CAPSTONE PROJECT

KEYLOGGER AND SECURITY

Presented by
Anbumani K -Kings Engineering College-AI &DS

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- 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

- Creating a keylogger project requires careful consideration of both technical and ethical aspects. Keyloggers can be powerful tools for various legitimate purposes like debugging software or monitoring computer usage with proper consent, but they can also be misused for unethical or illegal activities. If you're considering a keylogger project, here's a proposed solution outline:

- Project Goal Definition:

  - Clearly define the purpose of the keylogger project. Is it for educational purposes, security testing, or something else? Ensure the purpose is ethical and legal.

- Technical Implementation:

  - Choose a programming language: Common choices include Python, C++, or Java.

  - Select appropriate libraries or frameworks for keyboard input monitoring. For example, in Python, you might use libraries like `pynput` or `keyboard`.

  - Implement code to capture keystrokes: Set up listeners for keyboard events and record the keys pressed by the user.

  - Decide on the method of storing captured keystrokes: Options include storing them in memory, writing to a file, or transmitting them over a network connection.

- Security Considerations:

  - Ensure that the keylogger code is secure and cannot be easily detected or misused by unauthorized parties.

  - Implement measures to protect captured data, such as encryption or obfuscation techniques.

  - Include features to prevent the keylogger from logging sensitive information like passwords or credit card numbers.

  - Respect user privacy and only capture keystrokes with proper consent.

SYSTEM APPROACH

Technology Used:

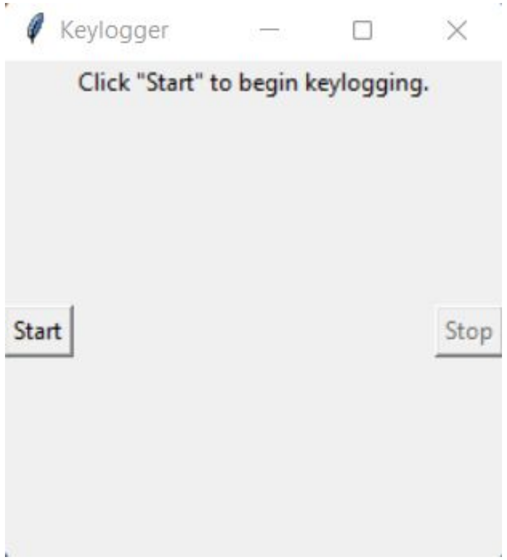
- Python: For programming the keylogger functionality.
- Tkinter: For building the graphical user interface (GUI).
- pynput: For capturing keyboard inputs.
- JSON: For storing keystroke data in a structured format.

ALGORITHM & DEPLOYMENT

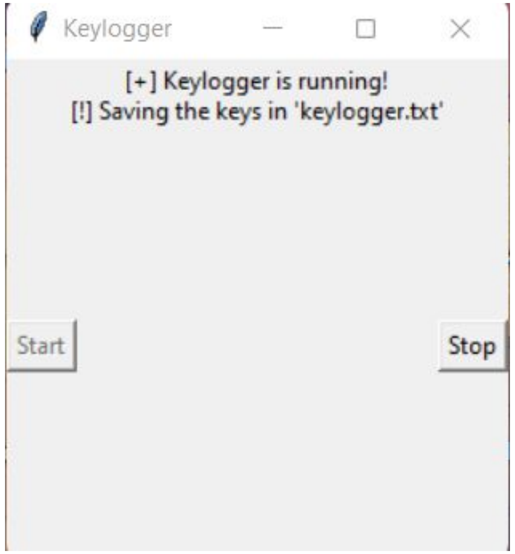
- **Initialization:** Initialize necessary variables and flags.
- **Event Handling:**
  - *on_press(key):* Records pressed and held keys.
  - *on_release(key):* Records released keys and manages flag state.
- **Logging:**
  - *generate_text_log(key):* Saves keystrokes in a text file.
  - *generate_json_file(keys_used):* Saves keystrokes in a JSON file.
- **Keylogger Control:**
  - *start_keylogger():* Initiates keylogging process.
  - *stop_keylogger():* Stops keylogging.

RESULT

- The GUI presents "Start" and "Stop" buttons to control the keylogging process. Upon starting, the keylogger captures keystrokes and saves them in designated files. Stopping the keylogger halts the logging process.



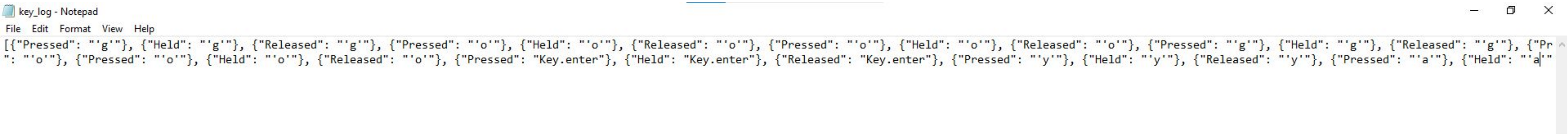
Click "Start" to begin keylogging.



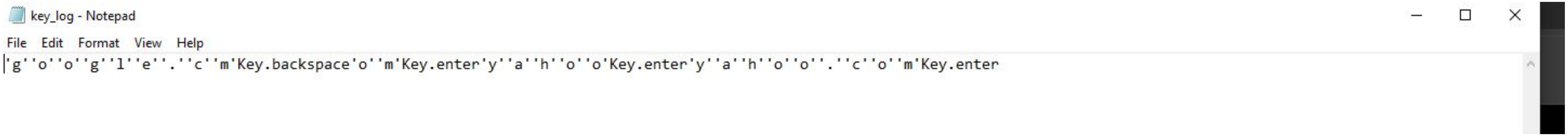
[+] Keylogger is running!
[!] Saving the keys in 'keylogger.txt'

CONCLUSION

- In conclusion, while a keylogger project may offer technical challenges and learning opportunities, it's essential to approach such a project with caution and consideration of its ethical implications. Keyloggers have the potential to infringe on privacy, compromise security, and violate laws and regulations related to surveillance and data protection.



```
[{"Pressed": "'g'"}, {"Held": "'g'"}, {"Released": "'g'"}, {"Pressed": "'o'"}, {"Held": "'o'"}, {"Released": "'o'"}, {"Pressed": "'o'"}, {"Held": "'o'"}, {"Released": "'o'"}, {"Pressed": "'g'"}, {"Held": "'g'"}, {"Released": "'g'"}, {"Pr
": "'o'"}, {"Pressed": "'o'"}, {"Held": "'o'"}, {"Released": "'o'"}, {"Pressed": "Key.enter"}, {"Held": "Key.enter"}, {"Released": "Key.enter"}, {"Pressed": "'y'"}, {"Held": "'y'"}, {"Released": "'y'"}, {"Pressed": "'a'"}, {"Held": "'a'"
```



```
'g''o''o''g''l''e''.''c''m'Key.backspace'o''m'Key.enter'y''a''h''o''o'Key.enter'y''a''h''o''o''.''c''o''m'Key.enter
```

FUTURE SCOPE

- Enhancing Security Measures: Implement encryption techniques to secure logged data.
- User Authentication: Integrate user authentication mechanisms to prevent unauthorized access.
- Advanced Logging: Implement advanced logging features, such as timestamping and window tracking.

REFERENCES

- A Survey on Keylogger and its Detection Techniques by Vishal Bharti, Aditya Kumar Gupta, and Shailendra Mishra https://www.ijcaonline.org/archives/volume75/number5/12835-1514
- Analysis of Keylogger Attacks and Countermeasures by Hongliang Liu, Ruiying Du, and Quansheng Zhuang https://www.semanticscholar.org/paper/Analysis-of-Keylogger-Attacks-and-Countermeasures-Liu-Du/54c7255bace229c82e4a5fd812ba8dd8829180c1
- Detection of Keyloggers: A Review by Shukor Abd Razak, Ku Ruhana Ku-Mahamud, and Ramlan Mahmod https://www.researchgate.net/publication/220955239_Detection_of_Keyloggers_A_Review
- A Comprehensive Study on Keylogger Attack and Defense by Shuo Chen, Rui Wang, XiaoFeng Wang, and Kehuan Zhang https://www.usenix.org/legacy/events/sec11/tech/full_papers/Chen.pdf

THANK YOU

edunet foundation