Write-Up

[Dreamhack] My Nervous PPT

gi11rl 2025. 5. 8. 22:30

은근 공부한 게 많은 문제여서 라업을 적어본다.

문제

[Category]

Forensics, Misc

[Description]

> Umm.. 난 PPT 발표를 하기 전, 긴장된 마음을 풀고자
> 프리젠터를 막 누르곤 하지.. :(
> FLag 형식은 KHK{}입니다.
> *수정모든 Flag는 대문자입니다.

[File]
My_PPT.pcapng

풀이

pcapng 파일이니까 일단 Wireshark로 열어봄.

PPT 랑 Wireshark랑 무슨 상관이지?
-> 이건 네트워크 패킷이 아니라 USB 패킷임.

USB? 정보저장매체?
-> 정확히는 "Universal Serial Bus" 규격의 인터페이스를 의미.
흔히 아는 그 정보저장매체 USB도 사실 Universal Serial Bus 규격에 따라 동작하는 저장매체를 줄여 부르는 관용어일 뿐.
여기서 USB 패킷이라고 함은
- USB 버스 위에 연결된 모든 장치 (키보드, 마우스, 프린터, 저장장치 등) 에 의해
USB 버스 상의 전송 로그다.

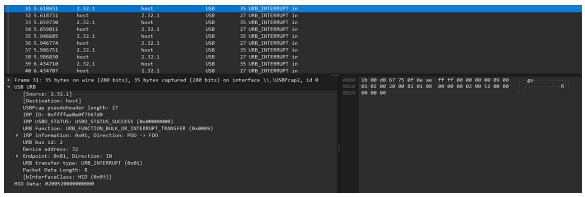
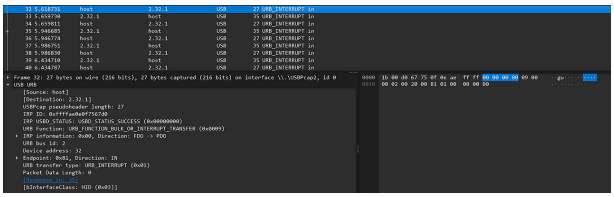
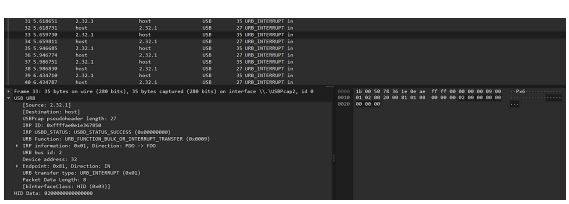
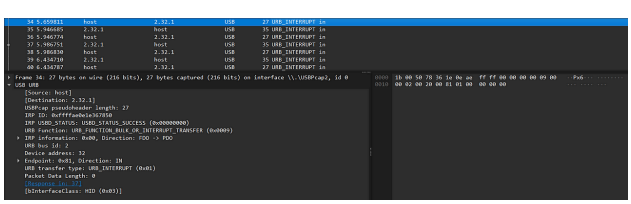
쉽게 말해서, 마우스 클릭, 키보드 입력 등의 패킷임.

No.	Time	Source	Destination	Protocol	Length	Info
31	5.618651	2.32.1	host	USB	35	URB_INTERRUPT in
32	5.618731	host	2.32.1	USB	27	URB_INTERRUPT in
33	5.659730	2.32.1	host	USB	35	URB_INTERRUPT in
34	5.659811	host	2.32.1	USB	27	URB_INTERRUPT in
35	5.946685	2.32.1	host	USB	35	URB_INTERRUPT in
36	5.946774	host	2.32.1	USB	27	URB_INTERRUPT in
37	5.986751	2.32.1	host	USB	35	URB_INTERRUPT in
38	5.986830	host	2.32.1	USB	27	URB_INTERRUPT in
39	6.434710	2.32.1	host	USB	35	URB_INTERRUPT in
40	6.434787	host	2.32.1	USB	27	URB_INTERRUPT in
41	6.482641	2.32.1	host	USB	35	URB_INTERRUPT in
42	6.482673	host	2.32.1	USB	27	URB_INTERRUPT in
43	9.618735	2.32.1	host	USB	35	URB_INTERRUPT in
44	9.618811	host	2.32.1	USB	27	URB_INTERRUPT in
45	9.658910	2.32.1	host	USB	35	URB_INTERRUPT in
46	9.659002	host	2.32.1	USB	27	URB_INTERRUPT in
47	10.066767	2.32.1	host	USB	35	URB_INTERRUPT in
48	10.066846	host	2.32.1	USB	27	URB_INTERRUPT in
49	10.106757	2.32.1	host	USB	35	URB_INTERRUPT in
50	10.106850	host	2.32.1	USB	27	URB_INTERRUPT in
51	10.514827	2.32.1	host	USB	35	URB_INTERRUPT in
52	10.514914	host	2.32.1	USB	27	URB_INTERRUPT in
53	10.562826	2.32.1	host	USB	35	URB_INTERRUPT in
54	10.562913	host	2.32.1	USB	27	URB_INTERRUPT in
55	11.050678	2.32.1	host	USB	35	URB_INTERRUPT in
56	11.050768	host	2.32.1	USB	27	URB_INTERRUPT in
57	11.090828	2.32.1	host	USB	35	URB_INTERRUPT in
58	11.090913	host	2.32.1	USB	27	URB_INTERRUPT in
59	18.138669	2.32.1	host	USB	35	URB_INTERRUPT in
60	18.138736	host	2.32.1	USB	27	URB_INTERRUPT in
61	18.178657	2.32.1	host	USB	35	URB_INTERRUPT in
62	18.178700	host	2.32.1	USB	27	URB_INTERRUPT in
63	18.474718	2.32.1	host	USB	35	URB_INTERRUPT in
64	18.474810	host	2.32.1	USB	27	URB_INTERRUPT in
65	18.522822	2.32.1	host	USB	35	URB_INTERRUPT in
66	18.522908	host	2.32.1	USB	27	URB_INTERRUPT in
67	19.114821	2.32.1	host	USB	35	URB_INTERRUPT in

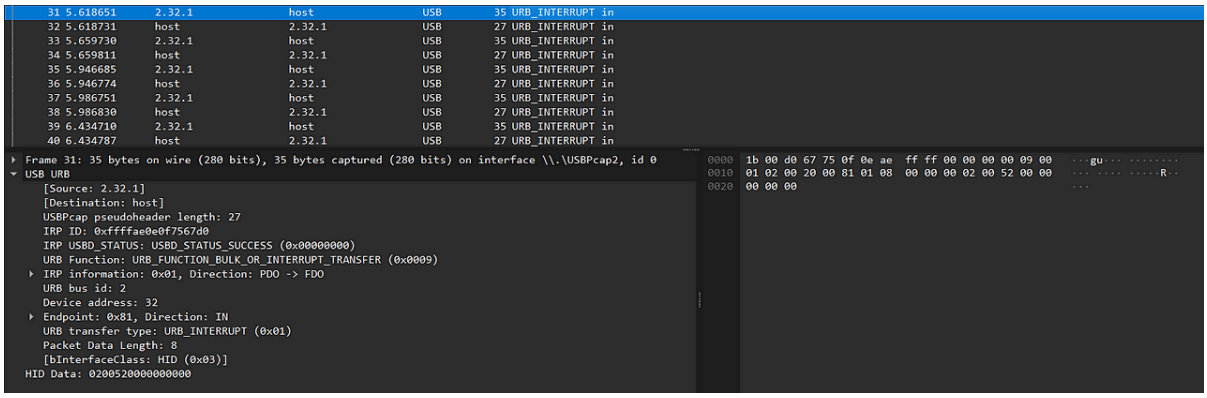
> URB_INTERRUPT 에서 URB가 뭐지?
> -> USB Request Block

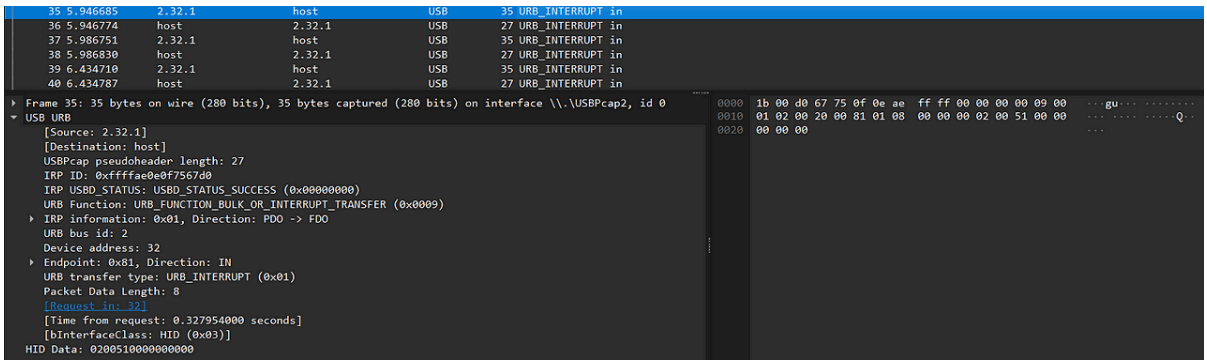
문제 설명에서 프리젠터를 막 눌렀다고 했음.
이로 인해 연달아 수많은 인터럽트가 발생한 것으로 보임.

FLAG를 얻기 위해 규칙을 찾아봄.

우선, 패킷 4개가 한 묶음인 것으로 보임.
- 의미있는 HID DATA는 4*n번째에 발생하고 있음





HEX 상 R 과 Q 로 나타나는 저 부분이 HID DATA 임.

> HID DATA란 실제로 디바이스가 보낸 바이트 스트림
> EX) 키보드로 누른 키 값

> HID DATA 해석하는 방법
> https://gist.github.com/MightyPork/6da26e382a7ad91b5496ee55fdc73db2
>
> R로 나타난 HID Data = 0x0200520000000000
>
> 첫번째 바이트 : Modifier mask
> 0x02 이므로, Left Shift를 누른 것
>
> 두번째 바이트는 reserved (0x00)
>
> 다음 바이트는 0x52 이므로, 위쪽 화살표를 누른 것
> 즉, Shift + Up Arrow 키가 동시에 눌린 상태임.

같은 방식으로 Q로 나타난 HID DATA는 Shift + Down Arrow 키가 동시에 눌린 상태.

그런데 이건 키보드가 아니라 프리젠터임
프리젠터 내부에서 어떻게 매핑하는지는 모르겠지만 .. 아마 이전 슬라이드 키 / 다음 슬라이드 키 누른 거 아닐까?

```
/**
 * Modifier masks - used for the first byte in the HID report.
 * NOTE: The second byte in the report is reserved, 0x00
 */
#define KEY_MOD_LCTRL  0x01
#define KEY_MOD_LSHIFT 0x02
#define KEY_MOD_LALT   0x04
#define KEY_MOD_LMETA  0x08
#define KEY_MOD_RCTRL  0x10
#define KEY_MOD_RSHIFT 0x20
#define KEY_MOD_RALT   0x40
#define KEY_MOD_RMETA  0x80
```

```
110   #define KEY_SYSRQ 0x46 // Keyboard Print Screen
111   #define KEY_SCROLLLOCK 0x47 // Keyboard Scroll Lock
112   #define KEY_PAUSE 0x48 // Keyboard Pause
113   #define KEY_INSERT 0x49 // Keyboard Insert
114   #define KEY_HOME 0x4a // Keyboard Home
115   #define KEY_PAGEUP 0x4b // Keyboard Page Up
116   #define KEY_DELETE 0x4c // Keyboard Delete Forward
117   #define KEY_END 0x4d // Keyboard End
118   #define KEY_PAGEDOWN 0x4e // Keyboard Page Down
119   #define KEY_RIGHT 0x4f // Keyboard Right Arrow
120   #define KEY_LEFT 0x50 // Keyboard Left Arrow
121   #define KEY_DOWN 0x51 // Keyboard Down Arrow
122   #define KEY_UP 0x52 // Keyboard Up Arrow
```

USB HID Keyboard scan codes

USB HID Keyboard scan codes. GitHub Gist: instantly share code, notes, and snippets.

gist.github.com

HID DATA에는 R과 Q만 나타남. 그런데 RQRQRQ.. 이런 식이 아니라 RQQQRRQR 이런 식으로 불규칙한 패턴이 있었음.
그래서 이게 FLAG를 의미할 것이라 생각하게 됨.
그런데 어떻게 끊어야할까 생각하다가

```
▸ Frame 35: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface \\.\USBPcap2, id 0
▾ USB URB
    [Source: 2.32.1]
    [Destination: host]
    USBPcap pseudoheader length: 27
    IRP ID: 0xffffae0e0f7567d0
    IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
    URB Function: URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER (0x0009)
  ▸ IRP information: 0x01, Direction: PDO -> FDO
    URB bus id: 2
    Device address: 32
  ▸ Endpoint: 0x81, Direction: IN
    URB transfer type: URB_INTERRUPT (0x01)
    Packet Data Length: 8
    [Request in: 32]
    [Time from request: 0.327954000 seconds]
    [bInterfaceClass: HID (0x03)]
  HID Data: 0200510000000000
```

```
▸ Frame 61: 35 bytes on wire (280 bits), 35 bytes captured (280 bits) on interface \\.\USBPcap2, id 0
▾ USB URB
    [Source: 2.32.1]
    [Destination: host]
    USBPcap pseudoheader length: 27
    IRP ID: 0xffffae0e1e367850
    IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
    URB Function: URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER (0x0009)
  ▸ IRP information: 0x01, Direction: PDO -> FDO
    URB bus id: 2
    Device address: 32
  ▸ Endpoint: 0x81, Direction: IN
    URB transfer type: URB_INTERRUPT (0x01)
    Packet Data Length: 8
    [Request in: 58]
    [Time from request: 7.087744000 seconds]
    [bInterfaceClass: HID (0x03)]
  HID Data: 0200000000000000
```

패킷마다 다른 건 Time from request 뿐이었음.
0. xxxx 초인 패킷이 대다수였고, 간혹 1초 이상인 패킷이 나타남.

이걸 기준으로 0.xxxx 초인 패킷들은 하나의 단어나 글자를 이루고, 1초 이상인 패킷을 경계로 단어나 글자를 끊어주면 되지 않을까 생각함.

```
RQR
QQQQ
RQR
RQRRQ
QRQQ
RRR
RRR
RQR
QQRRQR
QR
R
QQRRQR
RR
RQRR
QQRRQR
QRRQ
QRRQ
R
RQRRQR
```

끊어보니 이랬음.

여기서부터 헤맸는데 ..
생각해보니 단순
R과 Q로만 나타내졌으므로 1) 이진수 2) 모스부호 중 하나로 풀이가 가능할 것
(이진수만 생각해서 엄청나게 헤맸다.)

A •−	J •−−−	S •••	1 •−−−−
B −•••	K −•−	T −	2 ••−−−
C −•−•	L •−••	U ••−	3 •••−−
D −••	M −−	V •••−	4 ••••−
E •	N −•	W •−−	5 •••••
F ••−•	O −−−	X −••−	6 −••••
G −−•	P •−−•	Y −•−−	7 −−•••
H ••••	Q −−•−	Z −−••	8 −−−••
I ••	R •−•	0 −−−−−	9 −−−−•

FLAG가 KHK 로 시작한다 했는데 첫 글자가 RQR

K = RQR 이려면

-> R = - /// Q = .

이 방식으로 모두 모스부호로 대응시켜보면 FLAG가 나온다.

'Write-Up' 카테고리의 다른 글

[2025 Hacktheon 예선] Write-Up (5) 2025.07.29
[Dreamhack] Don't Do(S) That! (4) 2025.05.09
[UTCTF 2025] Finally, an un-strings-able problem (2) 2025.03.17
[DownUnder CTF 2024] intercepted transmissions (6) 2024.07.22
[DownUnder CTF 2024] SAM I AM (7) 2024.07.10

'Write-Up'의 다른글

이전글 [UTCTF 2025] Finally, an un-strings-able problem

현재글 : [Dreamhack] My Nervous PPT

다음글 [Dreamhack] Don't Do(S) That!