

# Challenges : Basic 04

Author : CodeEngn

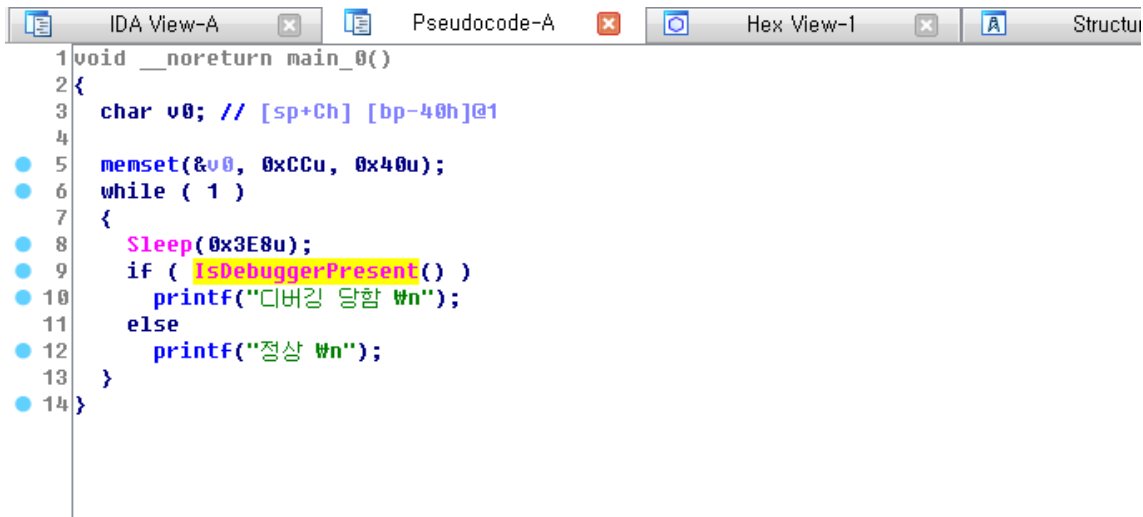
Korean :

이 프로그램은 디버거 프로그램을 탐지하는 기능을 갖고 있다. 디버거를 탐지하는 함수의 이름은 무엇인가

English :

This program can detect debuggers. Find out the name of the debugger detecting function the program uses.

[Download](#)



```
1 void __noreturn main_0()
2 {
3     char v0; // [sp+Ch] [bp-40h]@1
4
5     memset(&v0, 0xCCu, 0x40u);
6     while ( 1 )
7     {
8         Sleep(0x3E8u);
9         if ( IsDebuggerPresent() )
10             printf("디버깅 당함 %n");
11         else
12             printf("정상 %n");
13     }
14 }
```

바로 답이 나온다. IsDebuggerPresent

이 문제의 경우 참 쉽게 풀었다. IDA의 디컴파일기능을 이용해 풀었다(단축키: F5).  
쉽게 풀김에 변태성격이 발동하여 다른것들을 시도해보았다.(simple IsDebuggerResent Bypass)

## 1. EAX(Extended Accumulator Register)

첫번째로, EAX 레지스터에 대해 알아보도록 하겠습니다. EAX 레지스터는 산술(덧셈, 곱셈, 나눗셈 등), 논리 연산을 수행하며 함수의 반환값이 이 레지스터에 저장됩니다. 즉, 덧셈, 곱셈, 나눗셈 등의 명령은 모두 EAX 레지스터를 사용하며, 함수의 반환 값이 EAX 레지스터에 저장되므로 호출 함수의 성공 여부, 실패 여부를 쉽게 파악할 수 있으며, 반환값을 쉽게 얻어올 수 있습니다. 직접 EAX 레지스터의 값이 어떻게 변하는지 확인해보도록 하겠습니다.

즉 IsDebuggerPresent의 리턴값이 EAX에 저장된단 말씀!

cmp쪽에 브레이크 포인트를 걸고 run한후 IDA밑에 커맨드창이 있길래 EAX = 0을

시도해보았다.



결과는 잘먹혔다.(오 zz 역시 나의 직감이란..)

이번에는 어셈블리를 직접적으로 수정해서 EAX에 0을 대입하면 어떨까? 하는 마음에

IDA기능을 쭉쭉 찾아서 어셈블리 수정기능하는 법을 터득하였다.

디버깅하고(디버깅전에 수정하면 디버깅할 때 적용이 되지 않는데 왜 그런진 잘 모르겠다.)

해당 옴코드라인을(IsDebuggerPresent을 call하는부분)을 포커싱하고 Edit -> Path

Program -> Assemble로 가서

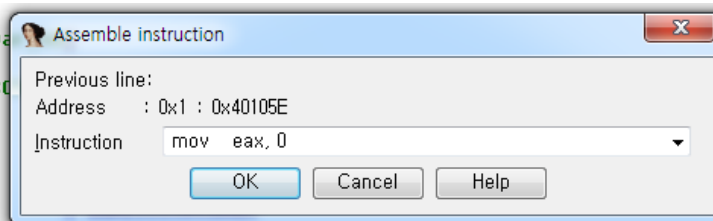
MOV EAX, 0

NOP

으로 수정하였다.

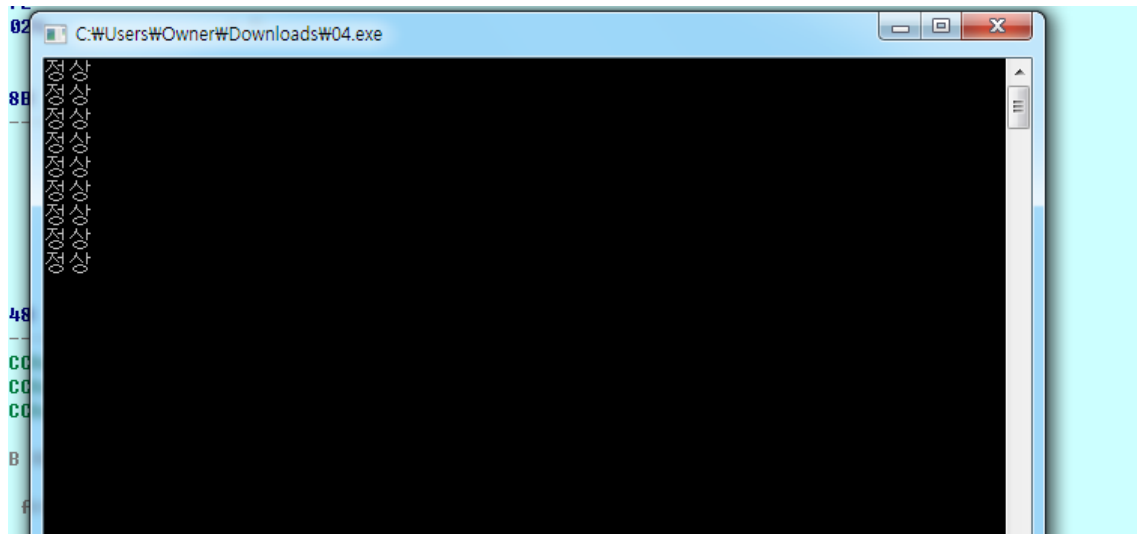
```
push edi
lea edi, [ebp+va
mov ecx, 10h
mov eax, 0CCCCC0
rep stosl
```

```
mov esi, esp
push 3E8h
call ds:Sleep
cmp esi, esp
call __chkesp
mov esi, esp
call ds:IsDebuggerPresent
cmp esi, esp
call __chkesp
```



```
.text:0040105C
.text:0040105E mov     eax, 0
.text:00401063 nop
.text:00401064 cmp     esi, esp
```

[수정된 모습의 사진] (귀여운 어셈블리들 ><)



올ㅋ 역시 디버깅함수를 바이패스한 모습을 볼 수 있었다.