

codeengn-basic-L19 풀이

리버싱 문제풀이 / Wonlf / 2022. 4. 22. 16:38

Basic RCE L19

이 프로그램은 몇 밀리세컨드 후에 종료 되는가

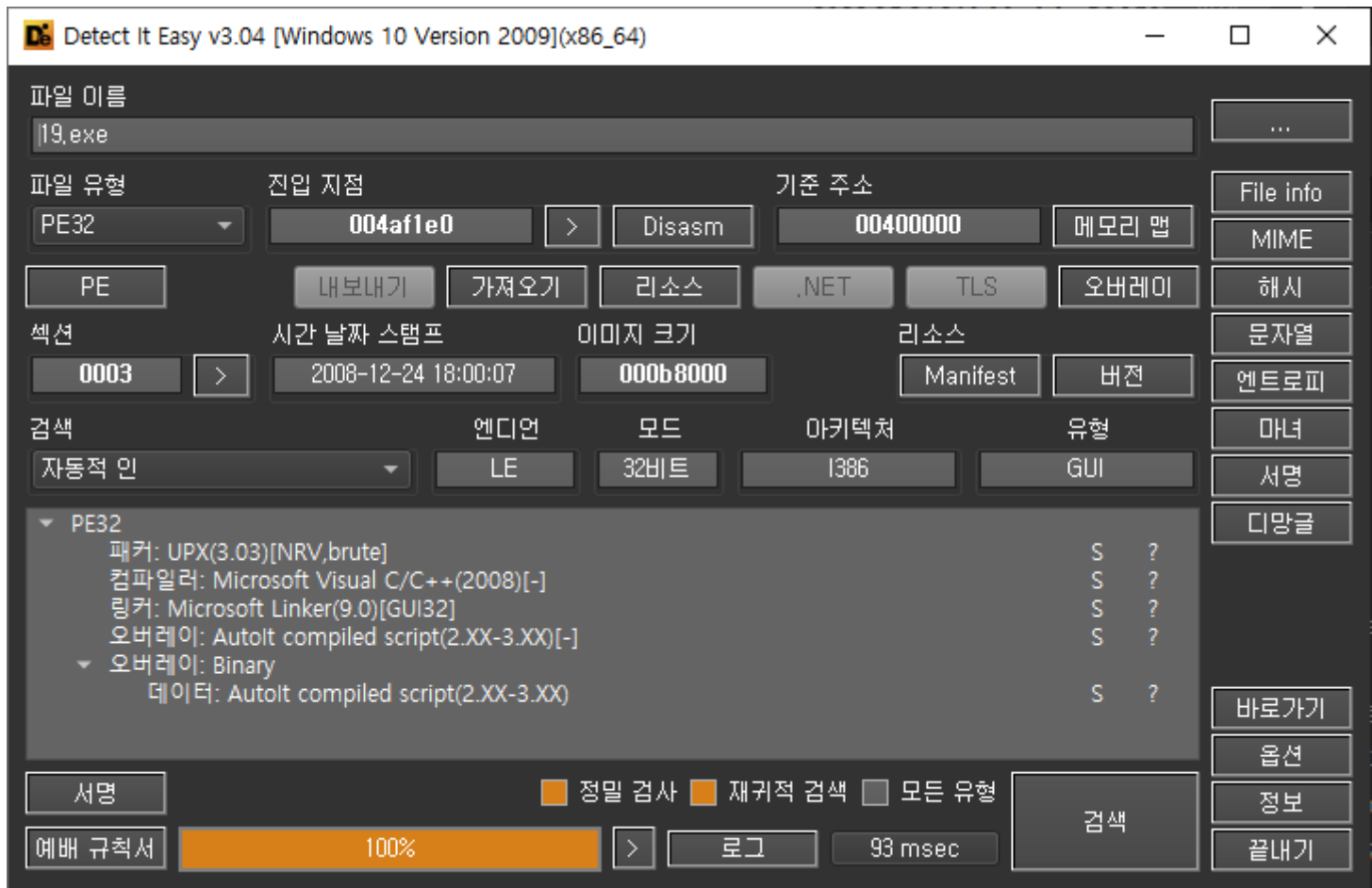
— Author: CodeEngn

— File Password: codeengn



문제는 프로그램이 몇 밀리세컨드 후에 종료 되는지를 원한다.

Die로 열어보면,



UPX패킹이 되어있다.

```
PS C:\Users\A02\Downloads\19 (2)> C:\Users\A02\Desktop\도구\upx-3.96-win64\upx.exe -d "C:\Users\A02\Downloads\19 (2)\19.exe"

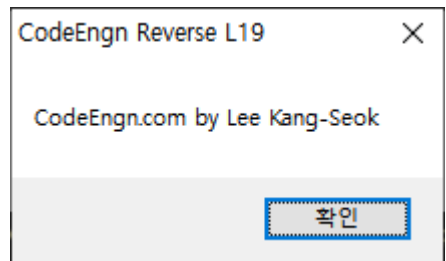
      Ultimate Packer for eXecutables
      Copyright (C) 1996 - 2020
UPX 3.96w      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

-----
File size      Ratio      Format      Name
-----
613176 <-    290616    47.40%    win32/pe    19.exe

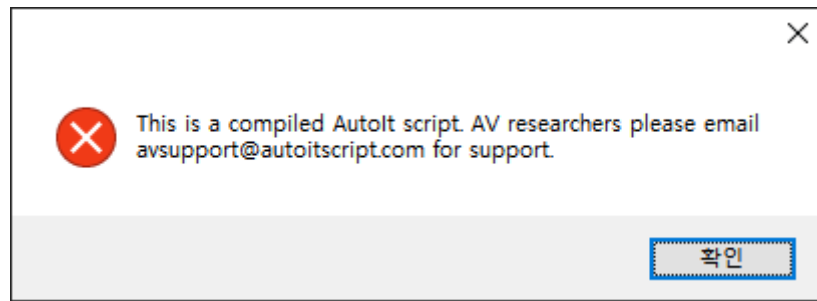
Unpacked 1 file.
```

언패커로 패킹을 해제해주고

프로그램을 실행하면 메시지박스를 띄우고 사라진다.



디버거로 열어본다.



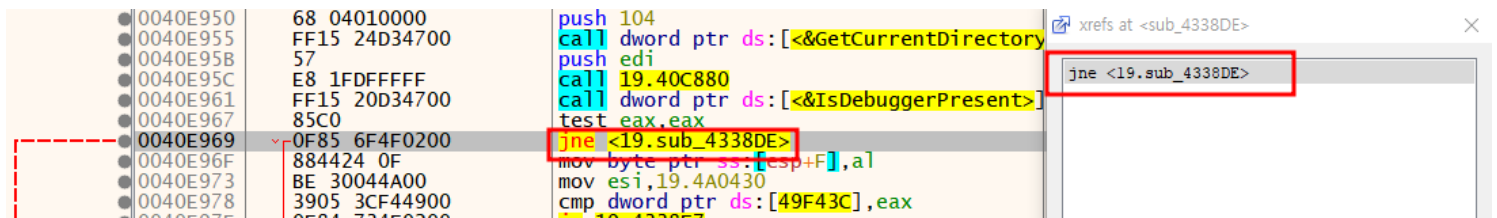
디버거로 열어서 실행하니 이상한 메시지 박스가 나온다. 문자열로 찾아 들어가서 확인해본다.

004338DE	[> 6A 10	push 10	sub_4338DE
004338E0	[. 68 9EF64700	push 19.47F69E	
004338E5	[. 68 A0F64700	push 19.47F6A0	
004338EA	[. 6A 00	push 0	47F6A0:"This is a compiled AutoIt script. AV researchers please email
004338EC	[. FF15 DCD64700	call dword ptr ds:[<&MessageBoxA	
004338F2	[. ^ E9 49B1DFFF	jmp <19.sub_40EA40>	

4338DE 함수에서 메시지 박스를 출력 하는데 이것을 호출하는 부분을 찾아보도록 한다.



함수 주소에 우클릭을 하고 외부참조버튼을 누르면,



```
0040E950 68 04010000 push 104
0040E955 FF15 24D34700 call dword ptr ds:[<&GetCurrentDirectory
0040E95B 57          push edi
0040E95C E8 1FDFFFFF call 19.40C880
0040E961 FF15 20D34700 call dword ptr ds:[<&IsDebuggerPresent>]
0040E967 85C0        test eax, eax
0040E969 0F85 6F4F0200 jne <19.sub_4338DE>
0040E96F 884424 0F   mov byte ptr ss:[esp+F], al
0040E973 BE 30044A00 mov esi, 19.4A0430
0040E978 3905 3CF44900 cmp dword ptr ds:[49F43C], eax
```

특정 창이 뜨고 이 함수를 호출하는 부분을 보여준다. 그 부분으로 가서 보니 4번 문제에 나왔던 IsDebuggerPresent 안티디버깅 함수를 사용하여 디버깅 당하고 있다면 특정 메시지 박스를 출력 하는 것이었다.

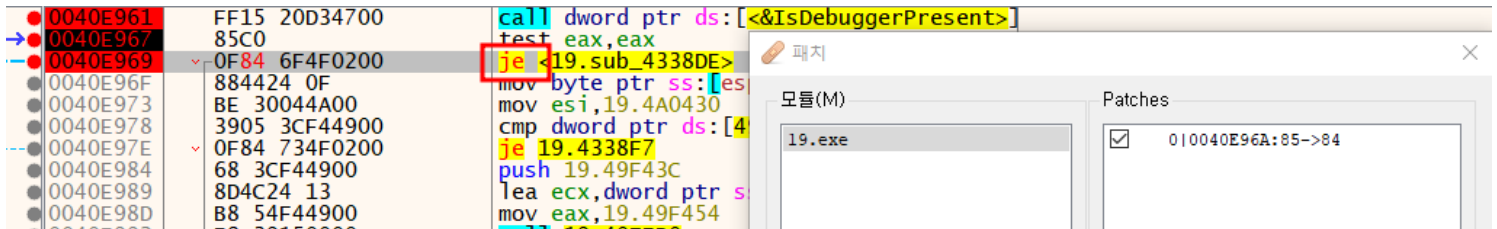
Return value

If the current process is running in the context of a debugger, the return value is nonzero.

If the current process is not running in the context of a debugger, the return value is zero.

test eax eax구문으로 0인지 아닌지 판단하고 0이라면 제로플래그가 셋팅이 됨으로 jne 점프를 하지 않게 되고, 0이 아닌 값이 들어있으면 제로플래그가 셋팅되지 않기 때문에 jne 점프를 하게 되고 특정 메시지 박스를 띄운다.

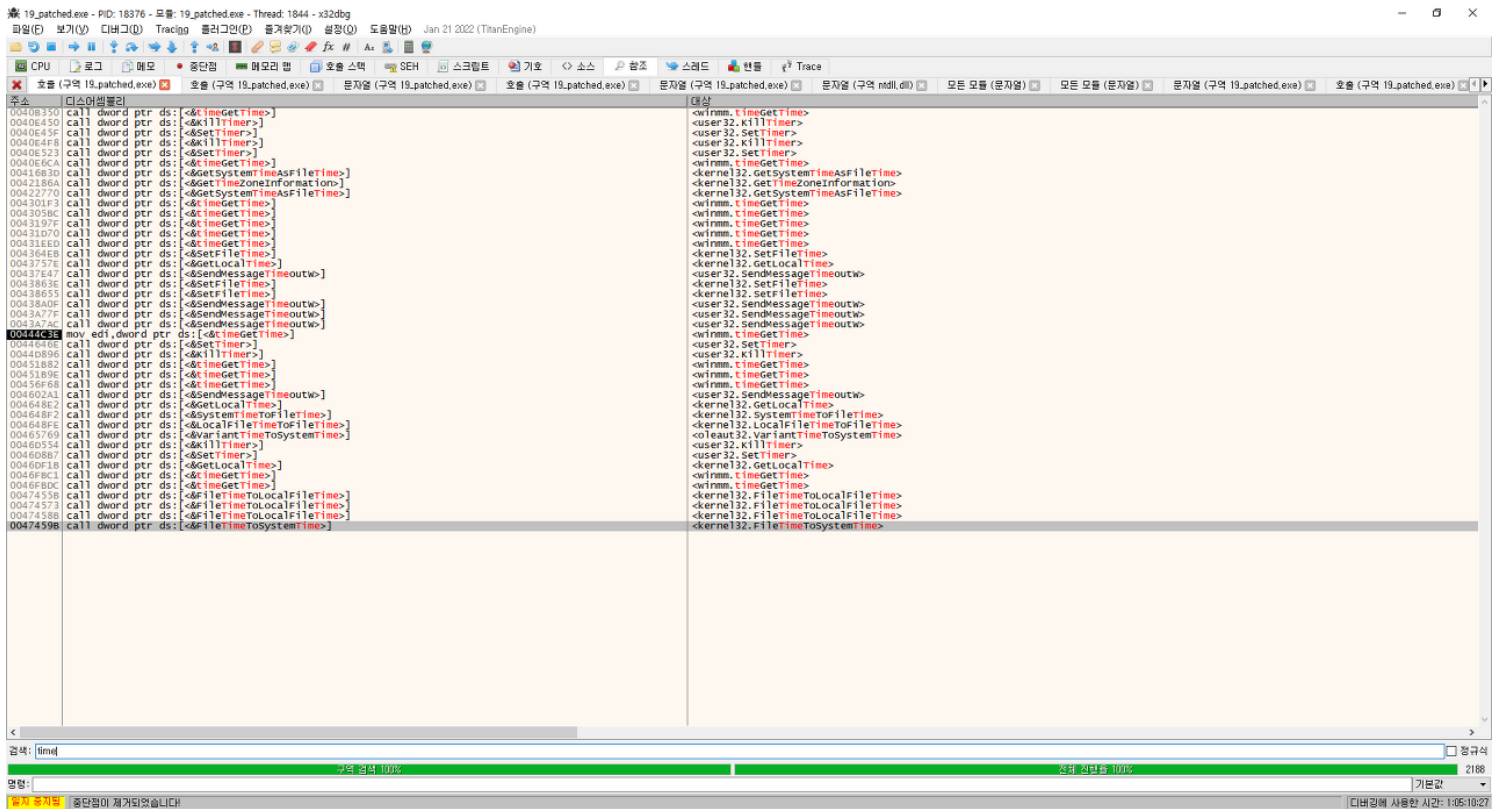
제로플래그가 세팅된 상태에서 점프하는 je구문으로 변경하게 되면 디버거로 열렸을때 제로플래그가 셋팅되지 않기 때문에 점프를 뛰지 않을 것이다. 그렇게 그 다음 구문으로 진행되기 때문에 jne부분을 je로 패치 후 저장한다.



```
0040E961 FF15 20D34700 call dword ptr ds:[<&IsDebuggerPresent>]
0040E967 85C0        test eax, eax
0040E969 0F84 6F4F0200 je <19.sub_4338DE>
0040E96F 884424 0F   mov byte ptr ss:[esp+F], al
0040E973 BE 30044A00 mov esi, 19.4A0430
0040E978 3905 3CF44900 cmp dword ptr ds:[49F43C], eax
0040E97E 0F84 734F0200 je 19.4338F7
0040E984 68 3CF44900 push 19.49F43C
0040E989 8D4C24 13   lea ecx, dword ptr s
0040E98D B8 54F44900 mov eax, 19.49F454
```

그리고 패치한 파일을 디버거로 다시 열어주면, 잘 작동 한다.
이제 문제의 목표인 몇 밀리 세컨드 이후 종료되는지 알아보자.

시간에 관련된 함수를 사용할 것이다. 호출하는 함수 중에 time이 들어가는 함수를 찾아본다.



time이 들어가는 많은 함수들이 있었지만, 전부 검색해보니 지금 문제에서 원하는 부분과 가장 연관성이 있는 함수는 timeGetTime 함수였다.

timeGetTime() 함수는

윈도우(운영체제)가 시작되어서 지금까지 흐른 시간을 1/1000 초 (milliseconds) 단위로 DWORD형을 리턴하는 함수다.

timeGetTime 함수만 남긴 뒤, 전부 브레이크 포인트를 걸고 디버깅을 해본다.

00444C3E	8B3D 58D74700	mov edi,dword ptr ds:[&timeGetTime]	
00444C44	FFD7	call edi	timeGetTime 호출
00444C46	803D D3E84800 00	cmp byte ptr ds:[48E8D3],0	
00444C4D	8BF0	mov esi,eax	timeGetTime 반환값 esi에 담음
00444C4F	0F84 FF000000	je 19_patched.444D54	
00444C55	8B5C24 14	mov ebx,dword ptr ss:[esp+14]	[esp+14]:"\$A"
00444C59	8B2D 58D14700	mov ebp,dword ptr ds:[&sleep]	ebp = sleep함수
00444C5F	FFD7	call edi	timeGetTime 한번더 호출 (시간이 더 지나있겠죠)
00444C61	3BC6	cmp eax,esi	지난 시간과 전의 시간을 비교
00444C63	0F83 CF000000	jae 19_patched.444D38	지난 시간과 전의 시간을 비교하니 무조건 점프

함수를 호출하는데서 걸리고 한줄씩 내려가며 해석해보았다. 점프한 뒤의 구문도 보게되면,

00444D38	> 2BC6	sub eax,esi	흐른 뒤 시간 에서 전의 시간을 뺀다
00444D3A	> 3B43 04	cmp eax,dword ptr ds:[ebx+4]	빼기한 결과를 ebx+4에 있는 값과 비교
00444D3D	^ 0F83 2EFFFFFF	jae 19_patched.444C71	빼기한 결과가 ebx+4에 있는 값보다 크면 다시 시간 연산
00444D43	. 6A 0A	push A	
00444D45	. FFD5	call ebp	
00444D47	. 803D D3E84800 00	cmp byte ptr ds:[48E8D3],0	
00444D4E	^ 0F85 0BFFFFFF	jne 19_patched.444C5F	
00444D54	> 5F	pop edi	
00444D55	. 5E	pop esi	
00444D56	. 5D	pop ebp	
00444D57	. 33C0	xor eax,eax	
00444D59	. 5B	pop ebx	ebx: " R"
00444D5A	. C2 0400	ret 4	

종합해보면, 흐른 시간(연산한 값)이 ebx+4보다 클 때까지 계속 다시 시간을 가져오는 함수로 점프한다.

ebx+4에 있는 값이 flag가 될 것이다.

ebx+4주소에 접근하여 살펴보면 dword만큼 가져와서 비교하니 2byte인 0x2B70이 되겠다.
이것을 10진수로 바꾸어 페이지에 인증해준다.