
CAPSTONE PROJECT

Keylogger and security

Presented By:

1. Arun Prabhu U S – College of Engineering Guindy – Department of Computer Science and Engineering

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

Problem Statement

In today's digital age, where cybersecurity threats loom large, one of the significant concerns is the proliferation of keyloggers, stealthy software tools designed to monitor and record keystrokes on a user's computer without their knowledge. Keyloggers pose a severe threat to individuals and organizations as they can capture sensitive information such as passwords, credit card details, and other personal data, leading to identity theft, financial loss, and privacy breaches.

Proposed Solution

- The proposed system aims to address the challenge of predicting the required bike count at each hour to ensure a stable supply of rental bikes. This involves leveraging data analytics and machine learning techniques to forecast demand patterns accurately. The solution will consist of the following components:
 1. **Logging Configuration:** Allow users to configure logging options, such as the frequency of log file rotation and maximum log file size, to customize logging settings based on user preferences.
 2. **User Interface Enhancements:** Improve the user interface by adding visual indicators and status updates to inform users about the keylogger's current state and activity, enhancing usability.

System Approach

In developing our keylogging application, we've adopted a systematic approach that integrates various tools and libraries to create a robust and user-friendly system. Here's how we've structured our approach:

- 1. Integration of Tools:** Our application seamlessly integrates the pynput and tkinter libraries to combine keylogging functionality with a user-friendly interface. This integration allows for efficient capturing of keystrokes while providing a smooth user experience.
- 2. Modular Design:** We've designed our codebase with a modular architecture, separating the keylogging functionality from the user interface components. This modular design promotes code reusability and maintainability, making it easier to enhance and extend the application in the future.
- 3. User Interaction:** The tkinter library serves as the backbone of our user interface, offering users a simple and intuitive way to interact with the keylogging system. Through the tkinter interface, users can effortlessly start and stop key recording sessions with just a few clicks.
- 4. Data Storage:** Keystrokes captured by our application are stored in both text and JSON file formats. This dual storage approach provides flexibility in accessing and analyzing the recorded data, catering to different user preferences and requirements.

Algorithm & Deployment

Our keylogging application utilizes a simple yet effective algorithm to capture keystrokes in real-time. Here's a breakdown of how it works:

Algorithm Overview:

- Keyboard Monitoring: We leverage the pynput library to monitor keyboard events, capturing key presses and releases as they occur.
- Event Handling: Upon detecting a key press or release event, the corresponding callback functions (on_press and on_release) are triggered to handle the event.
- Data Logging: Keystroke data is logged in two formats: a text file (key_log.txt) and a JSON file (key_log.json). This allows for easy storage and retrieval of captured keystrokes.

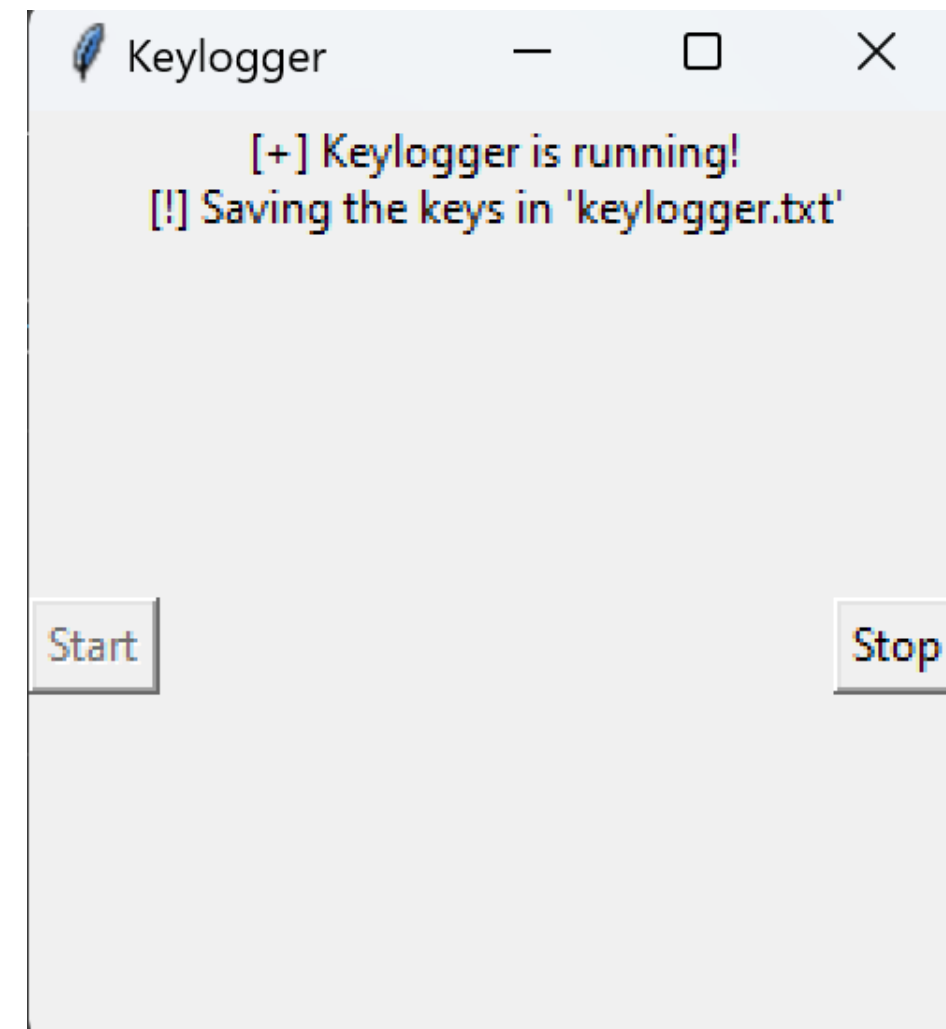
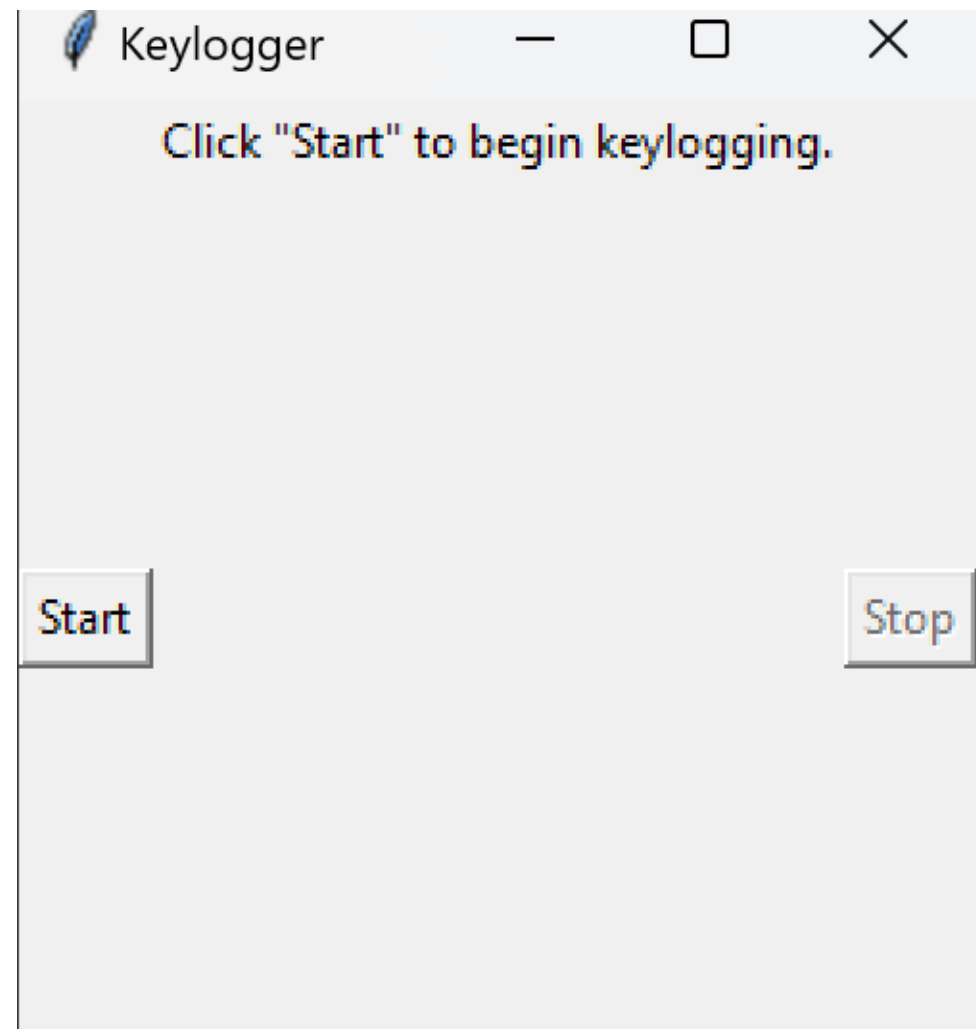
Deployment:

- Our application is deployed using the tkinter library to provide a user-friendly interface for starting and stopping the keylogging process.
- Users simply need to click the "Start" button to initiate the keylogging functionality, with the option to stop it at any time by clicking the "Stop" button.
- The generated log files (key_log.txt and key_log.json) are saved locally on the user's machine, ensuring ease of access and privacy of captured data.

By combining a robust algorithm with a user-friendly deployment approach, our keylogging application offers a seamless experience for capturing and logging keystrokes.

Result

The KeyLogger works flawlessly being able to register the keystrokes once the user starts the program from the GUI.



Conclusion

- Our keylogging application, built with Python's pynput and tkinter libraries, captures and logs keystrokes in real-time. Users can start and stop the keylogging process via a simple GUI interface. Keystroke data is saved in both text and JSON formats for easy access and analysis.

Future scope

Here's a glimpse into the future scope of the project:

1. **Enhanced User Interface:** Integrate advanced features into the GUI for better user interaction and customization options.
2. **Data Analysis Tools:** Develop tools to analyze the captured keystroke data, such as frequency analysis, pattern recognition, and anomaly detection.
3. **Remote Monitoring:** Implement remote monitoring capabilities to enable users to access and manage keylogging activities from any location.
4. **Security Measures:** Introduce encryption and authentication mechanisms to enhance data security and protect user privacy.
5. **Cross-Platform Compatibility:** Ensure compatibility with multiple operating systems and devices to cater to a wider user base.

With these future enhancements, the project can evolve into a more versatile and robust tool for keystroke monitoring and analysis.

References

- <https://www.youtube.com/watch?v=V5-LSvv2erk>
- <https://pypi.org/project/pynput/>
- <https://docs.python.org/3/library/tkinter.html>

THANK YOU