◆ POST-BREACH BLACK BOX LOGGER — ENCRYPTED,TAMPER-EVIDENT KEYLOGGER

◆ Introduction

Today's organizations face sophisticated cyberthreats and insider attacks that often destroy or manipulate logs to conceal traces. The **Post-Breach Black Box Logger** is designed to aid cybersecurity analysts by collecting key forensic data from a compromised endpoint in a lightweight, tamper-evident, and encrypted format — without needing elevated privilege.
This tool provides a trustworthy view into attacker activity after a breach, helping incident responders piece together a timeline of events quickly and accurately.

◆ Abstract

The **Post-Breach Black Box Logger** is a lightweight, post-incident forensic tool.
It performs:

- **AES-256-GCM encryption:** To keep captured keystrokes and context confidential.

- **HMAC-SHA256:** To guarantee tamper-evident storage.

- **Context Capture:** To aid investigations by adding directory and process details alongside keystrokes.

The encrypted messages are base64-encoded and serialized in JSON format for convenient storage and eventual decryption by analysts.

🔹 Tools Used

✅ **Python 3.x**
✅ **pynput:** for keystroke capture
✅ **pywin32, psutil:** for process and directory context
✅ **Cryptography:** for AES-GCM and HMAC
✅ **os, base64, json:** for file operations and packaging

🔹 Steps Involved in Building the Project

➥ **Initiate:**
Logger starts upon launch and attaches to keyboard events.

➥ **Capture:**
For each keystroke, it:

- Records the key pressed.
- Captures the active window title.
- Logs associated directory and process path.

➥ **Encrypt:**
Encrypts this data with AES-256-GCM using a unique 96-bit nonce.

➥ ** HMAC:**
Generulates HMAC-SHA256 to enable tamper-detection.

➥ **Serialize:**
Packages encrypted data, HMAC, and context into base64-encoded JSON.

➥ **Store:**
Writes messages safely to a local file for later decryption and analysis.

🔹 Conclusion

The **Post-Breach Black Box Logger** successfully provides:
✅ Tamper-evident, encrypted logging of keystrokes.
✅ Valuable context alongside captured data.
✅ A lightweight, non-intrusive solution for incident response — without needing elevated privilege.

This tool highlights proficient use of:
✅ AES-GCM, HMAC, and base64 techniques
✅ Python scripting and API integration
✅ Security best practices for lightweight post-breach investigations

The Post-Breach Black Box Logger can aid incident handlers in accurately reconstructing attack timelines and identifying suspicious activity — all while avoiding kernel components or elevated privilege.

✅ **Final Note:**
This tool underscores the ability to combine coding skills with cybersecurity principles — designing a lightweight, tamper-evident logging solution for incident response.