International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 10 Issue 3, May-Jun 2026 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Keylogger Simulation and Detection with GUI: An awareness and Educational Approach

Chityala Ajay

Centurion University of Technology and Management, R.Sitapur, Odisha, India

ABSTRACT

In the Era of cybersecurity, cybersecurity threats are increasing rapidly. One of them is keyloggers, where many people of unaware of their mechanism. The biggest challenge is its stealth and data-theft capabilities. This paper presents the design and development of an educational tool that simulates keylogger attacks and provides a GUI-based anti-keylogger detection mechanism. Developed in Python, the tool uses the pynput library for simulating keylogging behavior and psutil for monitoring system processes. A tkinter-based graphical interface was integrated for real-time alerts and mitigation. The project aims to create awareness, provide hands-on learning, and demonstrate keylogger detection techniques in a controlled, ethical environment. Results indicate reliable keystroke capture and successful detection of the simulated keylogger, making the tool suitable for academic training and awareness programs.

KEYWORDS: *Keylogger, Anti-Keylogger, Python, GUI Process, pynput, Security Simulation, User Awareness.*

How to cite this paper: Chityala Ajay "Keylogger Simulation and Detection with GUI: An awareness and Educational Approach" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-10 | Issue-3, June 2026, pp.929-933, URL: www.ijtsrd.com/papers/ijtsrd105020.pdf

IJTSRD105020

I. INTRODUCTION

Keyloggers are silent threats to cybersecurity, capable of silently capturing keystrokes to steal data from the input we provide, which is used mostly for malicious purposes. They can also be used as monitoring tools in schools and colleges when used ethically. This research builds a foundational framework for understanding, simulating, and detecting keylogging behavior. The dual-purpose Python-based tool focuses on demonstrating how keyloggers work and how they can be identified in real time, helping students and cybersecurity professionals understand practical system-level threats.

II. PROBLEM STATEMENT

Keyloggers pose a very serious threat, yet many users remain unaware of their presence due to their stealthy operations and ability to invade systems while running continuously without detection. Currently, there is no user-friendly GUI-based keylogger detection tool that can be effectively used by individuals with only basic knowledge of operating systems. Additionally, there is a lack of user-friendly simulation tools that demonstrate how keyloggers work, which limits awareness and understanding among users.

III. OBJECTIVES OF THE RESEARCH

The primary objective of this research is to create awareness regarding the functioning of keyloggers and the defensive mechanisms that can be employed to counter their impact. Since keyloggers operate in stealth mode and often remain undetected by users, it is essential to develop educational approaches that help individuals understand how these malicious tools work and how they can be effectively defended against. Another objective is to explore and demonstrate the role of GUI-based applications in keylogger and anti-keylogger detection. A user-friendly interface is crucial to ensure that even individuals with limited technical knowledge can identify and respond to potential keylogger threats. This study aims to show how such applications can simplify detection and enhance accessibility. Furthermore, the research seeks to analyze the underlying processes used by keyloggers to capture keystrokes, as well as the defensive mechanisms designed to mitigate these actions. By examining both offensive and defensive processes, the research intends to provide a clearer understanding of attack strategies and the corresponding preventive measures,

thereby bridging the gap between awareness and practical defense.

IV. SIGNIFICANCE OF THE STUDY.

This study proposes a GUI-based application designed for user-friendly operation. The tool records keystrokes along with their respective timestamps and serves as an educational aid to create awareness about how the underlying process functions, thereby emphasizing its overall significance

V. RELATED WORK

Several researchers worked on the mitigation of keyloggers, and they proposed different models and techniques to address this threat. Kazi et al. (2023) emphasized that keyloggers represent one of the most dangerous forms of spyware due to their stealthy nature and ability to steal sensitive information such as banking credentials and passwords. Their study highlighted that while signature-based detection methods are commonly used, they often fail against newly emerging keylogger variants, as even minor modifications in the source code can bypass detection. To address this, behavior-based approaches have been explored, which focus on profiling application activities. However, these methods frequently suffer from false positives, since many legitimate applications, such as shortcut managers or keyboard utilities, demonstrate behaviors similar to those of keyloggers (Kazi et al., 2023).

Stefano et al. (2011) introduced **KLIMAX**, a behavior-based detection technique that profiles memory write patterns to identify keystroke-harvesting malware. Anith et al. (2011) proposed a client-level detection approach using traffic analysis and the TAKD algorithm. Similarly, Fu et al. (2010) applied the **Dendritic Cell Algorithm** to monitor API calls and distinguish keylogger processes from legitimate ones, achieving high detection rates with low false alarms. Le et al. (2008) employed **dynamic taint analysis** to detect kernel-level keyloggers, while Aslam et al. (2004) developed **Anti-Hook Shield**, which scans processes and executables to identify malicious hooks. Despite their contributions, these methods exhibit notable limitations such as high computational requirements, dependency on privileged access, and susceptibility to false positives (Stefano et al., 2011; Anith et al., 2011; Fu et al., 2010; Le et al., 2008; Aslam et al., 2004). there remains a critical gap in developing user-friendly, GUI-based detection and simulation tools that not only improve security but also create awareness among individuals with basic system knowledge.

VI. METHODOLOGY

The methodology follows a modular development approach. The keylogger simulation captures keystrokes using pynput and logs them into a hidden text file. The detection module uses psutil to monitor active processes and identify suspicious activity. A tkinter-based GUI enables user alerts and mitigation through process termination and log file deletion. The development was done on Windows OS, designed for educational use, with minimal system requirements.

The two components are:

Key-logger Simulation Module: Built by using pynput Library in python this module silently captures all the keystrokes and logs them into hidden log file, keylog.txt. It runs without any detection, simulating how real malware works (Python foundation Module-2023

Detection Module with GUI: Implemented using psutil library of python which continuously monitors the system processes and checks for the suspicious activities such as logging scripts and logfiles and presents real-time alerts through an accessible graphical interface dashboard.

Both the modules were tested in a windows 10/11 environment, reflecting common endpoints targeted by attackers (Prasad & Saxena, 2017).

VII. IMPLEMENTATION

The tool consists of a keylogger script and a detection script with a GUI. The keylogger records all keystrokes in real time, while the detector scans Python processes and known keylogging patterns. The GUI displays detection logs, alerts, and controls. Testing confirmed that the scripts operate efficiently, detecting logging activity in seconds and offering intuitive user controls.

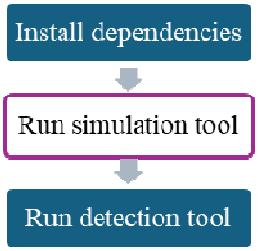

Image 1-process of installation

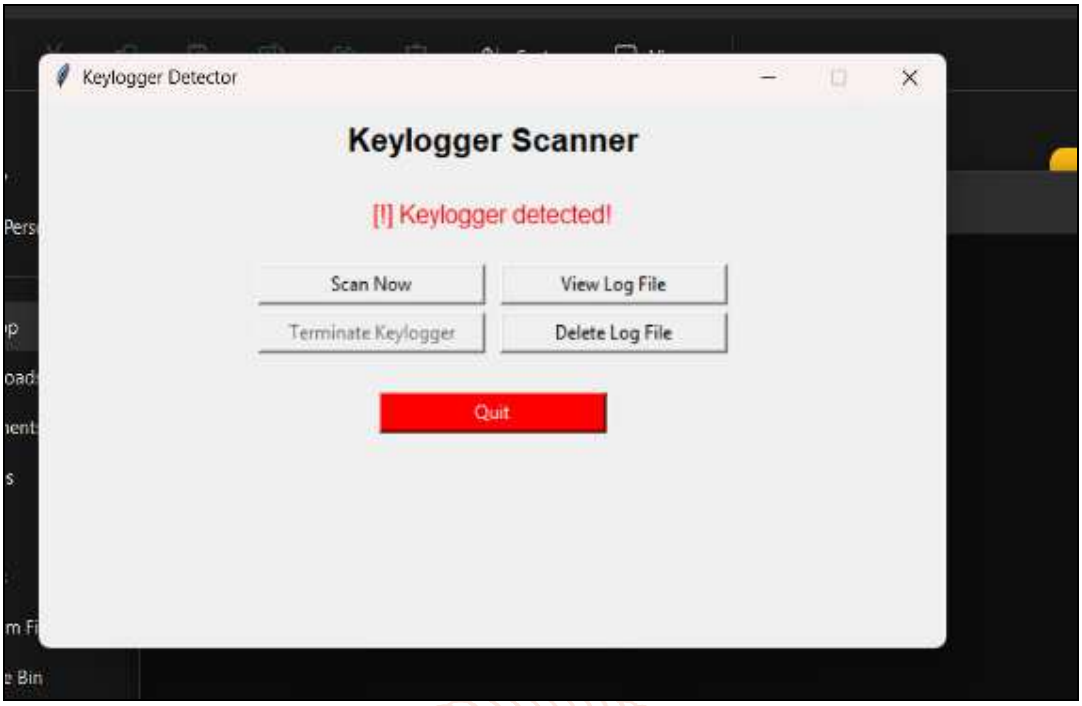

Image 2- Dashboard of Keylogger

VIII. RESULTS AND DISCUSSIONS

Testing revealed high detection accuracy and usability. All keypresses were recorded accurately, including modifier keys. The detection script flagged malicious behavior within seconds and had minimal false positives. Comparative testing against traditional antivirus tools showed improved performance for Python-based threats.

It shows high accessibility of User interaction in the GUI. Active processes and events of detection are immediately visible in the dashboard, allowing testers with limited technical knowledge to understand the behaviour as well as mitigation steps of the threat. Features like process termination and log deletion works efficiently and shows how defensive responses are done in real systems.

This work showed that behaviour-based system monitoring can detect keyloggers without usage of any threat database and signature based Anti-virus database, In the real-world, these results shows how accurately keyloggers can capture the keystrokes and log them in to the hidden file without any signs of detection, this results aid classroom practical learning effectively.

overall, the tool worked efficiently and gave results with accuracy, developing keylogger simulation with detection interface safe and realistic area to know about how keyloggers mechanisms works, the results also showed that this tool can bridge the gap between theoretical skills and practical implementation, making this tool which aids in providing awareness and academic training sessions. Screenshots validated the findings visually.

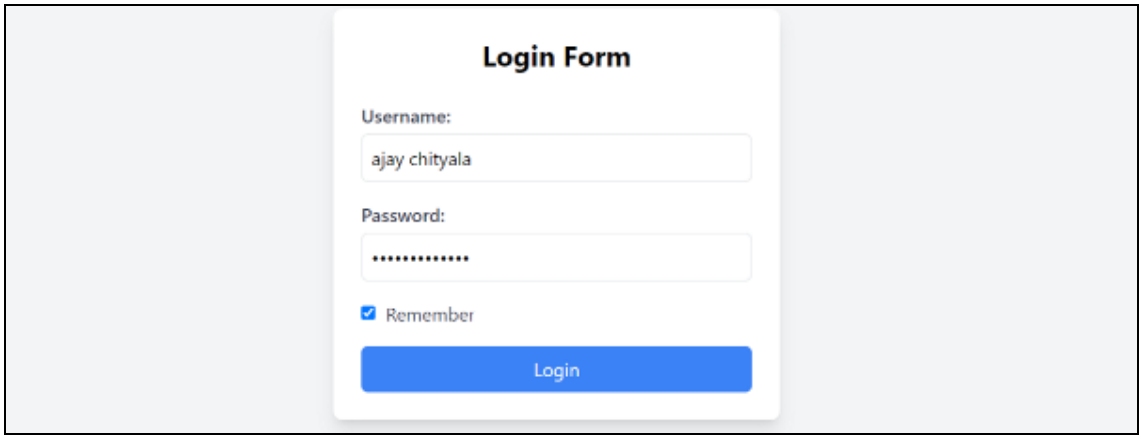

Image 3-testing website keystrokes capture

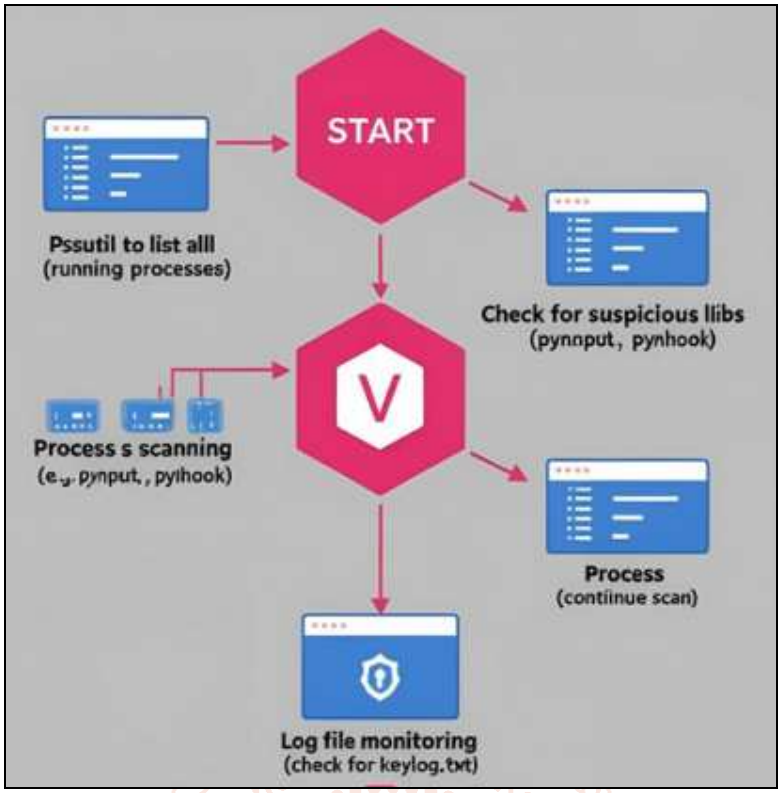

Image 4-process of scanning keyloggers

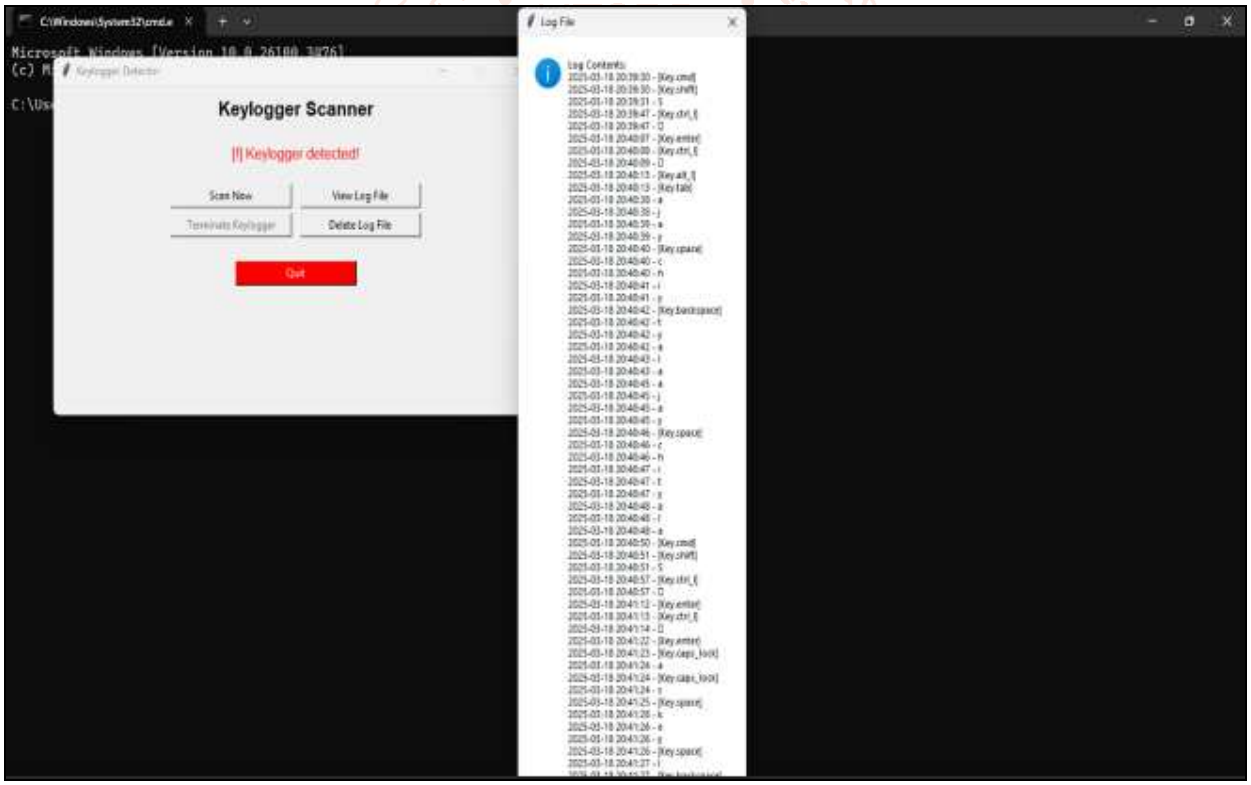

Image 5-captured keystores of the sample login page.

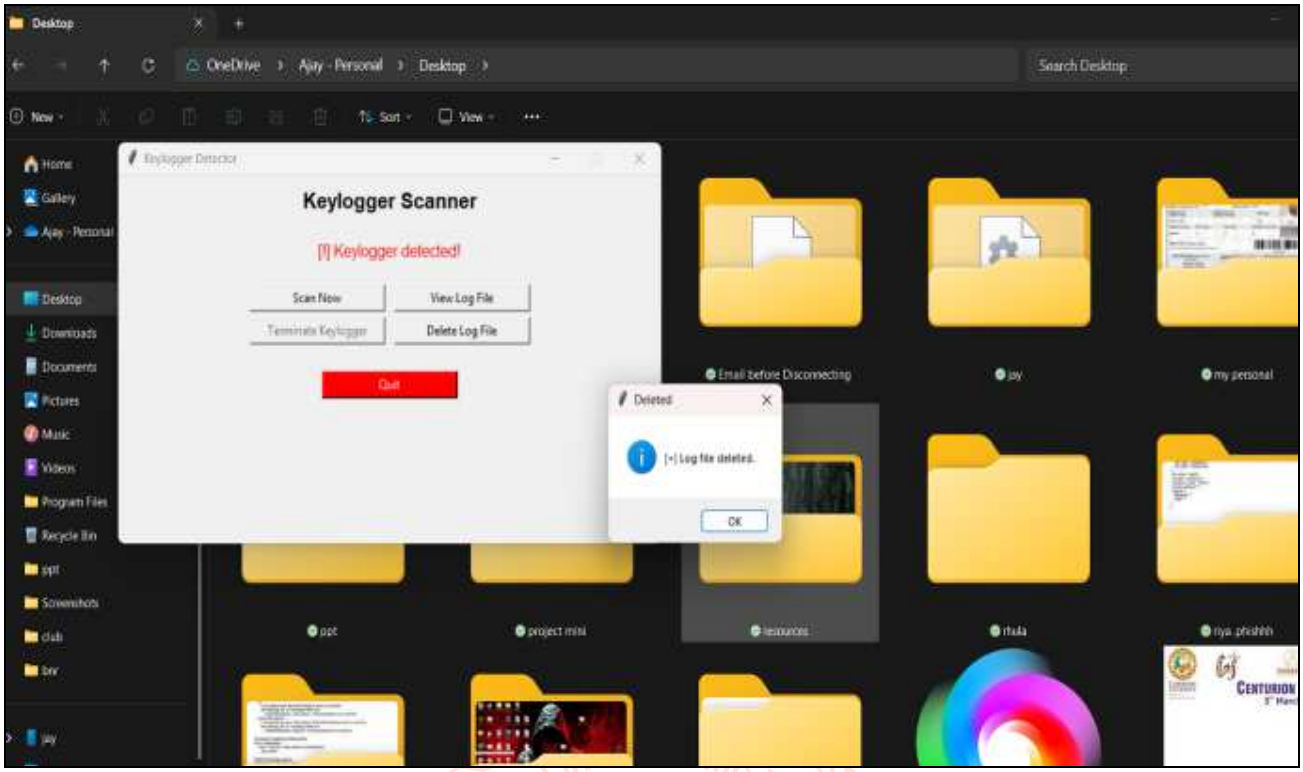

Image 6-captured keystores of the sample login page

IX. CONCLUSION AND FUTURE WORK

This project offers a practical and ethical framework for simulating and detecting keyloggers using Python. The tool promotes cybersecurity education through hands-on experience.

Future upgrades could include kernel-level detection, AI-based behavioral analysis, clipboard and screen logging detection, and macOS compatibility. The work contributes to educational cybersecurity by blending awareness and technical mitigation in a simple tool

REFERENCES

[1] Anith, S. S., & Anith, A. (2011). Detecting keyloggers based on traffic analysis with periodic behaviour. *PSG College of Technology, Coimbatore, India.*

[2] Aslam, M., Idrees, R. N., Baig, M. M., & Arshad, M. A. (2004). Anti-hook shield against the software key loggers. *Proceedings of the National Conference of Emerging Technologies.*

[3] Fu, J., et al. (2010). Detecting software keyloggers with the Dendritic Cell Algorithm.

[4] Kazi, A., Mungekar, M., Sawant, D., & Mirashi, P. (2023). Keylogger detection. *International Research Journal of Modernization in Engineering Technology and Science, 5*(4).

[5] Le, C. Y. D., Smart, T., & Wang, H. (2008). Detecting kernel-level keyloggers through dynamic taint analysis. *College of William & Mary, Department of Computer Science.*

[6] Anwar, F., & Khan. (2015). Software keyloggers using dendritic cell algorithm. Journal of Network and computer applications, 5(7), 1-6- https://doi.org/10.5120/21258-4200.

[7] Stefano, L., et al. (2011). KLIMAX: Profiling memory write patterns to detect keystroke-harvesting malware.