HY-457

Assignment 3

Due: 17/05/2017

In this assignment you are going to implement a linux keylogger in C. The purpose of this assignment is educational and its aim is to provide you the opportunity to get familiar with writing and defending against a malicious software.

What is a keylogger?
A keylogger is a software program or hardware device that is used to monitor and log every keystroke a user types into a computer keyboard. This is usually done with malicious intent in order to collect account information, credit card numbers, usernames, passwords, and other private data.

However, the use of keylogger is not always malicious. Parents can monitor their children's online activity or law enforcement may use it to analyze and track incidents linked to the use of personal computers. Moreover, employers can make sure their employees are working instead of surfing
the web all day.

TASK A

[Find your keyboard device]
In order to monitor your keyboard keystrokes, you must find and hook at your keyboard device (root privileges required).

You can find your connected devices in Linux using the command:

$ **ls -al /dev/input/**

You will see many event files.
All these are connected devices in your computer (keyboard, mouse, camera, touchscreen, etc.). Each device has a different event number (e.g event3). How will you know which of these events is your keyboard device? This information is not standard and defers from one pc to another.
The easier way to identify the event which is connected with your keyboard is to look at /proc/bus/input/devices :

$ **cat /proc/bus/input/devices**

You will see information for each device like this:

I: Bus=0011 Vendor=0001 Product=0001 Version=ab41
N: **Name**="AT Translated Set 2 **keyboard**"
P: Phys=isa0060/serio0/input0
S: Sysfs=/devices/platform/i8042/serio0/input/input3
U: Uniq=
H: **Handlers**=sysrq kbd **event3** leds
B: PROP=0
B: EV=120013
B: KEY=402000000 3803078f800d001 feffffdfffeffff fffffffffffffffe
B: MSC=10
B: LED=7

In order to find which of them is your keyboard, you should take a look at the field **Name** and search for the keyword "**keyboard**" or your keyboard's "brand/model" in some rare cases.
When you find your keyboard device you must go on to the field **Handlers.**
In this example our keyboard device is the **event3**.

Task B

[Input Events]

- https://github.com/torvalds/linux/blob/master/include/uapi/linux/input.h
- https://www.kernel.org/doc/Documentation/input/input.txt
- https://www.kernel.org/doc/Documentation/input/event-codes.txt

```
/* The input_events structure layout --> #include "linux/input.h" */
struct input_event {
        struct timeval time;
        unsigned short type;
        unsigned short code;
        unsigned int value;
};
```

- **time:** is the timestamp, it returns the time at which the event happened.
- **type:** is an event type (`EV_KEY`, `EV_ABS`, `EV_REL`, ...), for example EV_REL for relative moment, EV_KEY for a keypress or release.
- **code:** is event code, could be either a key code when using `EV_KEY`, or an axis for `EV_ABS` and `EV_REL`, for example REL_X or KEY_BACKSPACE.
- **value:** is the value the event carries. May be 1 (press) or 0 (release) for `EV_KEY`, or any values for others (positive integer for `EV_ABS`, signed integer for `EV_REL`, etc…).

Event Types:

- EV_KEY type represents key press and release events,
- EV_REL type represents relative axis events (such as mouse movements),
- EV_ABS type represents absolute axis events (such as touchscreen movements)

In order to handle an input event you must follow the below steps:

1) Create a file descriptor for the keyboard.
   (remember you need root privileges “sudo”, to read the keystrokes)
   int kb = open("/dev/input/event3", O_RDONLY);
2) Check for keyboard events
   a) Create an event **struct input_event event;**
   b) Read from the file descriptor ***read(int fd, void *buf, size_t nbytes);***
   c) Check the event.type
   d) Check for keypress
3) Print the pressed key to stdout or write it to a file.
   Your program must be able to handle two arguments. With -s the pressed keys appear to the screen (stdout) while -f file stores the keystrokes to the specified file.
   e.g **$ sudo ./keylogger -s**
   $ sudo ./keylogger -f output.txt
4) Print/write each keystroke using the provided **keycodes** arrays.
5) Handle when the Shift key (right or left Shift key) is pressed and use the corresponding **shifted_keycodes** array in this case.
6) Your program must be terminated in two ways:
   a) If the **ESC** key is pressed (or)
   b) If an interrupt signal **SIGINT**(Ctrl + C) is received a use a signal handler

Task C

[Questions]

1. How could we make this program run at the background all the time?
   Run your program at the background and check if all works as expected.

2. How could we protect our bank credentials from a keylogger?

Submit

A folder named assign_3_yourAM, containing all the source code of your tool, a README file and a Makefile. The README file should briefly describe your tool and contain the answers of **Task C**.