Project Report: Keylogger Implementation with Antivirus Detection

---

1. Problem Statement

Understanding how keyloggers work is crucial in cybersecurity to detect, mitigate, and defend against malicious software. This project aims to build a basic keylogger strictly for **educational and ethical purposes**, with enhanced awareness features such as **antivirus process detection**.

2. Objectives

- Create a keylogger that logs keystrokes to a local file.
- Implement real-time antivirus detection alerts for awareness.
- Educate users on ethical usage and responsible disclosure.
- Highlight how such tools are flagged or monitored by AV solutions.

3. Tools & Technologies

Tool/Library	Purpose
Python 3.x	Programming Language
pynput	Keyboard event listening
subprocess	To check system processes (AV scan)
os, datetime	File handling and timestamps

4. Functional Features

Keystroke Logging

- Captures each keystroke including special keys (Enter, Space, Backspace).
- Logs stored with timestamp in /logs/ folder.

Antivirus Detection

- Scans running processes for common AV tools (e.g., Windows Defender, Avast, Norton).
- Displays real-time warning in console if AV is detected.
- Logs detection result into the same log file.

Real-time Console Alerts

- ASCII box displays detected AV with warning message.
- Encourages ethical awareness and system-level visibility.

5. Output Structure

keylogger_project/

├── keylogger.py

├── README.md

├── DISCLAIMER.txt

└── logs/

  └── keylog_YYYYMMDD_HHMMSS.txt

Sample Log Output:

=== Keylogger Session Started ===

Timestamp: 2025-06-13 12:34:56.789

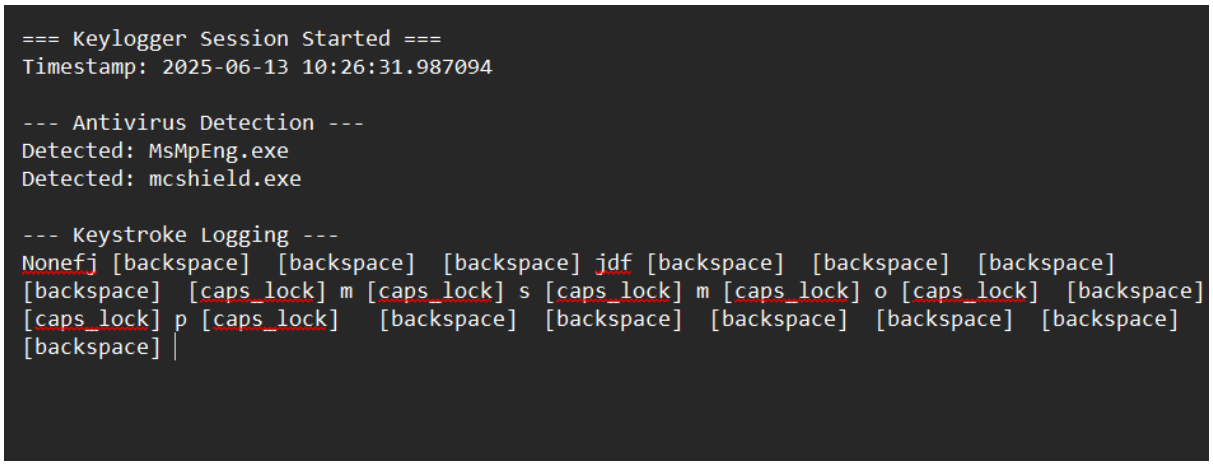
--- *Antivirus Detection* ---

Detected: MsMpEng.exe

--- *Keystroke Logging* ---

t e s t i n g [enter] h e l l o

6. Screenshots



7. Learning Outcomes

- Deep understanding of keylogger mechanics and limitations.
- Knowledge of how AV solutions detect runtime behaviors.
- Importance of ethical hacking and cybersecurity hygiene.
- Use of Python for system-level monitoring and file logging.

8. Future Improvements

- Encrypt the log file for privacy.
- Add GUI interface for better visualization.
- Add remote notification feature (only in ethical simulations).
- Implement cross-platform support (Linux, macOS).

9. Conclusion

This project showcases foundational knowledge of offensive security tools and, more importantly, how to detect and mitigate them. It's a strong demonstration of **security-first thinking**, with real-world relevance for positions in **cybersecurity**, **software engineering**, and **security research**.