PROJECT REPORT: KEYLOGGER SOFTWARE (ETHICAL USE)

OVERVIEW

Title: Keylogger Software (For Ethical Use Only)
Category: Surveillance & Logging
Platform: Kali Linux
Language: Python 3

PURPOSE:

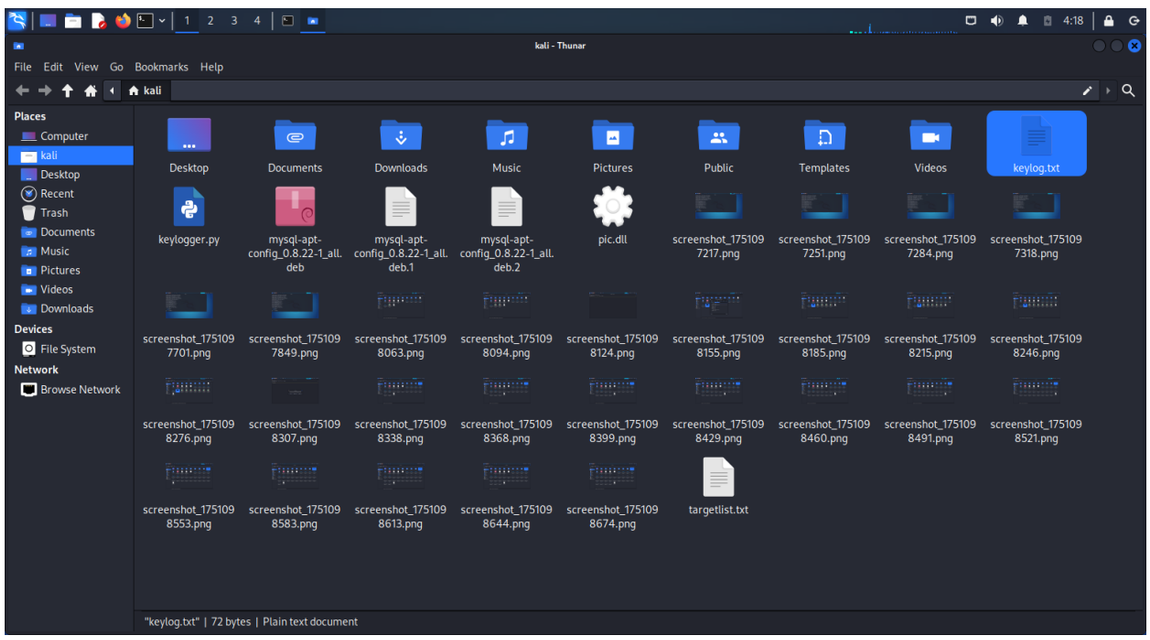
To ethically monitor system activity by logging keyboard strokes and capturing periodic screenshots for security analysis and awareness.

FEATURES IMPLEMENTED

FEATURE	STATUS	DESCRIPTION
Keyboard Activity Logging	Working	All keystrokes are saved in keylog.txt
Screenshot Logging	Working	Screenshots taken every 30 seconds using scrot

WORKING

- The script runs in the background.
- Every key you press is recorded.
- Every 30 seconds, the tool takes a screenshot of the current desktop.
- Logs are saved in:
    - keylog.txt (keyboard activity)
    - screenshot_TIMESTAMP.png (visual activity)



FILES OVERVIEW

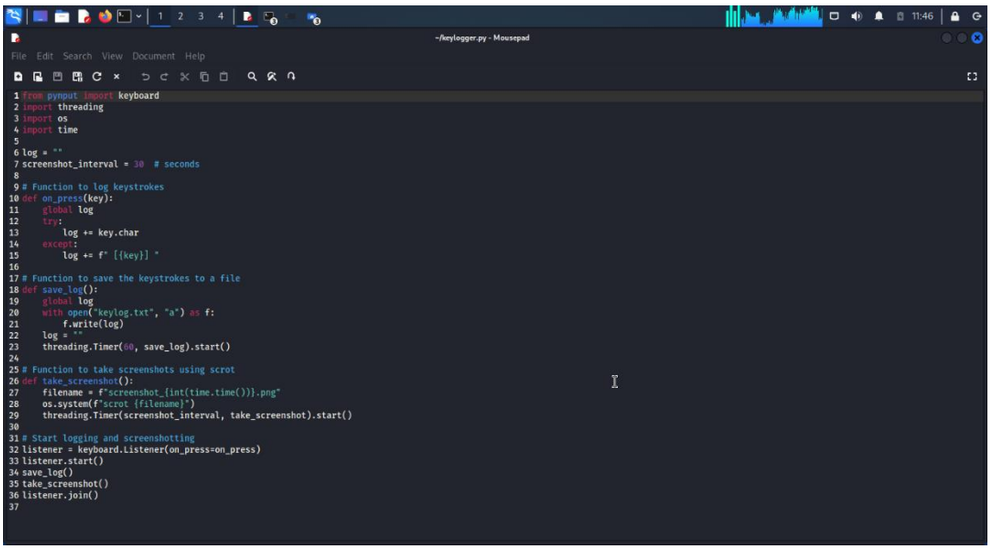
- **KEYLOGGER.PY**
  Main Python script that controls logging and screenshots.
- **KEYLOG.TXT**
  Sample log showing actual key usage.
  SAMPLE EXTRACT:

  [Key.enter] [Key.cmd] Keylogger [Key.ctrl] v [Key.backspace]

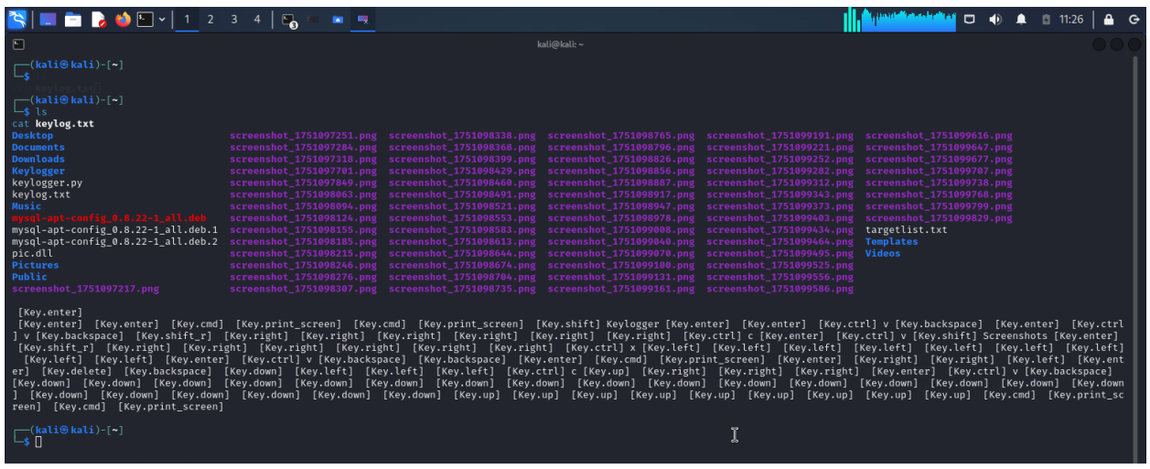
  [Key.right] [Key.ctrl] c [Key.ctrl] x [Key.left]

- **SAMPLE SCREENSHOT (BELOW):**
  Captured via scrot while the script was active.

KEYLOGGER.PY



KEYLOG.TXT & SCREENSHOT



CONCLUSION

The Keylogger Software developed in this project successfully achieves its intended purpose of ethically monitoring system activity. By logging keystrokes and capturing periodic desktop screenshots, it provides a clear understanding of how user behavior can be tracked for educational and security awareness purposes.

This tool offers a practical introduction to surveillance and logging in a controlled environment and demonstrates effective use of Python on Kali Linux.

SCREEN ACTIVITY MONITOR TOOL

OVERVIEW

Title: Screen Activity Monitor Tool

Category: Surveillance & Logging

Platform: Kali Linux

Language: Python 3

PURPOSE

To develop an ethical monitoring tool that captures desktop screenshots every X seconds for activity analysis. This tool can be used in educational, organizational, or research contexts to track screen usage behavior in a transparent and controlled manner.

FEATURES IMPLEMENTED

FEATURE	STATUS	DESCRIPTION
Periodic Screen Capture	Working	Screenshots are taken every 5 seconds using scrot and saved with timestamps.
Visual Activity Logging	Working	Screenshot files saved in the working directory with time-based filenames.

WORKING

- The script runs in the background.
- Every 5 seconds, it captures the current state of the desktop.
- Screenshots are saved as .png files named with a UNIX timestamp for clarity and uniqueness.
- Files are stored in the current folder and can later be uploaded securely to a cloud or email system (future enhancement).

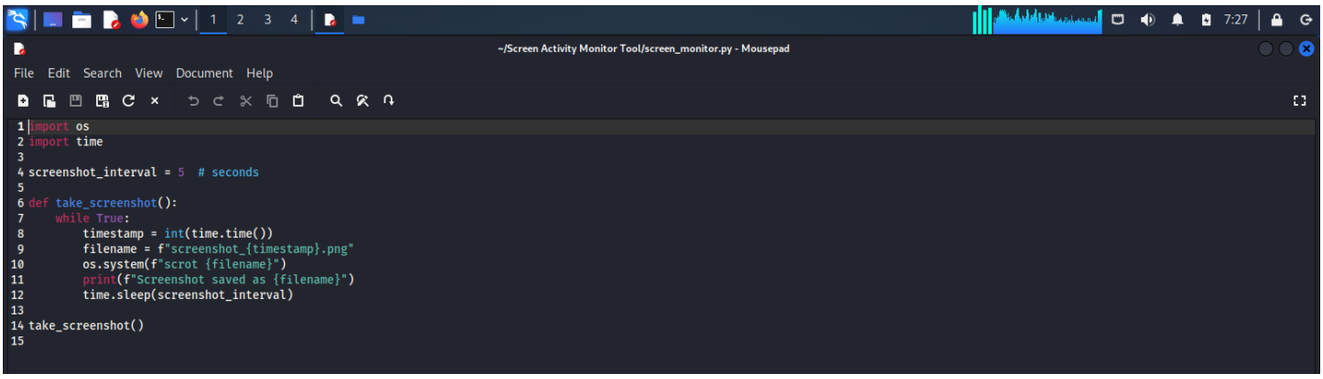
FILES OVERVIEW

Folder: Screen Activity Monitor Tool

Contains:

- **screen_monitor.py:** Main Python script that automates the screenshot functionality.
- **screenshot_.png:** Collection of auto-generated screenshots.
- Output from multiple runs, stored sequentially.

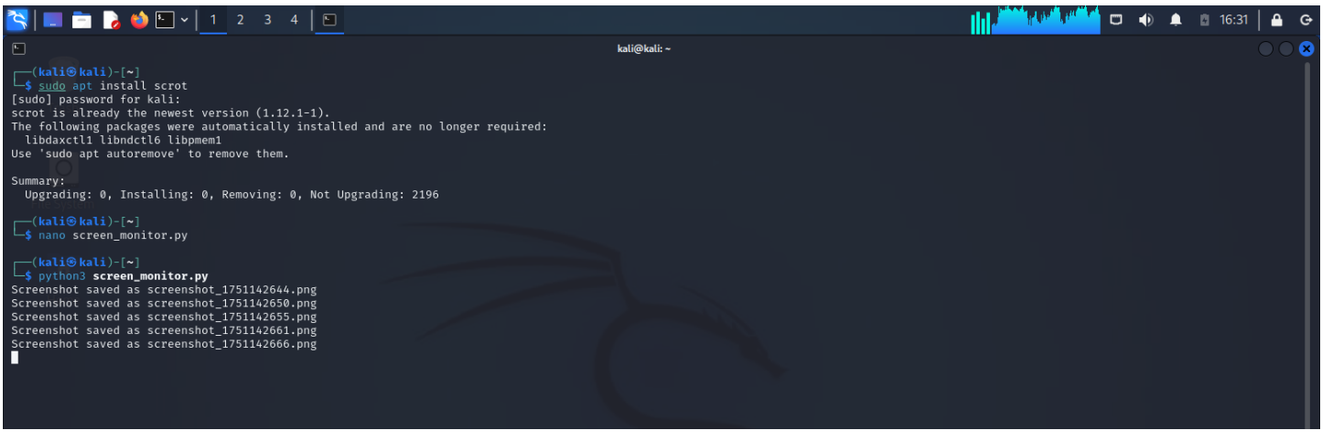
SCRIPT – SCREEN_MONITOR.PY



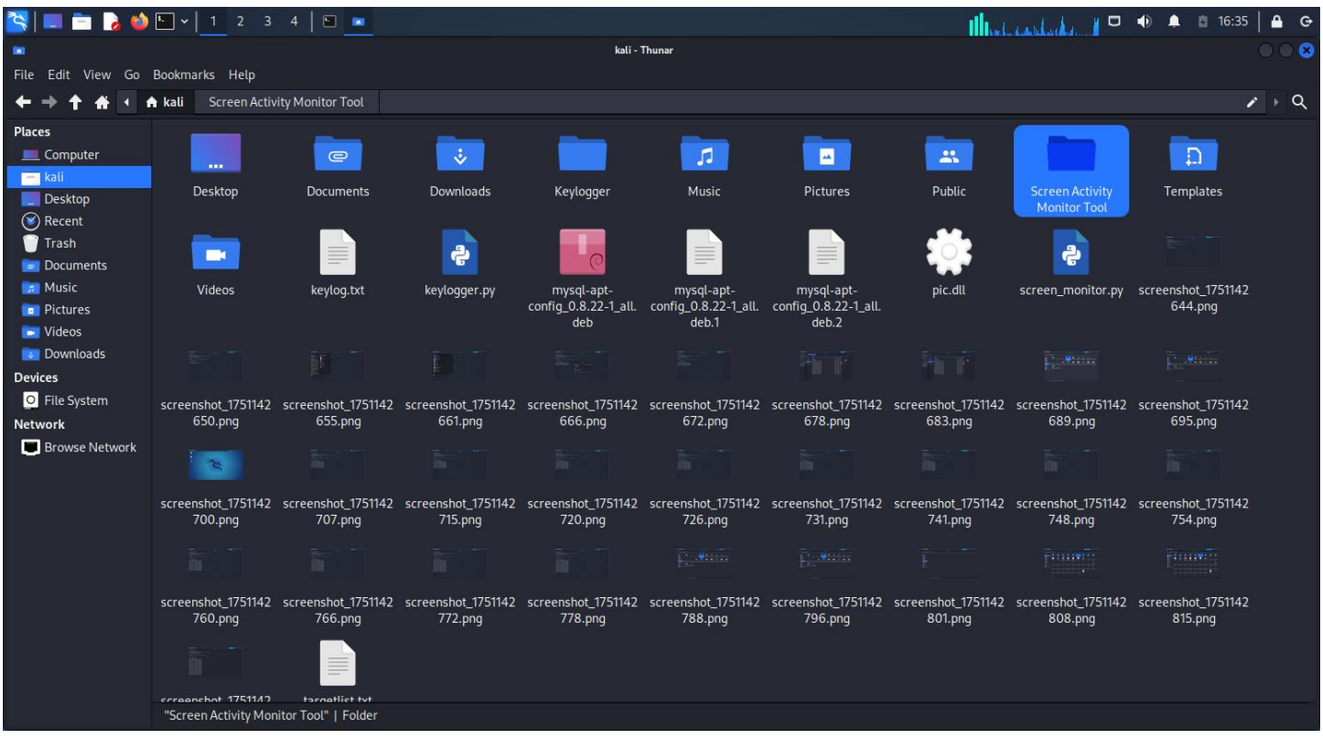
HIGHLIGHTS

- Uses the scrot utility to capture the desktop.
- Screenshots taken every 5 seconds (adjustable).
- Outputs are visible in the terminal for verification.

OUTPUTS



Each line confirms a screenshot was saved successfully.



All screenshots and the Python script are visible in the working directory.

CONCLUSION

The Screen Activity Monitor Tool successfully fulfills its core objective of capturing periodic desktop screenshots in an automated, efficient, and ethical manner. Designed using Python on the Kali Linux platform, it demonstrates the practical use of lightweight scripting and open-source utilities (like scrot) for system activity logging. By storing screenshots with timestamped filenames, the tool ensures organized visual logs that can be reviewed or archived as needed. Overall, this project not only meets its initial technical goals but also introduces a modular framework that can be expanded for more advanced monitoring systems, making it a valuable learning and demonstration tool in the field of system surveillance and automation.

USB ACTIVITY LOGGER (ETHICAL USE)

OVERVIEW

Title: USB Activity Logger (For Ethical Use Only)

Category: Surveillance & Logging

Platform: Kali Linux

Language: Python 3

PURPOSE

To ethically monitor USB port activity by tracking the connection and disconnection of USB devices. This tool is intended for awareness, auditing, and security monitoring in a controlled environment.

FEATURES IMPLEMENTED

FEATURE	STATUS	DESCRIPTION
USB Connection Logging	Working	Detects and logs when a USB device is connected.
USB Disconnection Logging	Working	Detects and logs when a USB device is safely removed.
USB Bind/Unbind Event Logging	Working	Tracks low-level USB binding events.
Timestamped Event Recording	Working	All activities are saved with date and time in usb_log.txt

WORKING

- The script runs in the background and listens to system USB events in real time.
- Upon detecting a USB-related event, it logs the type of event (ADD, REMOVE, BIND, UNBIND) along with the device path or ID.
- Every action is logged with a timestamp for precise tracking.

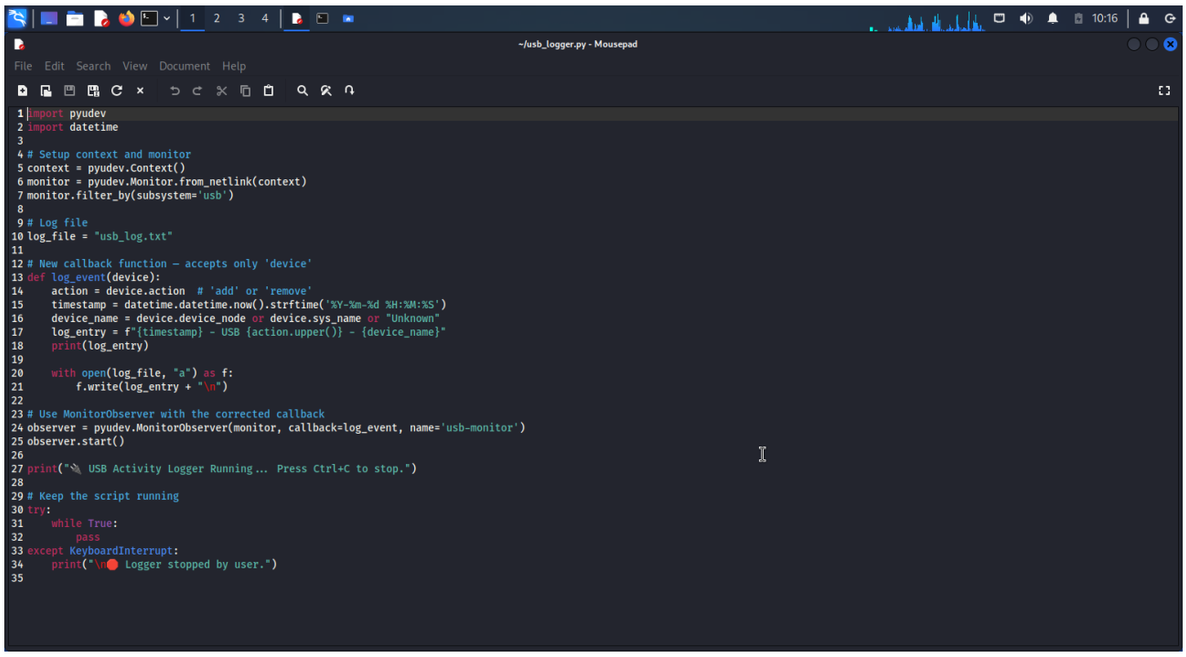
Logs are saved in:

- usb_log.txt — Records USB connection and removal details.

FILES OVERVIEW

USB_LOGGER.PY

Main Python script that initializes the USB monitor, defines the logging function, and handles real-time USB device tracking.



USB_LOG.TXT

Sample output log showing real-time USB activity captured during testing.

SAMPLE EXTRACT:

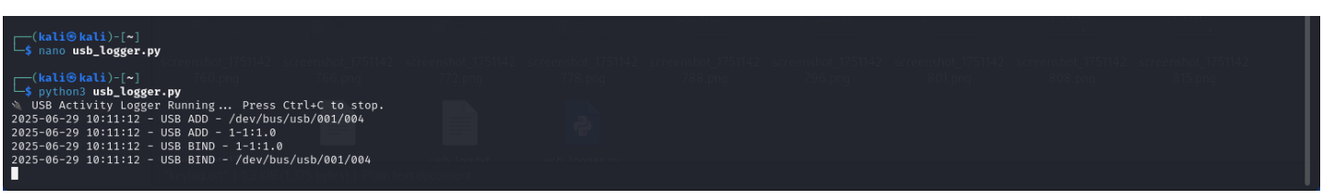
2025-06-29 10:11:12 - USB ADD - /dev/bus/usb/001/004

2025-06-29 10:11:12 - USB ADD - 1-1:1.0

2025-06-29 10:11:12 - USB BIND - 1-1:1.0

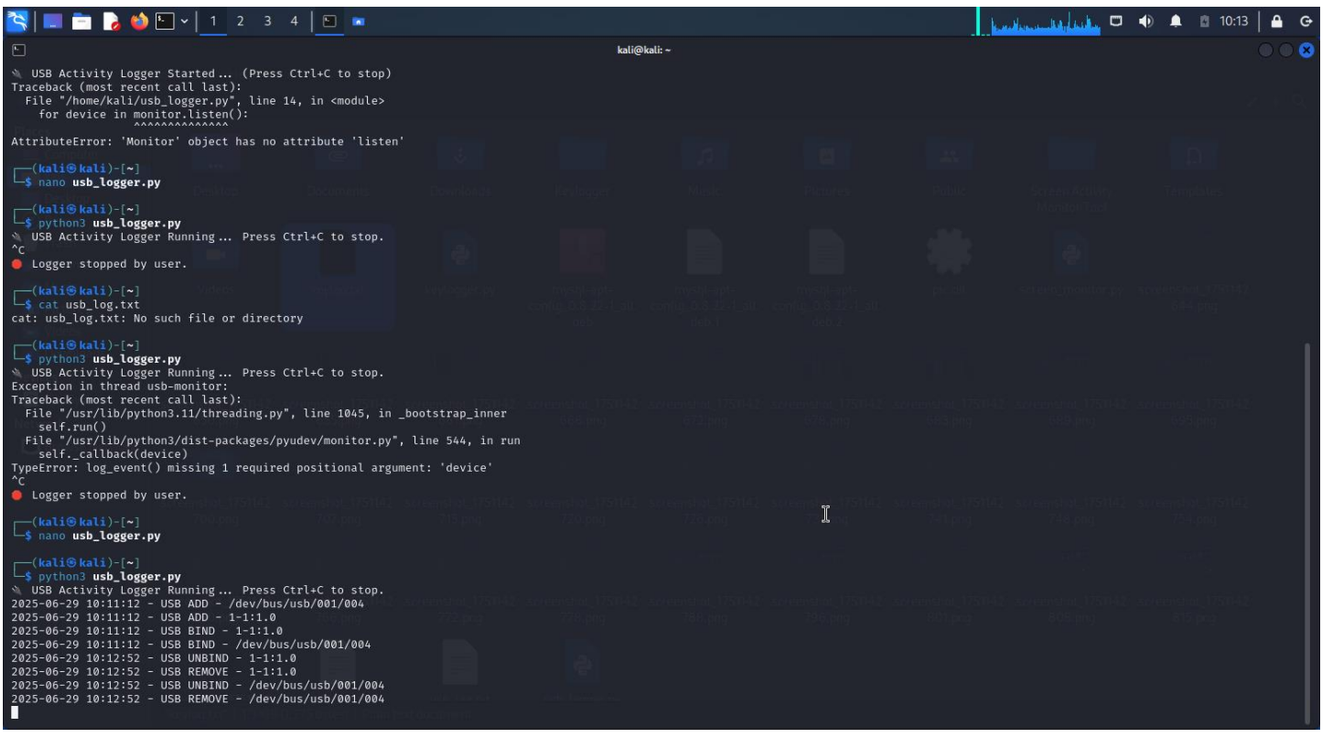
2025-06-29 10:12:52 - USB UNBIND - 1-1:1.0

2025-06-29 10:12:52 - USB REMOVE - 1-1:1.0



USB_LOG.TXT & SCREENSHOT

Log file and corresponding screenshot (usb.png) display successful detection of USB plug/unplug events.



CONCLUSION

The USB Activity Logger project effectively fulfills its purpose of ethically monitoring USB device interactions on a Kali Linux system. By capturing and logging real-time events such as USB connections, disconnections, and low-level binding actions, it offers a transparent and reliable method for tracking external device activity.

This tool provides a lightweight and efficient logging solution that can be used in academic, corporate, or cybersecurity environments to promote awareness, enforce device policies, and support forensic analysis. The implementation showcases practical use of Python and the pyudev library, demonstrating how open-source technologies can be leveraged to build useful surveillance tools in a controlled and responsible manner.

Overall, the USB Activity Logger stands as a strong example of how simple automation and monitoring can enhance system security and visibility without compromising ethical boundaries.