Encrypted, Tamper-Evident Keylogger

Introduction:

This project implements a secure, tamper-evident keylogger that records attacker activity after a breach.

It encrypts keystroke data using AES-256-GCM and verifies integrity with HMAC-SHA256.

Abstract:

Designed for forensic analysts, this logger captures keystrokes, window titles, and process names.

It ensures confidentiality and tamper resistance by encrypting each entry with AES-GCM and signing it

using HMAC. Data is stored as base64-encoded JSON for easy parsing and audit.

Tools Used:

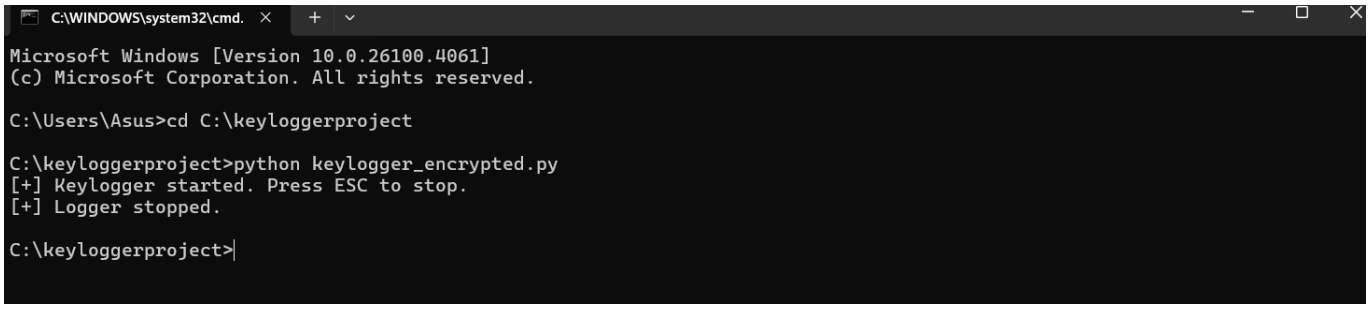
- Python 3.x
- pynput (keyboard listener)
- cryptography (AES-GCM, HMAC)
- psutil, pywin32 (window/process context)
- os, json, base64 (core utilities)

Key Features:

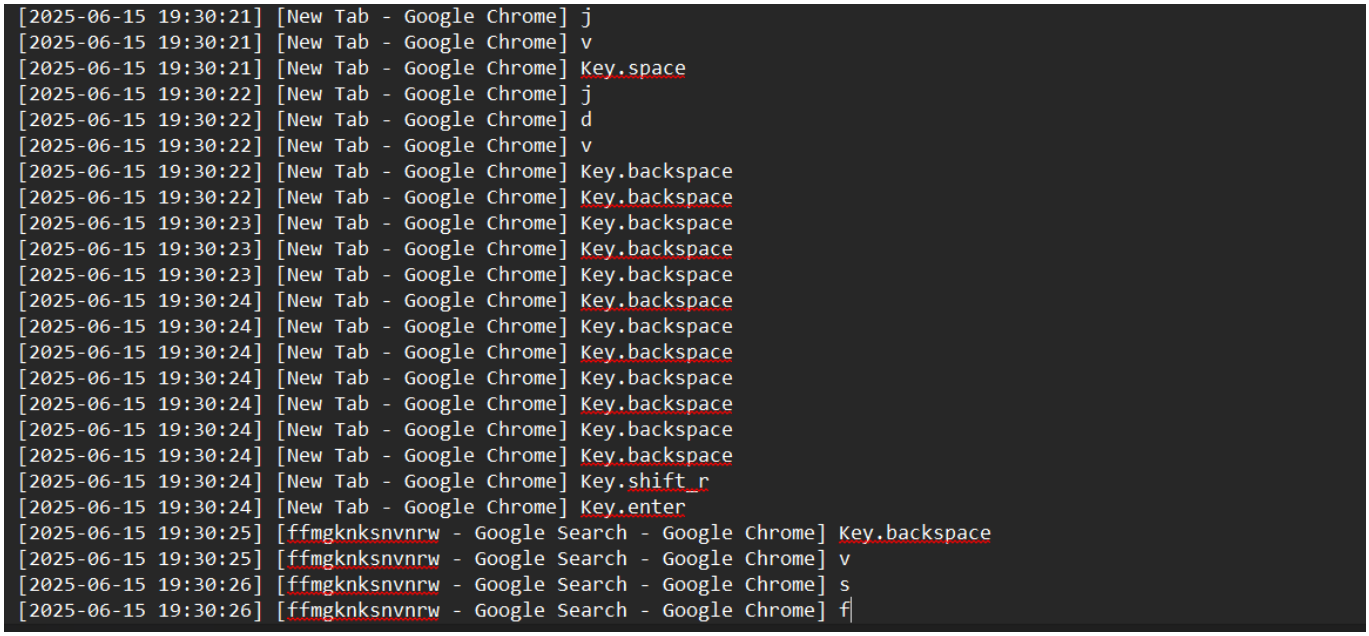
- AES-256-GCM encryption with random nonce
- HMAC-SHA256 tamper detection
- Records active window + process context
- Lightweight, runs without admin rights
- Logs stored locally in encrypted, base64-encoded format
- Stops with ESC key; logs are easily decrypted for analysis

Screenshots & Workflow

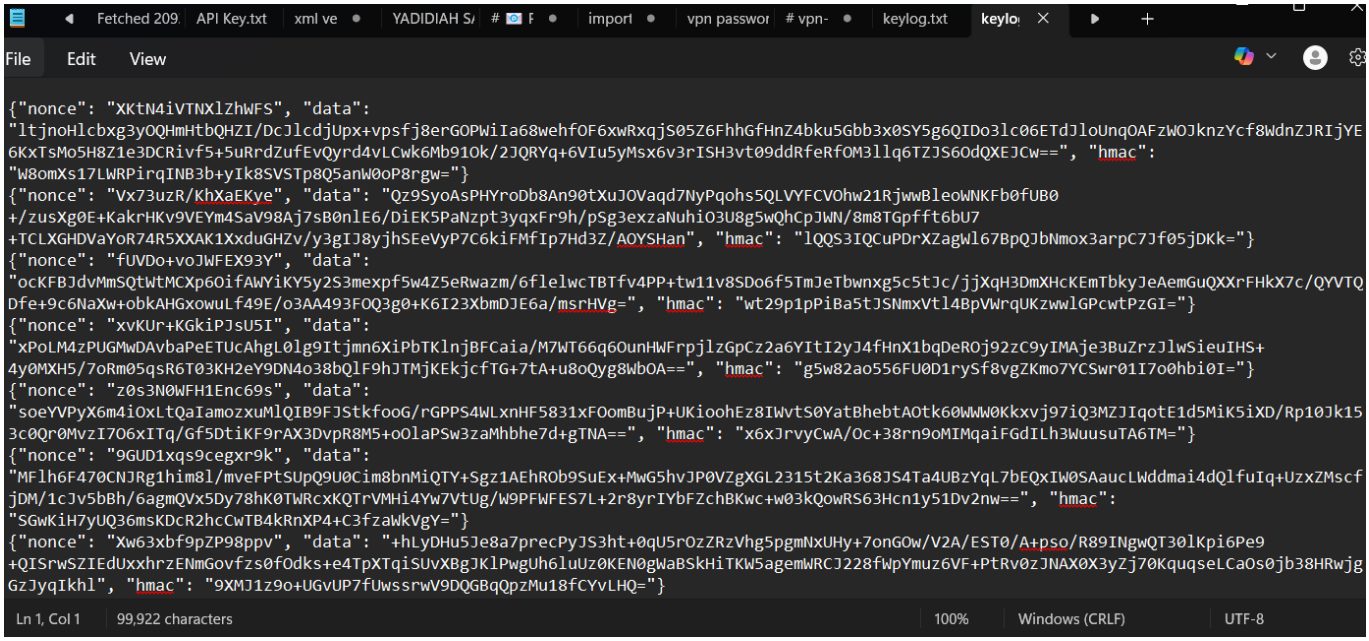
1. Logger running in CMD - starts and stops with ESC



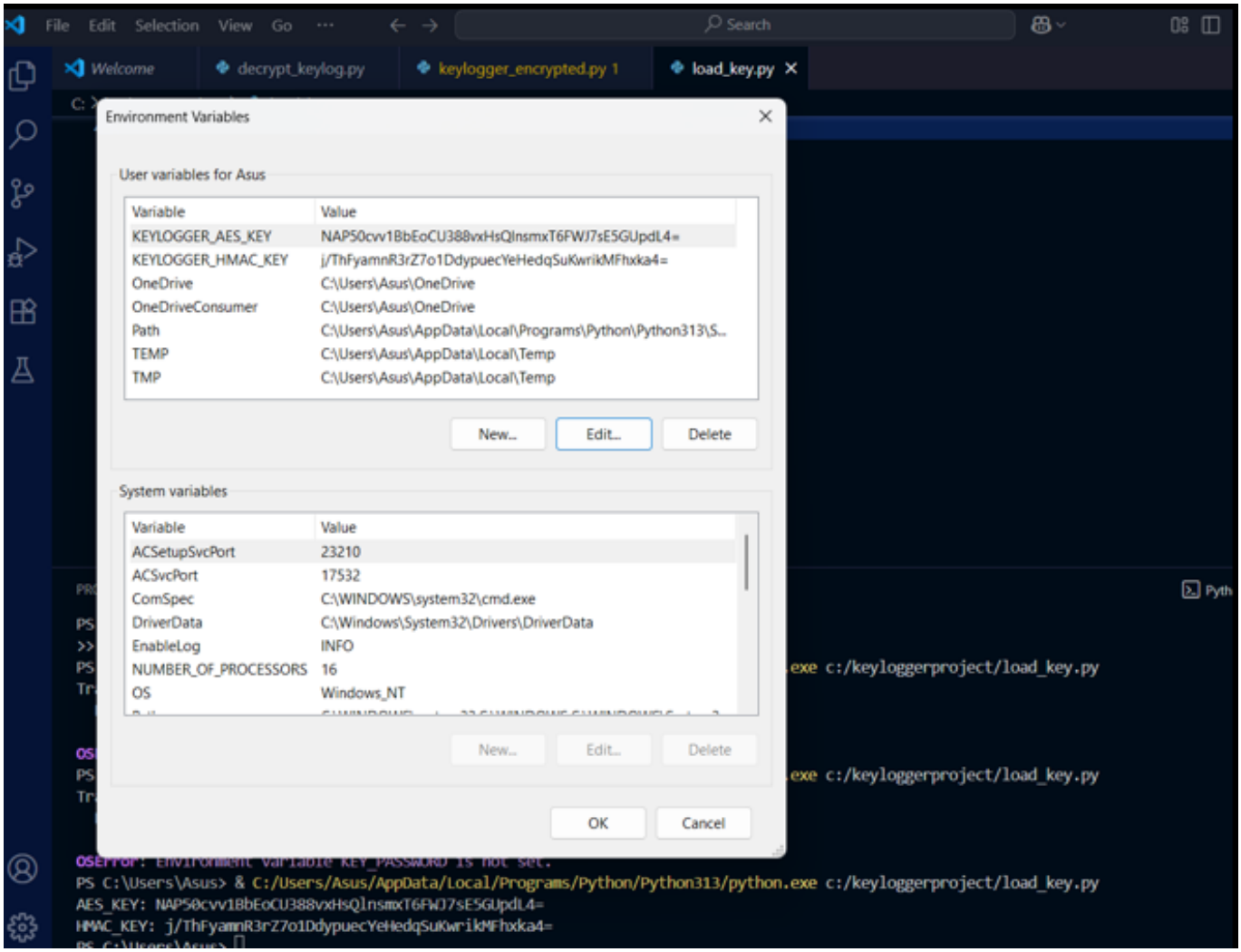
2. Decrypted log showing keystrokes and context



3. Encrypted JSON logs with AES-GCM and HMAC



4. Secure key storage in environment variables



Conclusion:

This project delivers a highly secure, ethical, and forensic-friendly keylogger solution.

Its tamper-evident and encrypted logging capabilities help reconstruct breach activity while preserving

evidence integrity. It is ideal for analysts working in environments without full-scale EDR solutions.