Part 1: Python Keylogger

Contents

Time required: 30 minutes

A keylogger is a program that can record keystrokes, mouse movements and screenshots. They can store logs locally, send them to email, or to a remote server. A well written keylogger program can do all that without the user knowing.

NOTE: Please program this series of tutorials in Windows and Linux.

Windows Install keyboard Library

1. In Windows: **pip install keyboard**

Linux Install keyboard Library

1. Open a terminal session.
2. Update software package lists: **sudo apt update**
3. Install pip3: **sudo apt install python3-pip**
4. Determine your version of Python: **python3**
5. Press **CTRL D** to exit the Python prompt.
6. Install the **keyboard** library.
    a. **Python 3.10 and lower**: **sudo pip3 install keyboard**
    b. **Python 3.11: sudo pip3.11 install keyboard**

Key Logger 1 in Windows

Over the next several assignments, we are going to build a keylogger in Python using the **keyboard** library. The keylogger program will work in Windows and Linux. We will start with Windows.

1. Go to a command prompt to install **keyboard**

```
# Windows
pip install keyboard
```

2. Create a new Python program named: **frog_1.py**

```
1   #!/usr/bin/env python3
2   """
3       Name: frog_1.py
4       Author:
5       Created:
6       Purpose: Capture keystrokes using keyboard library
7   """
8   # Windows: pip install keyboard
9   # Linux: sudo pip3.11 install keyboard
10  import keyboard
11  import os
```

The first line is called a **shebang**. If you have more than one version of Python, Linux uses this line to locate the first installation of python3 in its environment. This line has no effect in Windows.

Import the **keyboard** library. This module allows Python to capture local keystrokes.

```
14  class KermitTheFrog():
15      def __init__(self):
16          print("Kermit the Frog Started . . . ribbit ribbit ribbit")
17          # Create a keyboard listener object
18          # which will listen for a keyboard on_release event
19          # When a key is released,
20          # that key is passed to the process_key method
21          keyboard.on_release(callback=self.process_key)
22          # The main program thread waits for a key release
23          keyboard.wait()
```

The class init function sets up a **keyboard.wait()** object. This object waits until a key is released. When the key is released, that key is passed as an argument to the **process_key()** function.

```
25  # ------------------------ PROCESS KEY RELEASE ----------------------------#
26      def process_key(self, event):
27          """Callback function whenever a key is pressed"""
28          # Convert each key release to a string
29          name = event.name
30          # Print the key to the console
31          print(name)
32          # Press the Esc key to exit the program
33          if name == "esc":
34              print("Exiting Kermit the Frog")
35              os._exit(0)
36
37
38  # Create program object and start program
39  kermit_the_frog = KermitTheFrog()
```

The **process_key()** function takes in the current key press as a parameter and prints it to the console. If the **Esc** key is pressed, the program exits.

Run the program in both operating systems. You can type anywhere on your computer. Each keystroke will be logged.

Example run in Windows (Click outside of the program console). Type in some keystrokes:

```
Kermit the Frog Started . . . ribbit ribbit ribbit
k
e
y
l
o
g
g
e
r
esc
Exiting Kermit the Frog
```

Linux Keylogger

1. Change to your Code folder.

1. Using your favorite Linux editor → create a Python file named **frog_1.py**
2. Copy and paste the code into the editor.
3. Save the file.
4. At a terminal prompot use the following command to run the program.

```
python3 frog_1.py
```

5. Click outside of the terminal prompt. Type in some keystrokes.

Example run in Linux:

Assignment Submission

1. Attach all program files.
2. Attach a screenshot from Windows and Linux of your results.
3. Submit the assignment in BlackBoard.