

Keyloggers

Patrick Jayoma, Parker Bath, Amaya
Alviz, and Kyle Traverse





Project Description

- What is a keylogger?
 - usually a monitoring type of software to record keystrokes made by a user.
 - Can be used for accessing private information/ confidential corporate data



Why the team was interested?

- Pros and Cons

Pros

- A legitimate use would be to log keystrokes for auditing purposes, eg: logging what people do in a sensitive environment.
- Highlights weaknesses in IT security

Cons

- private information can be exposed if used in the wrong hands
- can cause an organisation to breach major pieces of legislation

- Getting a taste of what hackers do on a small level
- Learning how a keylogger lives and works in your os
- What to look for and how to possibly identify a keylogger.



Building Kernel 1

- What research was done?
 - Online sources
 - [Make a Linux Based Keylogger. Understand how keylogger works. How the... | by Nayan Das | Medium](#)
 - [https://www.bcs.org/content-hub/keyloggers-pros-and-cons/](#)



Building Kernel 2

- What problems you experienced?
 - Conflicting Schedules
 - How keyboard events are handled
 - Figuring out which key code maps to what character.

```
/*  
 * Keys and buttons  
 *  
 * Most of the keys/buttons are modeled after USB HUT 1.12  
 * (see http://www.usb.org/developers/hidpage).  
 * Abbreviations in the comments:  
 * AC - Application Control  
 * AL - Application Launch Button  
 * SC - System Control  
 */  
  
#define KEY_RESERVED      0  
#define KEY_ESC           1  
#define KEY_1             2  
#define KEY_2             3  
#define KEY_3             4  
#define KEY_4             5  
#define KEY_5             6  
#define KEY_6             7  
#define KEY_7             8  
#define KEY_8             9  
#define KEY_9             10  
#define KEY_0             11  
#define KEY_MINUS         12  
#define KEY_EQUAL         13  
#define KEY_BACKSPACE     14  
#define KEY_TAB           15  
#define KEY_Q             16  
#define KEY_W             17  
#define KEY_E             18  
#define KEY_R             19  
#define KEY_T             20  
#define KEY_Y             21  
#define KEY_U             22  
#define KEY_I             23  
#define KEY_O             24  
#define KEY_P             25  
#define KEY_LEFTBRACE    26  
#define KEY_RIGHTBRACE   27  
#define KEY_ENTER         28
```



Building Kernel 3

- How were these problems mitigated?
 - Coordinated better
 - `cat /proc/bus/input/devices`

```
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/input2
U: Uniq=
H: Handlers=sysrq kbd event2 leds
B: PROP=0
B: EV=120013
B: KEY=402000000 3803078f800d001 feffffdffffffffff ffffffff
B: MSC=10
B: LED=7
```



Building Kernel 3 cont.

- How problems were mitigated
 - `ls -l /dev/input`

```
kapp@kapp:~$ ls -l /dev/input/
total 0
drwxr-xr-x 2 root root 80 Apr 21 12:11 by-id
drwxr-xr-x 2 root root 180 Apr 21 12:11 by-path
crw-rw---- 1 root input 13, 64 Apr 21 12:11 event0
crw-rw---- 1 root input 13, 65 Apr 21 12:11 event1
crw-rw---- 1 root input 13, 66 Apr 21 12:11 event2
crw-rw---- 1 root input 13, 67 Apr 21 12:11 event3
crw-rw---- 1 root input 13, 68 Apr 21 12:11 event4
crw-rw---- 1 root input 13, 69 Apr 21 12:11 event5
```

- `/usr/include/linux/`
 - `nano input-event-codes.h` (for keys and buttons mapping)
- Adding `<"/linux"/.>` to the file path

Demonstration



What we learned

- Chmod permission codes
- Input devices
- User events
 - Keyboards



Thank you and questions