

**CodeEngn**

**Solving problems**

**basic level4**

Nick : C y \_\_ h

Email : [h61cker@gmail.com](mailto:h61cker@gmail.com)

Author : CodeEngn

Korean :

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

English :

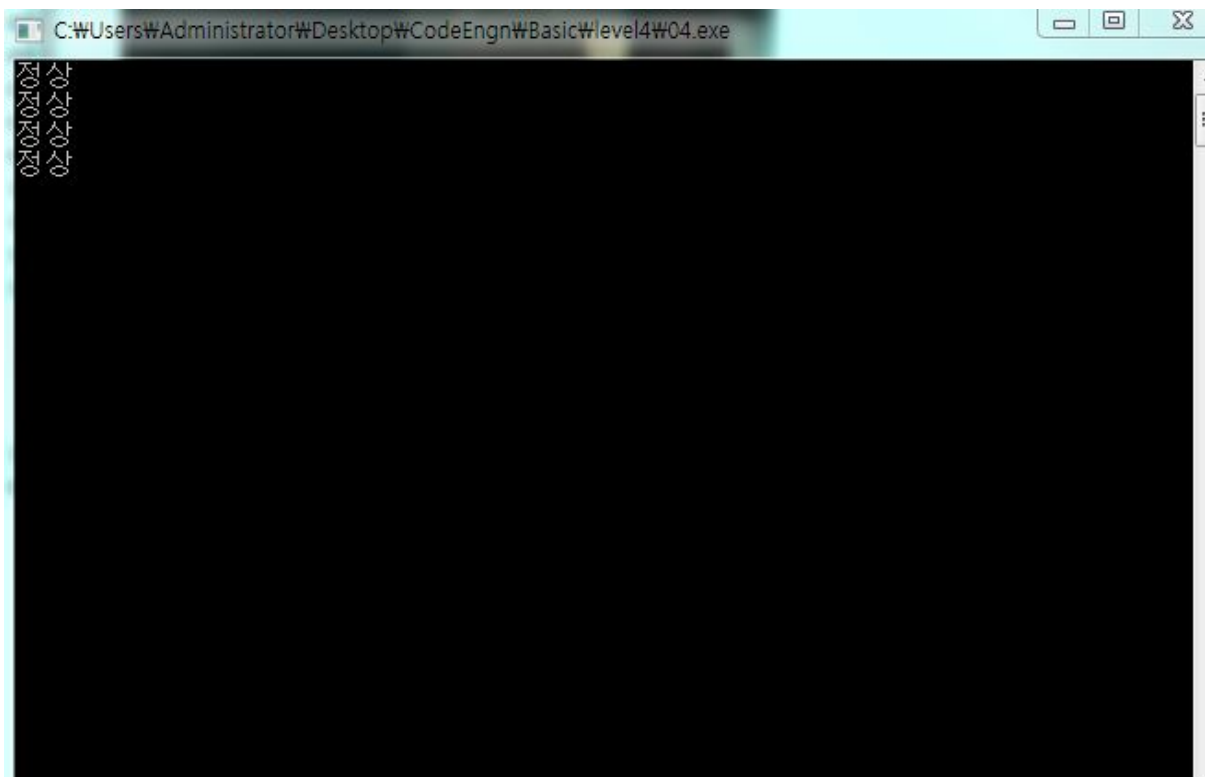
This program can detect debuggers. Find out the name of the debugger detecting function the program uses.

[Download](#)

Linode is a privately owned virtual private server provider based in Galloway, New Jersey

디버거를 탐지하는 함수의 이름은 무엇인가?

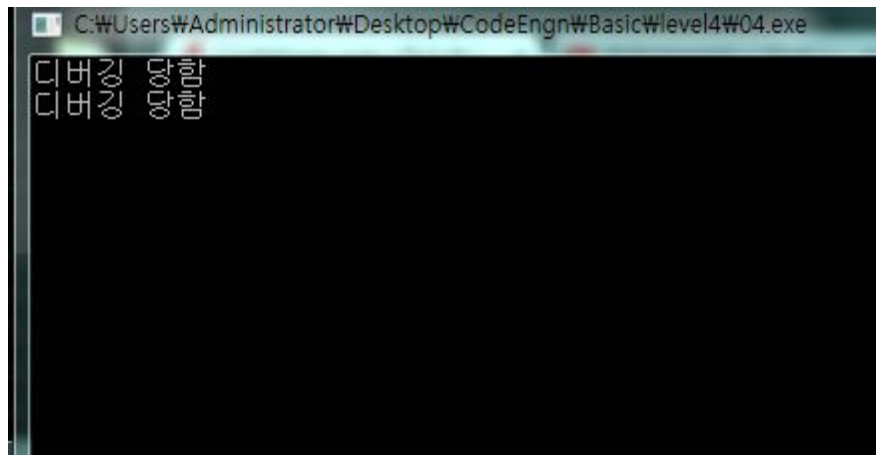
디버거를 탐지하는 함수...



일단 프로그램을 실행시켰더니 이렇게 나옵니다.

‘ 정상 정상 정상 정상 정상 .... ‘

olydbg 로 열어보겠습니다.



디버깅 당했다고 합니다.

불상하네요

일정한 간격으로 디버깅 당함 이라는 메시지가 호출됩니다.

그렇다면 시간 간격으로 호출하는 함수가 있다는 소리인데

그 함수를 찾아서 호출 부분을보면 도움이 될 것 같습니다.

00401041	. 80 000000	mov	eax, 00000000	
00401046	. F3:AB	rep	stos dword ptr es:[edi]	
00401048	> 8BF4	mov	esi, esp	
0040104A	. 68 E8030000	push	3E8	Timeout = 1000. ms
0040104F	. FF15 68B14300	call	near dword ptr ds:[<&KERNEL32.Sleep	Sleep
00401055	. 3BF4	cmp	esi, esp	
00401057	. E8 B4710000	call	04.00408210	
0040105C	. 8BF4	mov	esi, esp	
0040105E	. FF15 64B14300	call	near dword ptr ds:[<&KERNEL32.IsDebuggerPresent	kernel32.IsDebuggerPresent
00401064	. 3BF4	cmp	esi, esp	
00401066	. E8 A5710000	call	04.00408210	
0040106B	. 85C0	test	eax, eax	
0040106D	. 74 0F	je	short 04.0040107E	
0040106F	. 68 24104300	push	04.00431024	Arg1 = 00431024
00401074	. E8 17710000	call	04.00408190	04.00408190
00401079	. 83C4 04	add	esp, 4	
0040107C	. EB 0D	jmp	short 04.0040108B	
0040107E	> 68 1C104300	push	04.0043101C	Arg1 = 0043101C
00401083	. E8 08710000	call	04.00408190	04.00408190
00401088	. 83C4 04	add	esp, 4	

sleep 함수로 호출됩니다.

계속 디버깅을 한 결과 00401074 로 가서 디버깅 당함을 호출합니다.

00401033	. 83EC 40	sub esp, 40		
00401036	. 53	push ebx		
00401037	. 56	push esi		
00401038	. 57	push edi		
00401039	. 8D7D C0	lea edi, dword ptr ss:[ebp-40]		
0040103C	. B9 10000000	mov ecx, 10		
00401041	. B8 CCCCCCCC	mov eax, CCCCCCCC		
00401046	. F3:AB	rep stos dword ptr es:[edi]		
00401048	> 8BF4	mov esi, esp		
0040104A	. 68 E0030000	push 3E8	Timeout = 1000. ms	
0040104F	. FF15 68B14300	call near dword ptr ds:[<&KERNEL32.S	Sleep	
00401055	. 3BF4	cmp esi, esp		
00401057	. E8 B4710000	call 04.00408210		
0040105C	. 8BF4	mov esi, esp		
00401061	. FF15 64B14300	call near dword ptr ds:[<&KERNEL32.I	kernel32.IsDebuggerPresent	
00401064	. 3BF4	cmp esi, esp		
00401066	. E8 A5710000	call 04.00408210		
0040106B	. 85C0	test eax, eax		
0040106D	. 74 0F	je short 04.0040107E		
0040106F	. 68 24104300	push 04.00431024	Arg1 = 00431024	
00401074	. E8 17710000	call 04.00408190	04.00408190	
00401079	. 83C4 04	add esp, 4		
0040107C	. EB 00	jmp short 04.0040108B		
0040107E	> 68 1C104300	push 04.0043101C	Arg1 = 0043101C	
00401083	. E8 08710000	call 04.00408190	04.00408190	
00401088	. 83C4 04	add esp, 4		
0040108B	> EB B8	jmp short 04.00401048		
0040108D	. CC	int3		

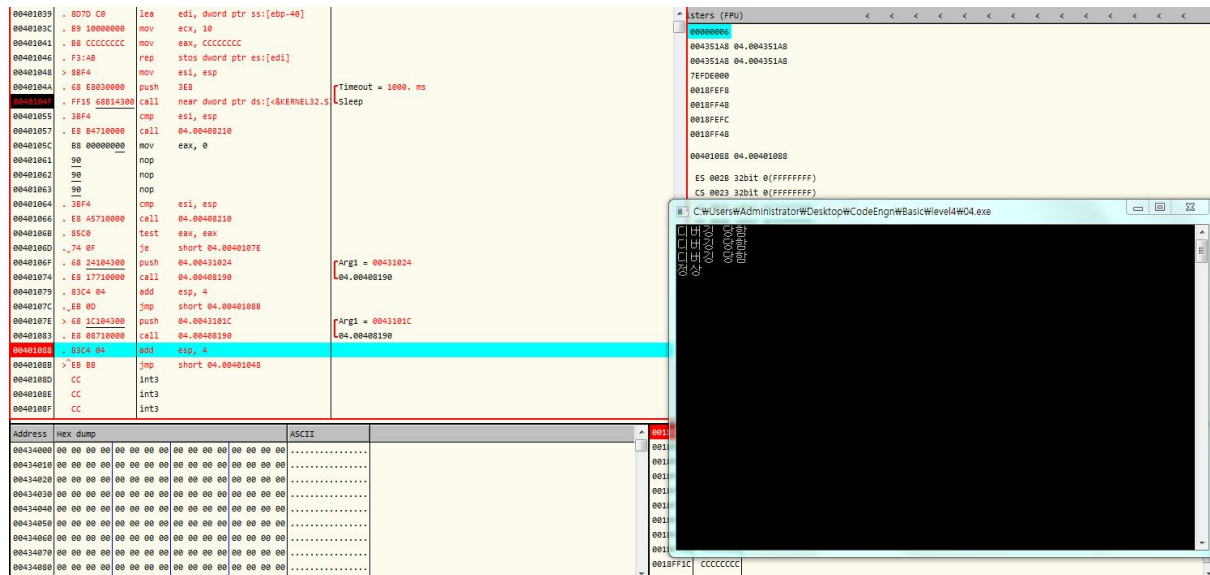
00401033	. 83EC 40	sub esp, 40		
00401036	. 53	push ebx		
00401037	. 56	push esi		
00401038	. 57	push edi		
00401039	. 8D7D C0	lea edi, dword ptr ss:[ebp-40]		
0040103C	. B9 10000000	mov ecx, 10		
00401041	. B8 CCCCCCCC	mov eax, CCCCCCCC		
00401046	. F3:AB	rep stos dword ptr es:[edi]		
00401048	> 8BF4	mov esi, esp		
0040104A	. 68 E0030000	push 3E8	Timeout = 1000. ms	
0040104F	. FF15 68B14300	call near dword ptr ds:[<&KERNEL32.S	Sleep	
00401055	. 3BF4	cmp esi, esp		
00401057	. E8 B4710000	call 04.00408210		
0040105C	. 8BF4	mov esi, esp		
00401061	. FF15 64B14300	call near dword ptr ds:[<&KERNEL32.I	kernel32.IsDebuggerPresent	
00401064	. 3BF4	cmp esi, esp		
00401066	. E8 A5710000	call 04.00408210		
0040106B	. 85C0	test eax, eax		
0040106D	. 74 0F	je short 04.0040107E		
0040106F	. 68 24104300	push 04.00431024	Arg1 = 00431024	
00401074	. E8 17710000	call 04.00408190	04.00408190	
00401079	. 83C4 04	add esp, 4		
0040107C	. EB 00	jmp short 04.0040108B		
0040107E	> 68 1C104300	push 04.0043101C	Arg1 = 0043101C	
00401083	. E8 08710000	call 04.00408190	04.00408190	
00401088	. 83C4 04	add esp, 4		
0040108B	> EB B8	jmp short 04.00401048		
0040108D	. CC	int3		

두 사진의 차이점을 아시나요?

바로 eax 값입니다. 리턴되는 값이 0 에서 1로 바뀌고

디버깅 당함 이라는 문자열이 호출됩니다.

그렇다면 1을 0으로 바꿔보겠습니다.



중간에 mov eax,0 이라는 명령어를 삽입하여

eax 값에다가 0으로 넣었습니다.

그리고 디버깅을 해본 결과 정상이라고 호출됩니다.

그렇다면 디버깅을 탐지하는 함수의 이름은 무엇일까요

IsDebuggerPresent : 디버깅당하고 있는지 여부를 확인하는 함수.

key flag : IsDebuggerPresent