codeengn-advance-L01 풀이

리버싱 문제풀이 / Wonlf / 2022. 5. 2. 11:38

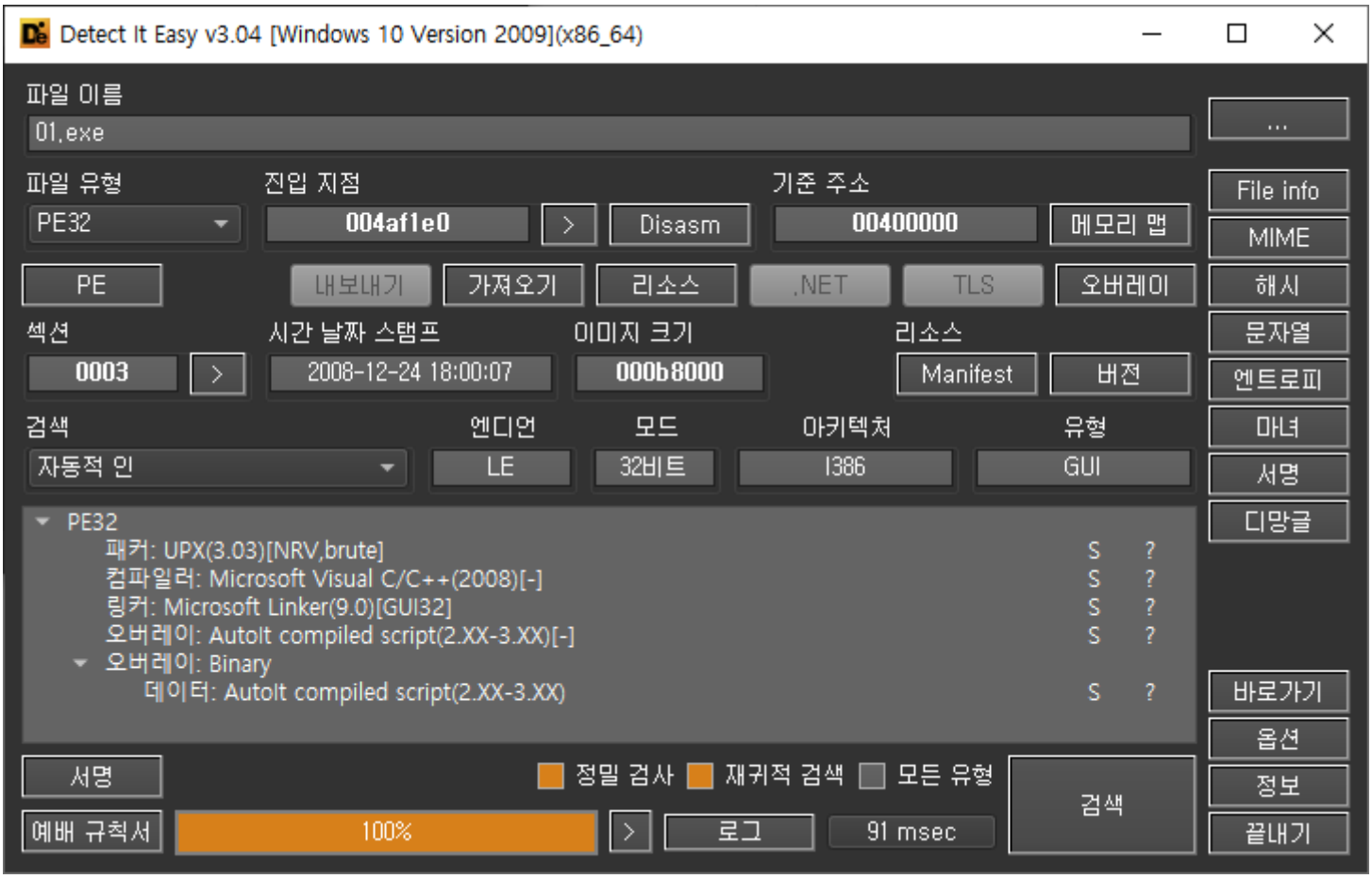
이 문제는 basic에 있었던 19번 문제와 유사하다. 같이 보면 좋을 것 같다.

2022.04.22 - [리버싱 문제 풀이/CodeEngn.com_Basic (Clear)] - codeengn-basic-L19 풀이

codeengn-basic-L19 풀이

wonlf.tistory.com

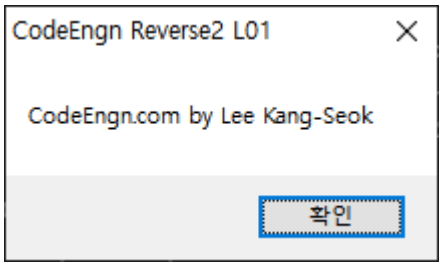
먼저 Die로 열어보면,



UPX패킹이 되어 있다. 언패커를 통해 패킹을 해제 시켜준다.

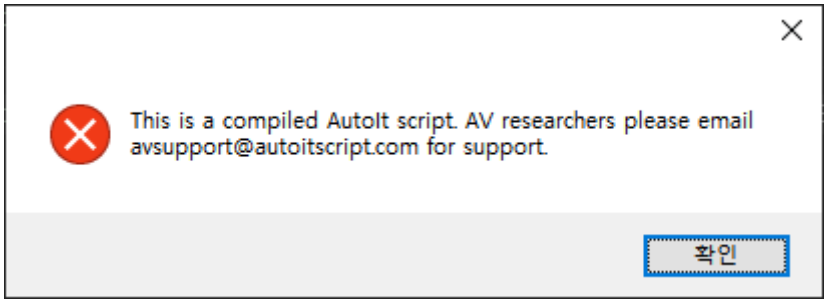
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\a02> C:\Users\a02\Desktop\도구\upx-3.96-win64\upx.exe -d C:\Users\a02\Downloads\01\01.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    613178 <-    290618   47.40%    win32/pe     01.exe

Unpacked 1 file.
```

패킹을 해제하고 프로그램을 실행시켜보면,



메시지 박스가 뜨고 프로그램이 종료 된다. 디버거로 열어본다.

디버거로 열고 파일을 실행시켜보면,

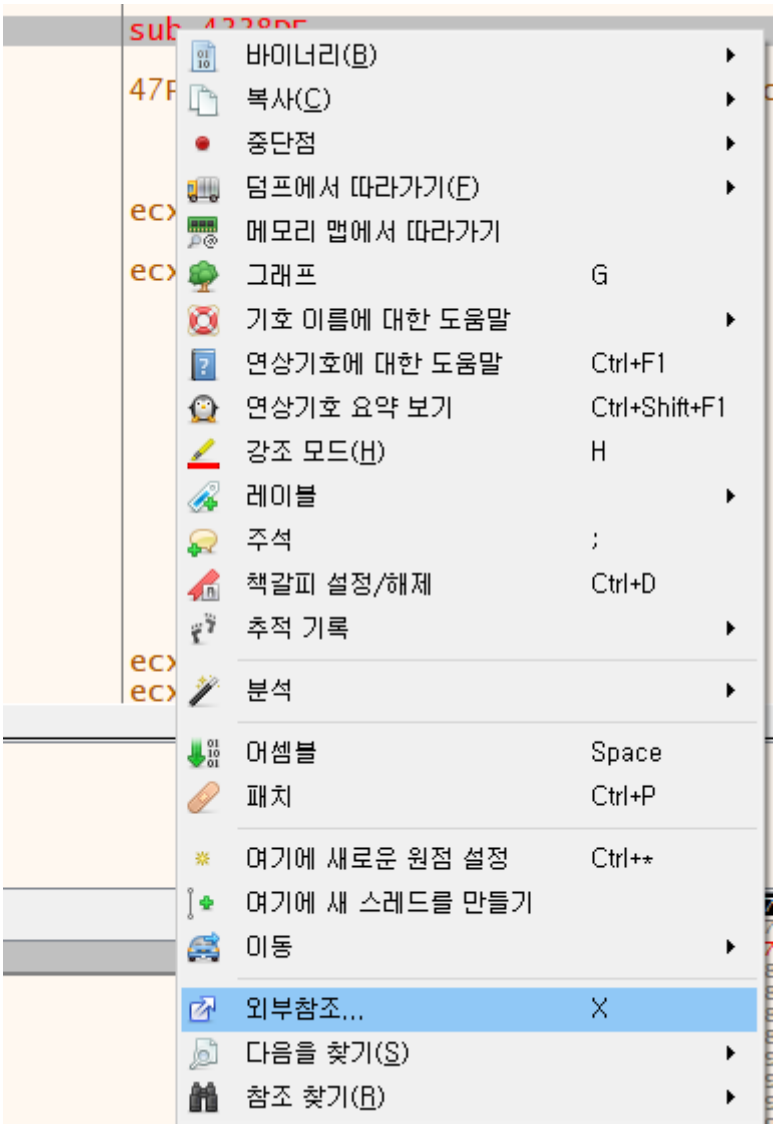


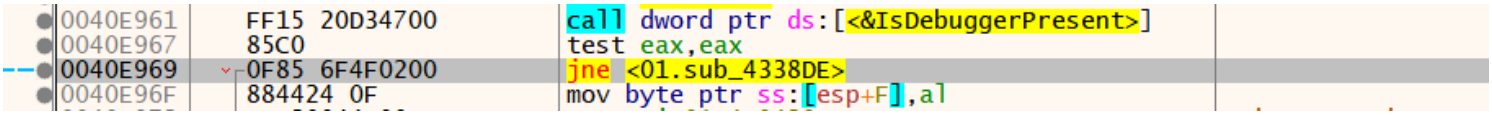
basic 19번 문제와 동일한 알림창이 뜬다. 아마도 안티 디버깅 기법이 들어있는 것 같다.

```
004338DE  r> 6A 10          push 10                          sub_4338DE
004338E0  .  68 9EF64700    push 01.47F69E
004338E5  .  68 A0F64700    push 01.47F6A0                   47F6A0:"This is a compiled AutoIt script. AV researchers
004338EA  .  6A 00          push 0
004338EC  .  FF15 DCD64700  call dword ptr ds:[<&MessageBoxA>]
004338F2  L. E9 49B1FDFF    jmp <01.sub_40EA40>
```

해당 메시지 박스가 뜨는 구문을 찾아 주었다.

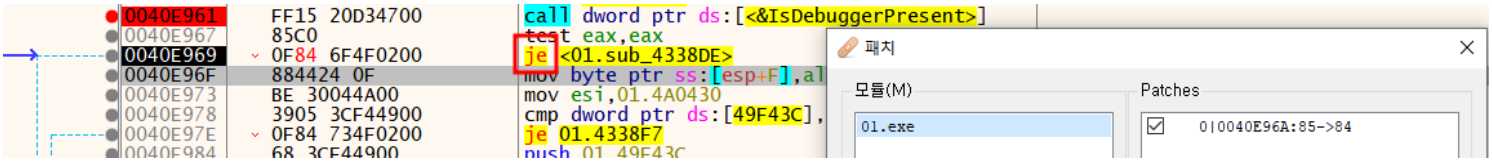
xref로 이 함수를 호출하는 곳을 찾아 주겠다.





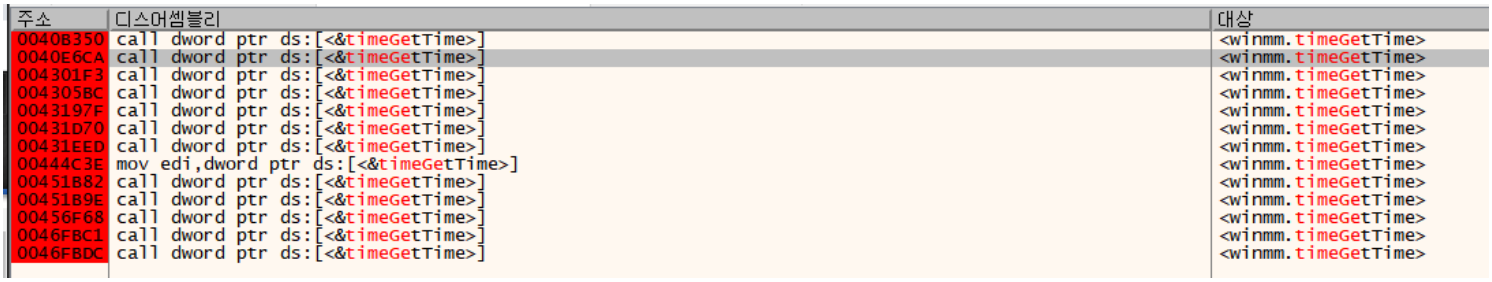
예상한대로 디버깅을 감지하는 함수가 호출되고 디버깅 당하고 있다면 해당 메시지 박스를 띄우는 것이였다.

19번에서 했던 것처럼 jne를 je로 변경해주고 패치해서 새로운 파일로 만든다.



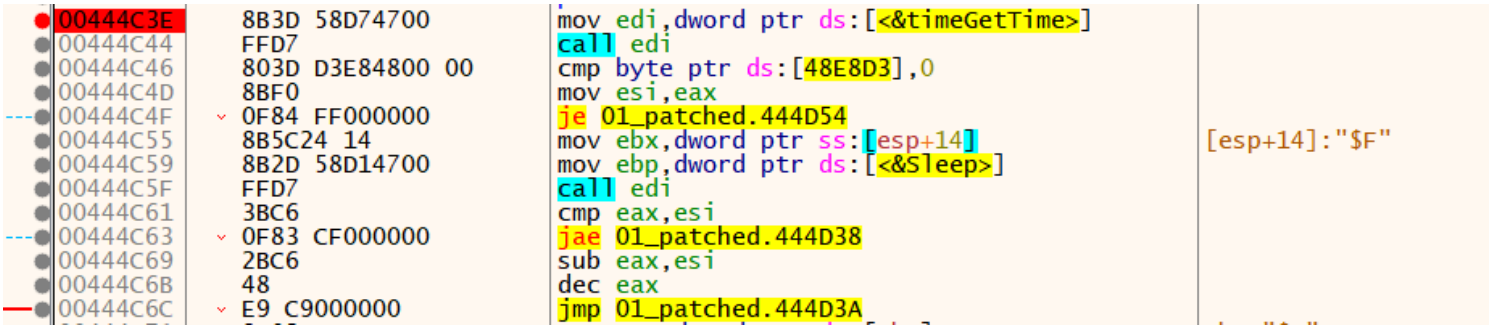
패치하고 실행하니 정상적으로 실행 된다.

이제 문제의 KEY인 시간을 구해보겠다.

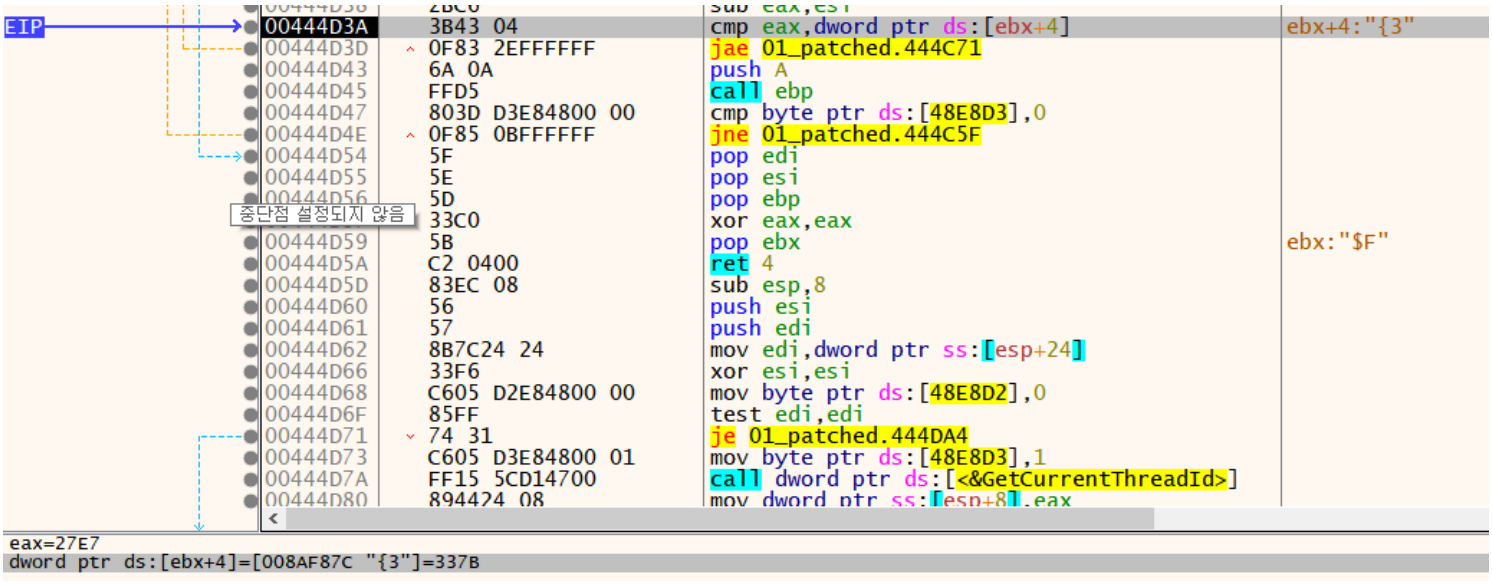


주소	디스어셈블리	대상
0040B350	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
0040E6CA	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
004301F3	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
004305BC	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
0043197F	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00431D70	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00431EED	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00444C3E	mov edi,dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00451B82	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00451B9E	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
00456F68	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
0046FBC1	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>
0046FBDC	call dword ptr ds:[<&timeGetTime>]	<winmm.timeGetTime>

19번처럼 TimeGetTime 함수에 브레이크 포인트를 걸어주고 실행 시키면

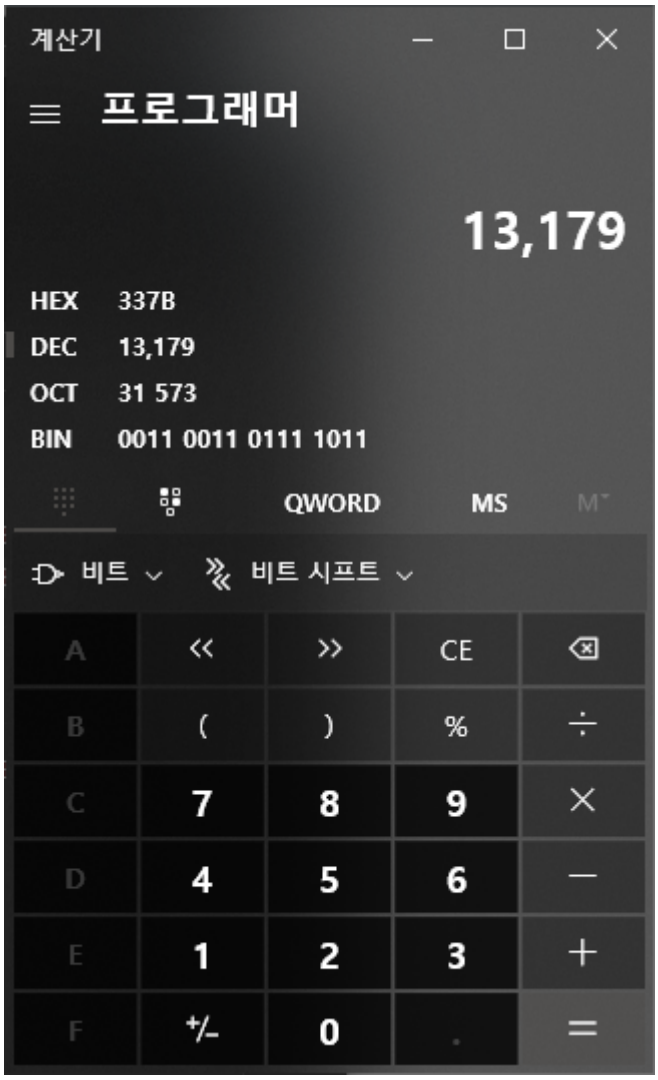


```
00444C3E   8B3D 58D74700      mov edi,dword ptr ds:[<&timeGetTime>]
00444C44   FFD7               call edi
00444C46   803D D3E84800 00   cmp byte ptr ds:[48E8D3],0
00444C4D   8BF0               mov esi,eax
00444C4F   0F84 FF000000      je 01_patched.444D54
00444C55   8B5C24 14          mov ebx,dword ptr ss:[esp+14]      [esp+14]:"$F"
00444C59   8B2D 58D14700      mov ebp,dword ptr ds:[<&Sleep>]
00444C5F   FFD7               call edi
00444C61   3BC6               cmp eax,esi
00444C63   0F83 CF000000      jae 01_patched.444D38
00444C69   2BC6               sub eax,esi
00444C6B   48                 dec eax
00444C6C   E9 C9000000        jmp 01_patched.444D3A
```

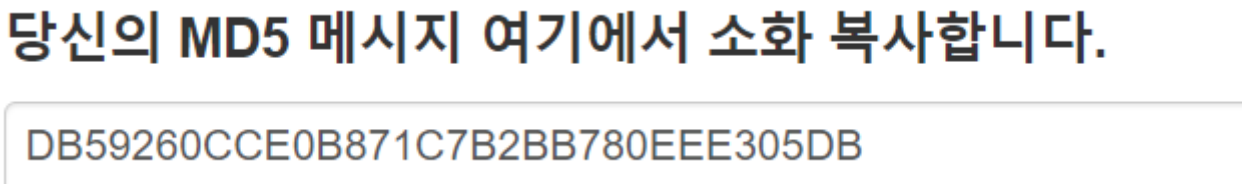


```
EIP
00444D3A   3B43 04            cmp eax,dword ptr ds:[ebx+4]       ebx+4:"{3"
00444D3D   0F83 2EFFFFFF      jae 01_patched.444C71
00444D43   6A 0A              push A
00444D45   FFD5               call ebp
00444D47   803D D3E84800 00   cmp byte ptr ds:[48E8D3],0
00444D4E   0F85 0BFFFFFF      jne 01_patched.444C5F
00444D54   5F                 pop edi
00444D55   5E                 pop esi
00444D56   5D                 pop ebp
중단점 설정되지 않음
00444D59   5B                 pop ebx                            ebx:"$F"
00444D5A   C2 0400            ret 4
00444D5D   83EC 08            sub esp,8
00444D60   56                 push esi
00444D61   57                 push edi
00444D62   8B7C24 24          mov edi,dword ptr ss:[esp+24]
00444D66   33F6               xor esi,esi
00444D68   C605 D2E84800 00   mov byte ptr ds:[48E8D2],0
00444D6F   85FF               test edi,edi
00444D71   74 31              je 01_patched.444DA4
00444D73   C605 D3E84800 01   mov byte ptr ds:[48E8D3],1
00444D7A   FF15 5CD14700      call dword ptr ds:[<&GetCurrentThreadId>]
00444D80   894424 08          mov dword ptr ss:[esp+8],eax

eax=27E7
dword ptr ds:[ebx+4]=[008AF87C "{3"]=337B
```

동일하게 특정 값과 시간을 비교하는 구문이 있다. ebx+4에 들어있는 값은 337B. 10진수로 바꿔주면,



13179가 나오고 문제는 MD5를 원했으니 또 변환해주면,



정답