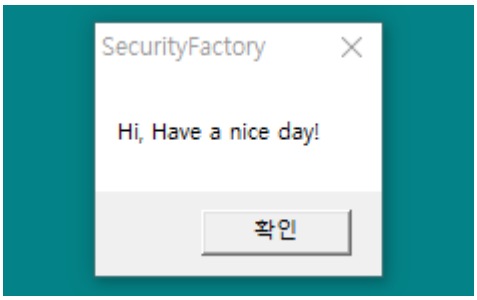
[Sample 01.exe 분석]

1. 실행 내용

실행 시 Beep음이 나오고, 이후 MessageBox가 출력됩니다.



이때 실행이 잘 된다는 사실을 알았으므로, 32비트 동작 실행 파일이며, AntiVM 기법이 적용되어 있지 않다는걸 쉽게 알 수 있었습니다.

2. 디버깅 과정

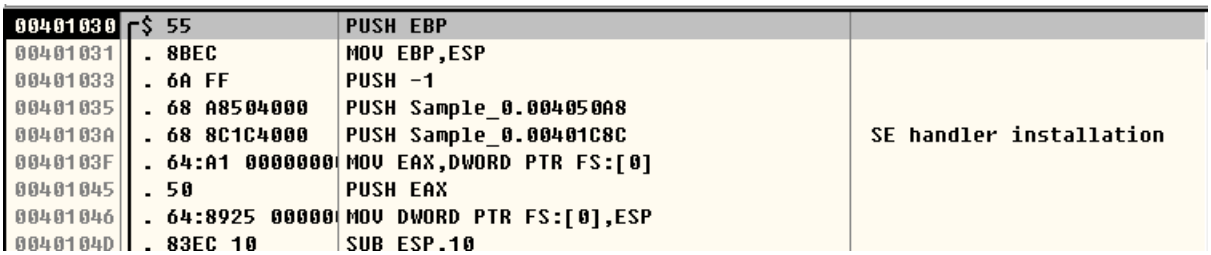
OllyDBG로 디버깅을 진행합니다.

올릴 시 00401030주소부터 시작을 하며, 해당 구문이

PUSH EBP

MOV EBP, ESP

인 것으로 보아 Stack의 시작점으로, StartUp코드의 위치인 듯 합니다.



그럼 이제 해당 프로그램에서 사용된 API를 알아봅시다. 우선 .Beep음과 메시지 박스가 출력되므로, 이에 관련된 API를 사용할 것입니다.

역시나, Win32의 API인 Beep()와 MessageBoxA()를 사용하는 모습을 볼 수 있습니다.

해당 API의 분석을 위해 MSDN에 이름을 검색해 보면,

Beep function

Generates simple tones on the speaker. The function is synchronous; it performs an alertable wait and does not return control to its caller until the sound finishes.

Syntax

```
BOOL WINAPI Beep(
       _In_   DWORD  dwFreq,
       _In_   DWORD  dwDuration
);
```

Parameters

dwFreq [in]

The frequency of the sound, in hertz. This parameter must be in the range 37 through 32,767 (0x25 through 0x7FFF).

dwDuration [in]

The duration of the sound, in milliseconds.

Return value

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call GetLastError.

MessageBoxA function

Displays a modal dialog box that contains a system icon, a set of buttons, and a brief application-specific message, such as status or error information. The message box returns an integer value that indicates which button the user clicked.

Syntax

```
int MessageBoxA(
       _In_    HWND    hWnd,
       _In_    LPCSTR lpText,
       _In_    LPCSTR lpCaption,
       _In_    UINT   uType
);
```

Parameters

hWnd [in]

A handle to the owner window of the message box to be created. If this parameter is NULL, the message box has no owner window.

lpText [in]

The message to be displayed. If the string consists of more than one line, you can separate the lines using a carriage return and/or linefeed character between each line.

lpCaption [in]

The dialog box title. If this parameter is NULL, the default title is Error.

uType [in]

The contents and behavior of the dialog box. This parameter can be a combination of flags from the following groups of flags.

Return value

If a message box has a Cancel button, the function returns the IDCANCEL value if either the ESC key is pressed or the Cancel button is selected. If the message box has no Cancel button, pressing ESC has no effect.

해당 API의 설명을 읽어볼 수 있습니다.

이제 해당 API의 내용을 알았으니 프로그램이 어떻게 돌아가는지 알 수 있을 것입니다.

```
00401000 ┌$ 68 00030000   PUSH 300                          ┌Duration = 768. ms
00401005 |  68 00020000   PUSH 200                          |
0040100A |. FF15 00504000 CALL DWORD PTR DS:[<&KERNEL32.Beep>] └Beep
```

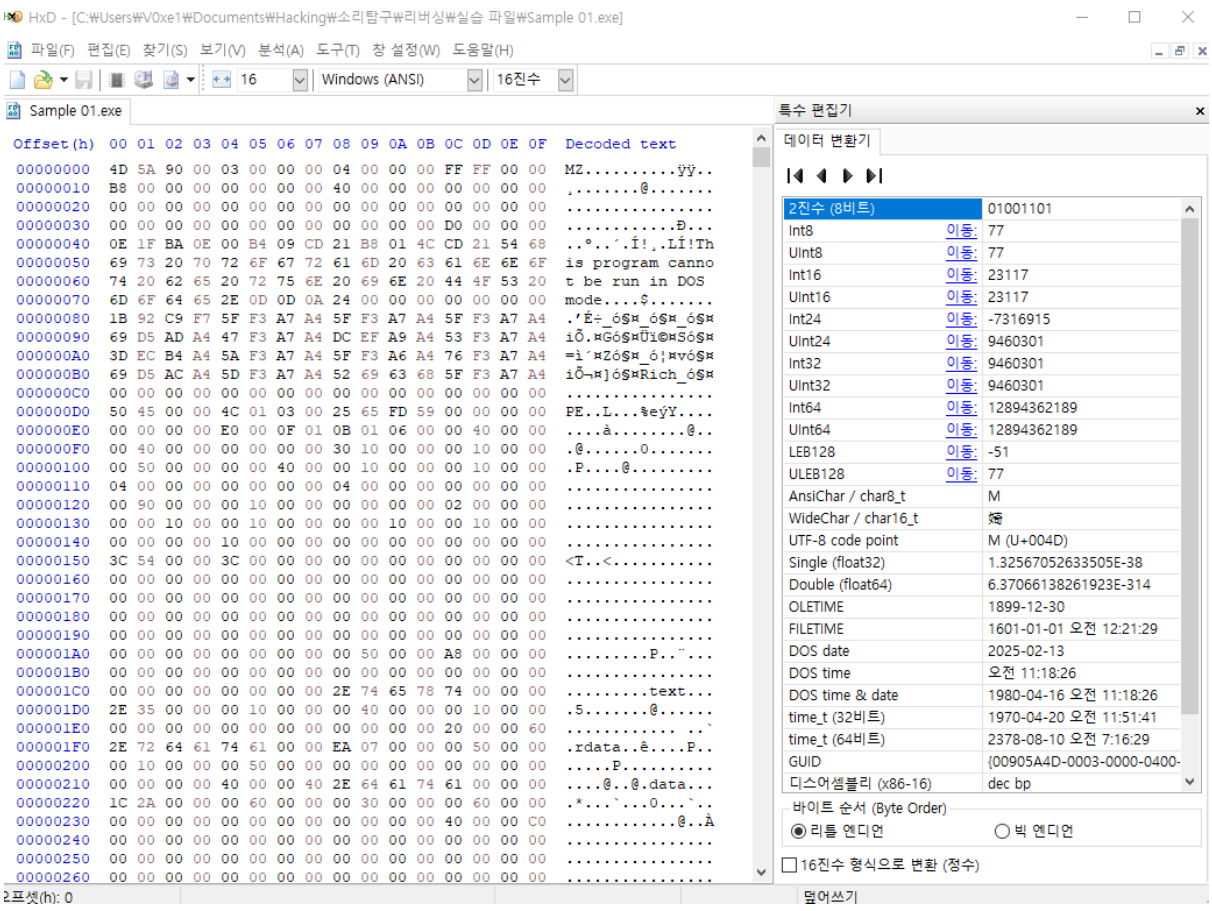
약 768ms만큼의 시간동안, 200값의 소리를 출력하며,

```
00401010 . 6A 00          PUSH 0                           ┌Style = MB_OK|MB_APPLMODAL
00401012 . 68 48604000    PUSH Sample_0.00406048           |Title = "SecurityFactory"
00401017 . 68 30604000    PUSH Sample_0.00406030           |Text = "Hi, Have a nice day!"
0040101C . 6A 00          PUSH 0                           |hOwner = NULL
0040101E . FF15 A0504000  CALL DWORD PTR DS:[<&USER32.MessageBoxA> └MessageBoxA
00401024 . B8 01000000    MOV EAX,1
00401029 L. C3            RETN
```

제목이 SecurityFactory이며, 내용으로 Hi, Have a nice day!를 가진 메시지 박스를 호출해 생성하고, main을 종료하는 모습을 볼 수 있습니다.

3. Hxd로 분석해보기.

Hxd는 파일 데이터를 16진수로 출력해주는 프로그램입니다. 이를 이용해서 열어보면,



PE파일의 포멧에 맞게 기록되어 있음을 알 수 있습니다.

PE File Format 은 다음 ppt 에 정의되어 있으니, 해당 문서에선 일단 넘어가겠습니다.

이 외에도 해당 프로그램에서 분석해야 할 것들은 더 있습니다. 바로 API 를 호출하기 위한 DLL 파일 로드입니다.

Address	Size	Owner	Section	Contains	Type	Access	Initial
003D0000	00003000				Priv	RW	RW
00400000	00001000	Sample_0		PE header	Imag	R	RWE
00401000	00004000	Sample_0	.text	code	Imag	R	RWE
00405000	00001000	Sample_0	.rdata	imports	Imag	R	RWE
00406000	00003000	Sample_0	.data	data	Imag	R	RWE
00410000	00101000				Map	R	R
00520000	00054000				Map	R	R
01160000	00002000				Priv	RW	RW
75C10000	00001000	KERNELBA		PE header	Imag	R	RWE
75C11000	00043000	KERNELBA	.text	code,imports	Imag	R	RWE
75C54000	00002000	KERNELBA	.data	data	Imag	R	RWE
75C56000	00001000	KERNELBA	.rsrc	resources	Imag	R	RWE
75C57000	00003000	KERNELBA	.reloc	relocations	Imag	R	RWE
76AD0000	00001000	USP10		PE header	Imag	R	RWE
76AD1000	0005B000	USP10	.text	code,imports	Imag	R	RWE
76B2C000	00002000	USP10	.data	data	Imag	R	RWE
76B2E000	0002A000	USP10	Shared		Imag	R	RWE
76B58000	00012000	USP10	.rsrc	resources	Imag	R	RWE
76B6A000	00003000	USP10	.reloc	relocations	Imag	R	RWE
76C70000	00001000	kernel32		PE header	Imag	R	RWE
76C71000	000C5000	kernel32	.text	code,imports	Imag	R	RWE
76D36000	00001000	kernel32	.data	data	Imag	R	RWE
76D37000	00001000	kernel32	.rsrc	resources	Imag	R	RWE
76D38000	0000C000	kernel32	.reloc	relocations	Imag	R	RWE

메모리 구간을 보면, 이런 파일들을 DLL 이라고 부릅니다. exe 파일의 실행을 돕는 용도의 파일들입니다.

4. 만들어 보기

이제 Sample 01.exe 에 대한 분석은 마쳤으니, 직접 만들어 보도록 합니다. 사용된 Beep()와 MessageBoxA()만을 잘 사용해서 main 에 구현하면 쉽게 해결할 수 있을 것입니다.

```
1  #include <windows.h>
2
3  int main()
4  {
5      Beep(0x200, 0x300);
6      MessageBoxA(0, "Hi, Have a nice day!", "SecurityFactory", 0);
7
8      return 1;
9  }
```