LOGGER

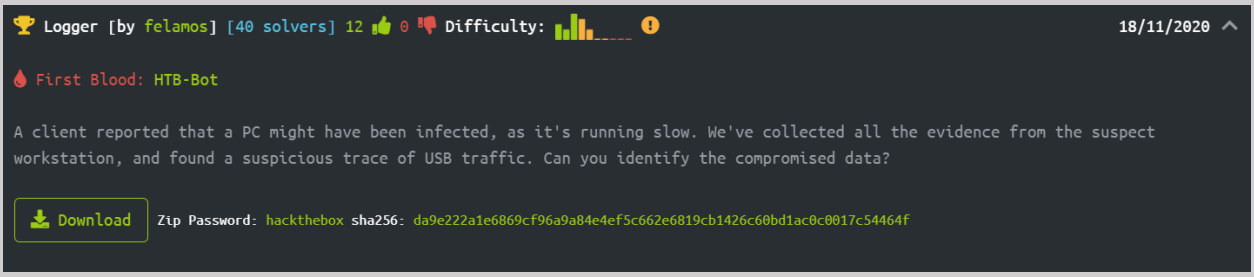
Hack the Box writeup

2/19/2021

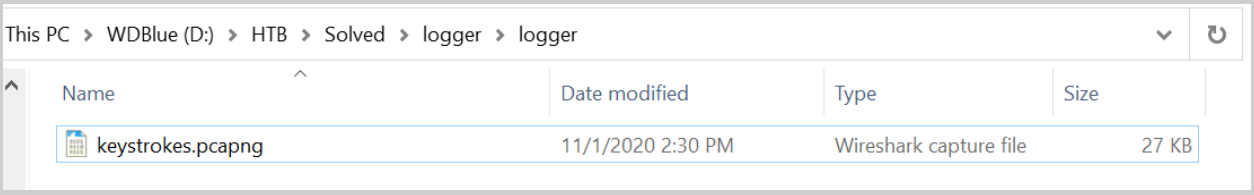
Contents

0x6B

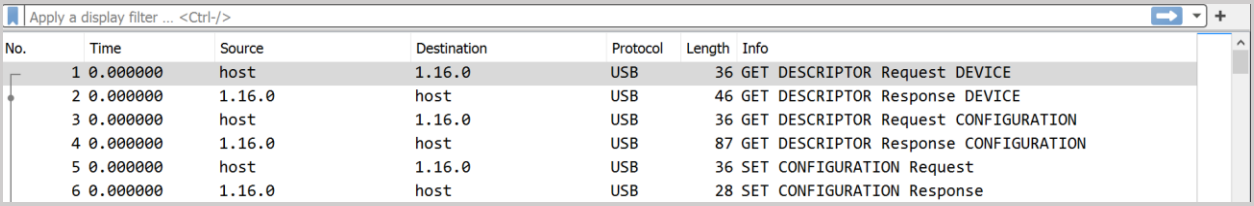
Challenge



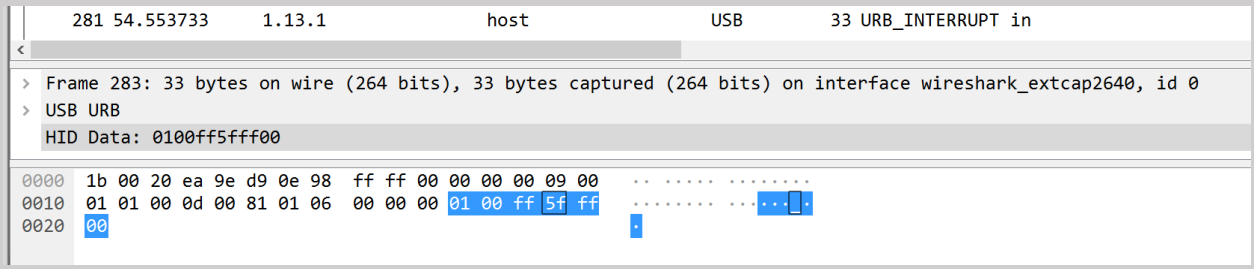
Process

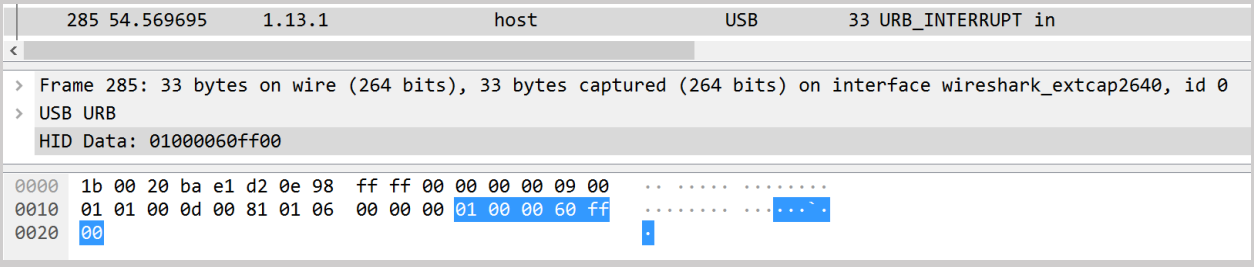


We start out with a pcap file that contains some USB info on it.



Scrolling through this, we don't see much change in the overview, but if we look at some of the data sending to the host, we see very slight changes in the HID Data Field





Unfortunately, it does not look like the data given is actually the keystrokes that are being sent, but the hex codes should be accurate.

I looked up a USB HID Code Table and came across this documentation:

https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf

Searching for keyboard information lead me to pg 53

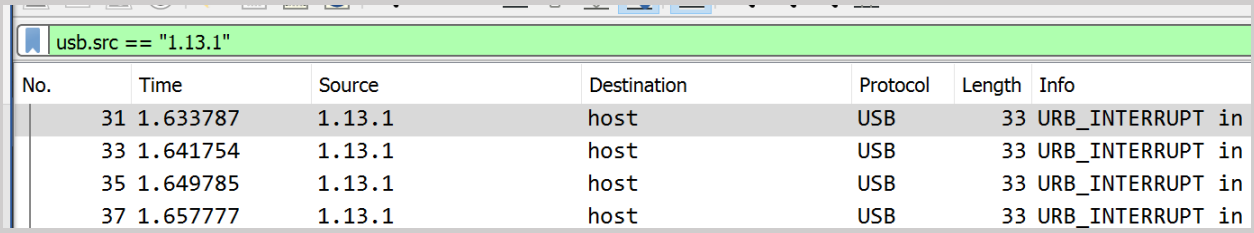
Table 12: Keyboard/Keypad Page

Usage ID (Dec)	Usage ID (Hex)	Usage Name	Ref: Typical AT-101 Position	PC-AT	Mac	UNIX	Boot
0	00	Reserved (no event indicated)[9]	N/A	√	√	√	4/101/104
1	01	Keyboard ErrorRollOver[9]	N/A	√	√	√	4/101/104
2	02	Keyboard POSTFail[9]	N/A	√	√	√	4/101/104
3	03	Keyboard ErrorUndefined[9]	N/A	√	√	√	4/101/104
4	04	Keyboard a and A[4]	31	√	√	√	4/101/104
5	05	Keyboard b and B	50	√	√	√	4/101/104
6	06	Keyboard c and C[4]	48	√	√	√	4/101/104
7	07	Keyboard d and D	33	√	√	√	4/101/104
8	08	Keyboard e and E	19	√	√	√	4/101/104
9	09	Keyboard f and F	34	√	√	√	4/101/104
10	0A	Keyboard g and G	35	√	√	√	4/101/104
11	0B	Keyboard h and H	36	√	√	√	4/101/104
12	0C	Keyboard i and I	24	√	√	√	4/101/104
13	0D	Keyboard j and J	37	√	√	√	4/101/104
14	0E	Keyboard k and K	38	√	√	√	4/101/104
15	0F	Keyboard l and L	39	√	√	√	4/101/104
16	10	Keyboard m and M[4]	52	√	√	√	4/101/104
17	11	Keyboard n and N	51	√	√	√	4/101/104
18	12	Keyboard o and O[4]	25	√	√	√	4/101/104
19	13	Keyboard p and P[4]	26	√	√	√	4/101/104
20	14	Keyboard q and Q[4]	17	√	√	√	4/101/104

This looks perfect, now I just need to check and see if this data is actually what I think it is.

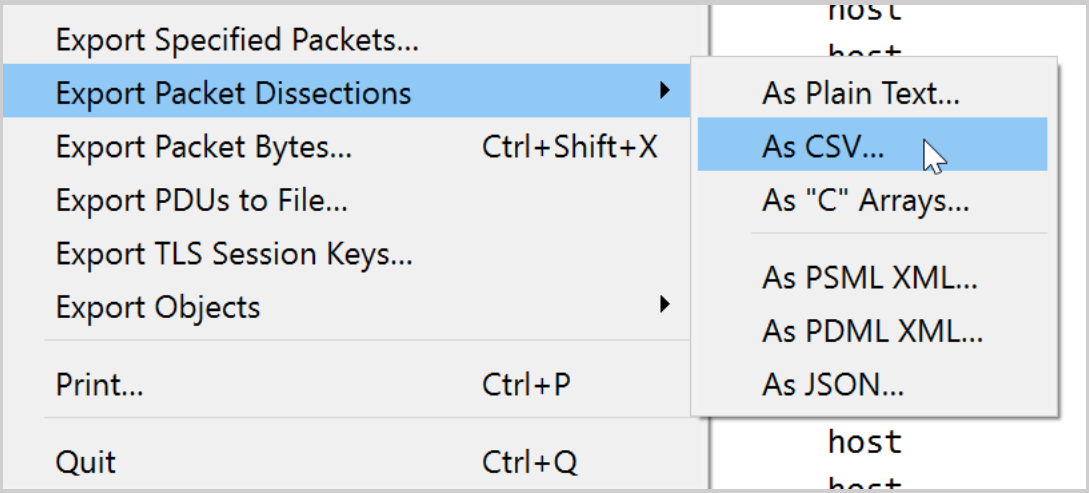
I filtered down the pcap to only show items from usb.src == "1.13.1"

This way I reduce the noise I am getting.



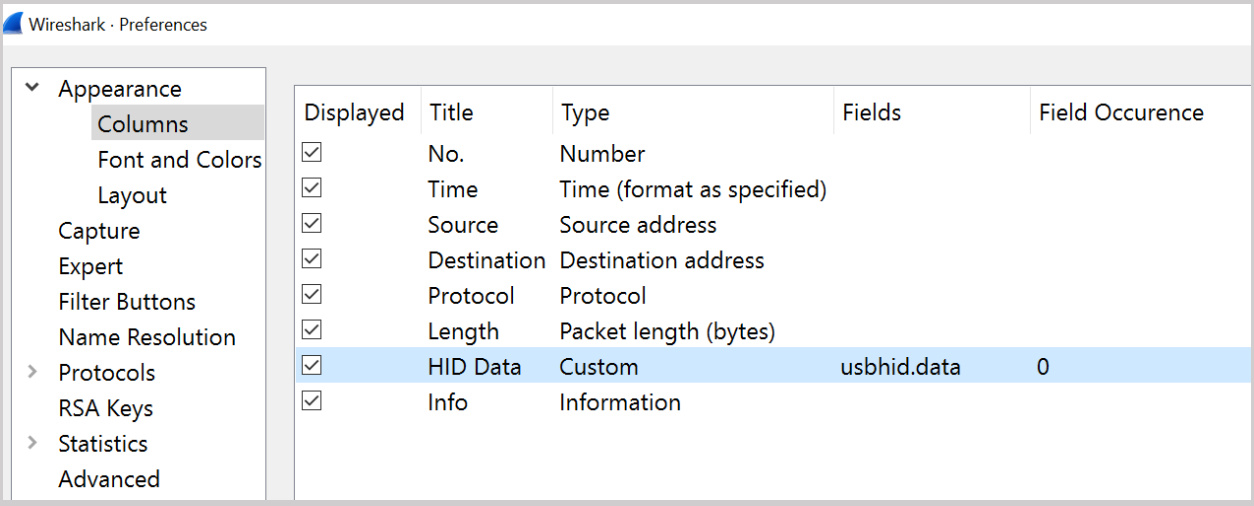
No.	Time	Source	Destination	Protocol	Length	Info
31	1.633787	1.13.1	host	USB	33	URB_INTERRUPT in
33	1.641754	1.13.1	host	USB	33	URB_INTERRUPT in
35	1.649785	1.13.1	host	USB	33	URB_INTERRUPT in
37	1.657777	1.13.1	host	USB	33	URB_INTERRUPT in

I then exported the list to a csv so I could quickly look through the data and parse it out as needed.



	A	B	C	D	E	F	G
1	No.	Time	Source	Destinatio	Protocol	Length	Info
2	31	1.633787	1.13.1	host	USB	33	URB_INTERRUPT in
3	33	1.641754	1.13.1	host	USB	33	URB_INTERRUPT in
4	35	1.649785	1.13.1	host	USB	33	URB_INTERRUPT in
5	37	1.657777	1.13.1	host	USB	33	URB_INTERRUPT in
6	39	1.665767	1.13.1	host	USB	33	URB_INTERRUPT in
7	41	1.673759	1.13.1	host	USB	33	URB_INTERRUPT in
8	43	1.681751	1.13.1	host	USB	33	URB_INTERRUPT in

Unfortunately, this does not have the data I need so I had to add a HID Data Column to wireshark



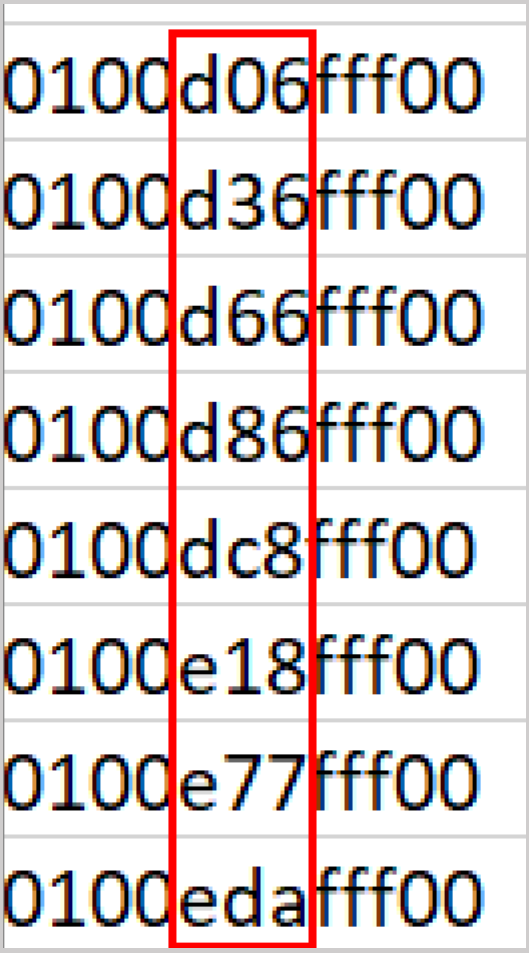
usb.src == "1.13.1"

o.	Time	Source	Destination	Protocol	Length	HID Data	Info
31	1.633787	1.13.1	host	USB	33	010001000000	URB_INTERRUPT in
33	1.641754	1.13.1	host	USB	33	010004000000	URB_INTERRUPT in
35	1.649785	1.13.1	host	USB	33	010006100000	URB_INTERRUPT in
37	1.657777	1.13.1	host	USB	33	010006100000	URB_INTERRUPT in
39	1.665767	1.13.1	host	USB	33	010003100000	URB_INTERRUPT in
41	1.673759	1.13.1	host	USB	33	010001000000	URB_INTERRUPT in

Now I should have what I need, so I exported it again

	A	B	C	D	E	F	G	H
1	No.	Time	Source	Destinatio	Protocol	Length	HID Data	Info
2	31	1.633787	1.13.1	host	USB	33	10001000000	URB_INTERRUPT in
3	33	1.641754	1.13.1	host	USB	33	10004000000	URB_INTERRUPT in
4	35	1.649785	1.13.1	host	USB	33	10006100000	URB_INTERRUPT in
5	37	1.657777	1.13.1	host	USB	33	10006100000	URB_INTERRUPT in
6	39	1.665767	1.13.1	host	USB	33	10003100000	URB_INTERRUPT in
7	41	1.673759	1.13.1	host	USB	33	10001000000	URB_INTERRUPT in
8	43	1.681751	1.13.1	host	USB	33	10001000000	URB_INTERRUPT in
9	45	2.00187	1.13.1	host	USB	33	10100000000	URB_INTERRUPT in
10	47	2.081778	1.13.1	host	USB	33	10000000000	URB_INTERRUPT in
11	217	54.24974	1.13.1	host	USB	33	0100ff0f0000	URB_INTERRUPT in
12	219	54.25771	1.13.1	host	USB	33	0100f2cfff00	URB_INTERRUPT in
13	221	54.26572	1.13.1	host	USB	33	0100dd7fff00	URB_INTERRUPT in
14	223	54.2737	1.13.1	host	USB	33	0100d14fff00	URB_INTERRUPT in
15	225	54.28172	1.13.1	host	USB	33	0100cc4fff00	URB_INTERRUPT in

Now I need to see what items are changing



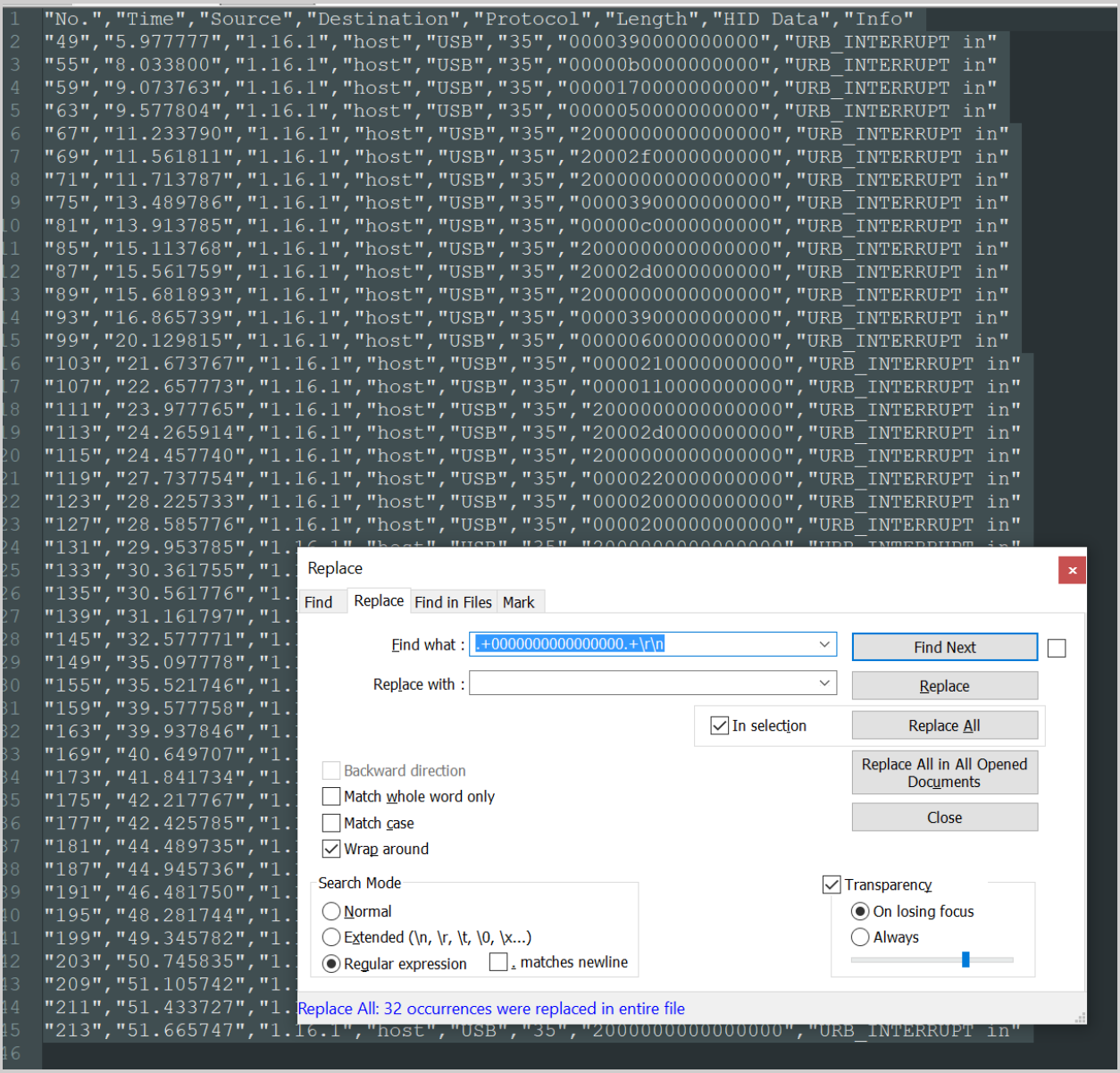
The 5th, 6th, and 7th values seem to be changing, so one of them will hopefully be the data we want.

The 5th and 6th values don't seem to make a lot of sense so I tried the 6th and 7th and they didn't seem much better.

I did start at the top though, so it's possible this is a normal startup procedure.

I scrolled down a little bit towards the middle of the list and tried again.

Eventually I confirmed that I needed values 5 and 6. So I set out extracting those values with some regex. I also removed all 0 value entries.


"No.","Time","Source","Destination","Protocol","Length","HID Data","Info"
"49","5.977777","1.16.1","host","USB","35","0000390000000000","URB_INTERRUPT in"
"55","8.033800","1.16.1","host","USB","35","00000b0000000000","URB_INTERRUPT in"
"59","9.073763","1.16.1","host","USB","35","0000170000000000","URB_INTERRUPT in"
"63","9.577804","1.16.1","host","USB","35","0000050000000000","URB_INTERRUPT in"
"67","11.233790","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"69","11.561811","1.16.1","host","USB","35","20002f0000000000","URB_INTERRUPT in"
"71","11.713787","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"75","13.489786","1.16.1","host","USB","35","0000390000000000","URB_INTERRUPT in"
"81","13.913785","1.16.1","host","USB","35","00000c0000000000","URB_INTERRUPT in"
"85","15.113768","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"87","15.561759","1.16.1","host","USB","35","20002d0000000000","URB_INTERRUPT in"
"89","15.681893","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"93","16.865739","1.16.1","host","USB","35","0000390000000000","URB_INTERRUPT in"
"99","20.129815","1.16.1","host","USB","35","0000060000000000","URB_INTERRUPT in"
"103","21.673767","1.16.1","host","USB","35","0000210000000000","URB_INTERRUPT in"
"107","22.657773","1.16.1","host","USB","35","0000110000000000","URB_INTERRUPT in"
"111","23.977765","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"113","24.265914","1.16.1","host","USB","35","20002d0000000000","URB_INTERRUPT in"
"115","24.457740","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
"119","27.737754","1.16.1","host","USB","35","0000220000000000","URB_INTERRUPT in"
"123","28.225733","1.16.1","host","USB","35","0000200000000000","URB_INTERRUPT in"
"127","28.585776","1.16.1","host","USB","35","0000200000000000","URB_INTERRUPT in"
"131","29.953785","1.
"133","30.361755","1.
"135","30.561776","1.
"139","31.161797","1.
"145","32.577771","1.
"149","35.097778","1.
"155","35.521746","1.
"159","39.577758","1.
"163","39.937846","1.
"169","40.649707","1.
"173","41.841734","1.
"175","42.217767","1.
"177","42.425785","1.
"181","44.489735","1.
"187","44.945736","1.
"191","46.481750","1.
"195","48.281744","1.
"199","49.345782","1.
"203","50.745835","1.
"209","51.105742","1.
"211","51.433727","1.
"213","51.665747","1.16.1","host","USB","35","2000000000000000","URB_INTERRUPT in"
Replace
Find
Replace
Find in Files
Mark
Find what :
.+0000000000000000.+\r\n
Find Next
Replace with :
Replace
In selection
Replace All
Backward direction
Match whole word only
Match case
Wrap around
Replace All in All Opened Documents
Close
Search Mode
Normal
Extended (\n, \r, \t, \0, \x...)
Regular expression
. matches newline
Transparency
On losing focus
Always
Replace All: 32 occurrences were replaced in entire file

```
HID Data
0000390000000000
00000b0000000000
0000170000000000
0000050000000000
2000000000000000
20002f0000000000
2000000000000000
0000390000000000
00000c0000000000
2000000000000000
20002d0000000000
2000000000000000
0000390000000000
0000060000000000
0000210000000000
0000110000000000
2000000000000000
20002d0000000000
2000000000000000
0000220000000000
0000200000000000
0000200000000000
2000000000000000
20002d0000000000
2000000000000000
0000390000000000
00001c0000000000
0000390000000000
```

```
HID Data
39
0b
17
05
2f
39
0c
2d
39
06
21
11
2d
22
20
20
2d
39
1c
39
12
18
39
15
2d
39
0e
20
1c
1f
39
30
```

Once I got my values carved out, I threw them into excel and made a vlookup against the table I had, this way I don't need to look everything up by hand

The document I was using does not allow me to copy text so I grabbed this table

https://www.toomanyatoms.com/computer/usb_keyboard_codes.html

B1 =VLOOKUP(A1,Sheet3!A:B,2,FALSE)

	A	B	C	D	E
1	39	Keyboard Caps Lock11			
2	0b	Keyboard h and H			
3	17	Keyboard t and T			
4	5	Keyboard b and B			

	A	B
1	Usage ID (Hex)	Usage Name
2	0	Reserved (no event indicated)9
3	1	Keyboard ErrorRollOver9
4	2	Keyboard POSTFail9
5	3	Keyboard ErrorUndefined9
6	4	Keyboard a and A4
7	5	Keyboard b and B
8	6	Keyboard c and C4
9	7	Keyboard d and D
10	8	Keyboard e and E

All said and done, I got this

	A	B
1	39	Keyboard Caps Lock11
2	0b	Keyboard h and H
3	17	Keyboard t and T
4	5	Keyboard b and B
5	2f	Keyboard [and {4
6	39	Keyboard Caps Lock11
7	0c	Keyboard i and I
8	2d	Keyboard - and (underscore)4
9	39	Keyboard Caps Lock11
10	6	Keyboard c and C4
11	21	Keyboard 4 and $4
12	11	Keyboard n and N
13	2d	Keyboard - and (underscore)4
14	22	Keyboard 5 and %4
15	20	Keyboard 3 and #4
16	20	Keyboard 3 and #4
17	2d	Keyboard - and (underscore)4
18	39	Keyboard Caps Lock11
19	1c	Keyboard y and Y4
20	39	Keyboard Caps Lock11
21	12	Keyboard o and O4
22	18	Keyboard u and U
23	39	Keyboard Caps Lock11
24	15	Keyboard r and R
25	2d	Keyboard - and (underscore)4
26	39	Keyboard Caps Lock11
27	0e	Keyboard k and K
28	20	Keyboard 3 and #4
29	1c	Keyboard y and Y4
30	1f	Keyboard 2 and @4
31	39	Keyboard Caps Lock11
32	30	Keyboard] and }4
33		

Flag

HTB{i_C4N_533_yOUr_K3Y2}