

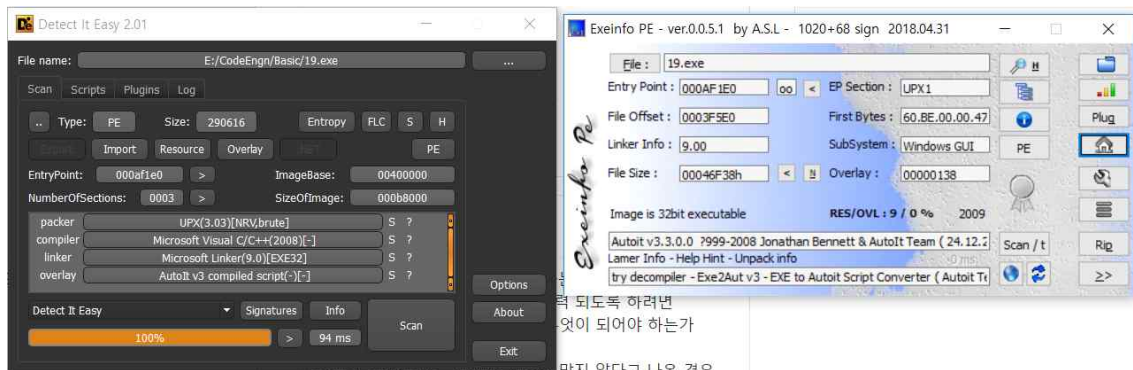
2019.02.21. CodeEngn basic 19

Tree to Tree



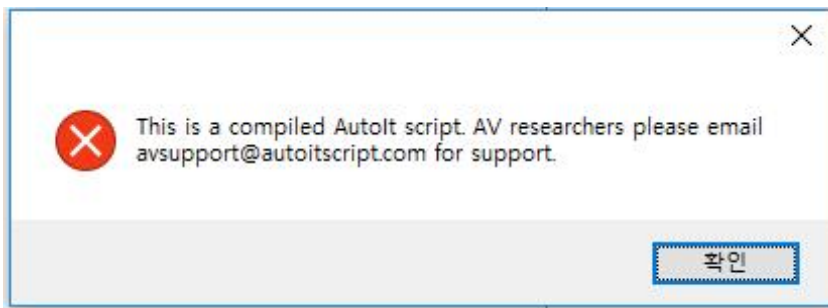
몇 밀리세컨드 후에 종료되는지 찾아야되는 문제
basic지금까지 문제 중 가장 오래걸림...

우선 UPX로 패되어있고 AutoIt이라는 Window응용프로그램 쉽게 만들어주는 툴? 로 만들어
졌다고 나온다.



일반적으로 실행시키면 위처럼 창이 뜨고 약 11초정도 후에 꺼진다.

디버거로 실행시키니까 아래 창처럼 뜨고 종료됨.
안티디버깅 되어있음.

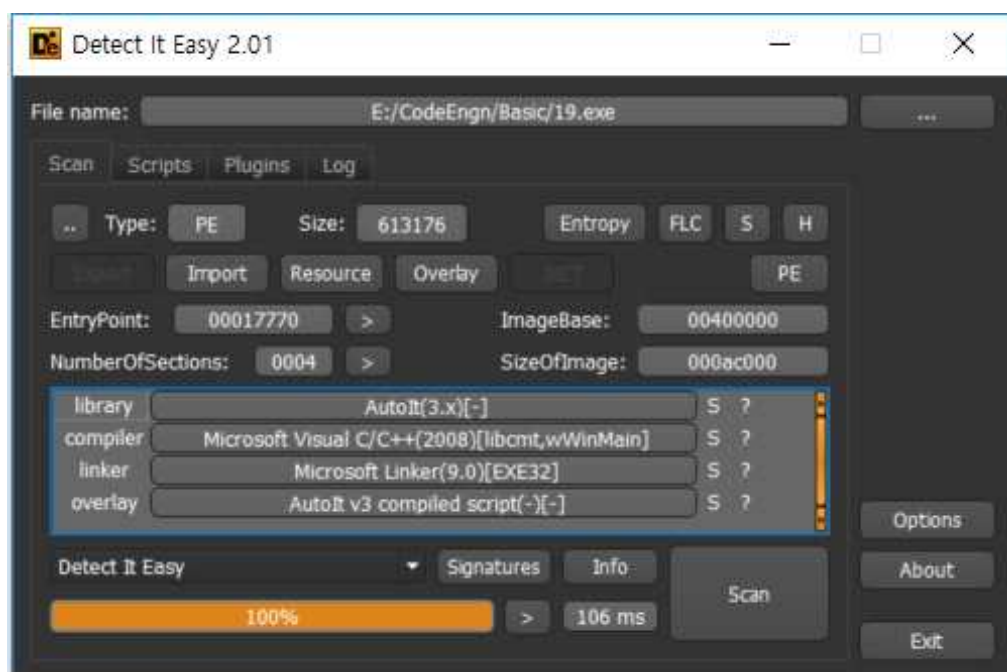


우선 완벽하게 풀기위해 UPX만든 사람이 배포한 툴을 이용해서 unpack

```
UPX comes with ABSOLUTELY NO WARRANTY; for details visit https://upx.github.io
C:\Users\kwlee>C:\Users\kwlee\OneDrive\upx.exe -d E:\CodeEngn\Basic\19.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2018
UPX 3.95w Markus Oberhumer, Laszlo Molnar & John Reiser Aug 26th 2018

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

Unpacked 1 file.
C:\Users\kwlee>
```



언팩된 모습.

어떤종류의 안티디버깅인지 찾아보니 IsDebuggerPresent를 사용한다는 사실을 찾음
다행이다 가장 쉽고생각하는 안티디버깅! TLS도 한번 풀어보고싶다.

004176FA	5b	push esi	
004176FB	68 00 00 40 00	push 19,400000	
00417700	E8 4B 73 FF FF	call 19,40EA5D	
00417705	89 45 E0	mov dword ptr ss:[ebp-20],eax	[ebp-20]:EntryPoint

주소	유형	Ordinal	Symbol
0047D320	가져오기		IsDebuggerPresent
0047D4C4	가져오기		IsDialogMessageW
0047D520	가져오기		IsDlgButtonChecked

766BEC50	64: A1 30 00 00 00	mov eax,dword ptr ds:[30]
766BEC56	0F B6 40 02	movzx eax,byte ptr ds:[eax+2]
766BEC5A	C3	ret

003BD000	00 00 01 00	
----------	-------------	--

PEB+2에 있는 값만 00으로 바꿔주면 회피가능

0040E940	81 EC 38 04 00 00	sub esp,438	
0040E946	53	push ebx	
0040E947	56	push esi	
0040E948	57	push edi	
0040E949	8B F8	mov edi,eax	
0040E94B	8D 44 24 20	lea eax,dword ptr ss:[esp+20]	
0040E94F	50	push eax	
0040E950	68 04 01 00 00	push 104	
0040E955	FF 15 24 D3 47 00	call dword ptr ds:[<GetCurrentDirectoryW>]	
0040E958	57	push edi	
0040E95C	E8 1F DF FF FF	call 19,40C880	
0040E961	FF 15 20 D3 47 00	call dword ptr ds:[<IsDebuggerPresent>]	
0040E967	85 C0	test eax,eax	
0040E969	0F 85 6F 4F 02 00	jne 19,43380E	
0040E96F	88 44 24 0F	mov byte ptr ss:[esp+F],al	
0040E973	8E 30 04 4A 00	mov esi,19,4A0490	
0040E978	39 05 3C F4 49 00	cmp dword ptr ds:[49F43C],eax	
0040E97E	0F 84 73 F4 02 00	je 19,4338F7	
0040E984	68 3C F4 49 00	push 19,49F43C	
0040E989	8D 4C 24 13	lea ecx,dword ptr ss:[esp+13]	
0040E98D	B8 54 F4 49 00	mov eax,19,49F454	49F454:L"E:\CodeEngn\Basic\19.exe"
0040E992	E8 39 15 00 00	call 19,40FED0	
0040E997	84 C0	test al,al	
0040E999	0F 84 78 F4 02 00	je 19,43391A	
0040E99F	8A 0D 30 04 4A 00	mov cl,byte ptr ds:[4A0430]	
0040E9A5	8A 1D 31 04 4A 00	mov bl,byte ptr ds:[4A0431]	
0040E9AB	68 38 F4 49 00	push 19,49F438	49F438:&"19.exe"
0040E9B0	8D 94 24 34 02 00 00	lea edx,dword ptr ss:[esp+234]	
0040E9B7	52	push edx	
0040E9B8	68 04 01 00 00	push 104	
0040E9BD	68 54 F4 49 00	push 19,49F454	
0040E9C2	88 0D 40 F4 49 00	mov byte ptr ds:[49F440],cl	49F454:L"E:\CodeEngn\Basic\19.exe"
0040E9C8	FF 15 F4 D2 47 00	call dword ptr ds:[<GetFullPathNameW>]	
0040E9CE	A1 3C F4 49 00	mov eax,dword ptr ds:[49F43C]	
0040E9D3	50	push eax	
0040E9D4	68 54 F4 49 00	push 19,49F454	49F454:L"E:\CodeEngn\Basic\19.exe"
0040E9D9	E8 82 2D FF FF	call 19,401760	
0040E9DE	85 C0	test eax,eax	
0040E9E0	0F 85 48 F4 02 00	jne 19,43392E	
0040E9E6	80 FB 01	cmp bl,1	
0040E9E9	0F 84 63 F4 02 00	je 19,433952	

그 후에 Message창이 띄워지는 부분을 계속 트레이싱하면서 변화들을 관찰하던중에

유형	주소	Module/Label/Exception	상태	디스어셈블리	Hits	Sur
소프트웨어	0040821A	19.exe	종성화됨	call 19.exe,40BC70	1	
	00408CC0	19.exe	종성화됨	call eax	1	
	0040EA13	19.exe	종성화됨	call 19.exe,40AEF0	1	
	0040EAF9	19.exe	종성화됨	call 19.exe,40E940	1	
	00417700	19.exe	종성화됨	call 19.exe,40EA50	1	
	00444DB8	19.exe	종성화됨	call dword ptr ds:[<MessageBox>]	1	
	0045E05F	19.exe	종성화됨	call 19.exe,44405D	1	
	745A7868	user32.dll	종성화됨	call user32,SoftModalMessageBox	1	
	745A8260	user32.dll	종성화됨	call user32,745A75C8	1	
	745A82A5	user32.dll	종성화됨	call user32,MessageBoxTimeoutW	1	
	745A89E2	user32.dll	종성화됨	call user32,74544A28	1	

MessageBoxW로 진입하는 순간에 스레드가 실행되고 약 11초뒤에 6번스레드가 종료됨

Assembly code snippet:

```

00444D73  C605 D3E84800 01  mov byte ptr ds:[48E8D3],1
00444D74  FF15 5CD14700  call dword ptr ds:[<GetCurrentThreadId>]
00444D75  894424 08      mov dword ptr ss:[esp+8],eax
00444D76  8D4424 24      lea eax,dword ptr ss:[esp+24]
00444D77  50          push eax
00444D78  56          push esi
00444D79  8D4C24 10      lea ecx,dword ptr ss:[esp+10]
00444D7A  51          push ecx
00444D7B  68 3A4C4400  push 19패치.444C3A
00444D7C  56          push esi
00444D7D  897C24 24      mov dword ptr ss:[esp+24],edi
00444D7E  E8 2715FDFF  call 19패치.4162C5
00444D7F  83C4 18      add esp,18
00444D80  88F0      mov esi,edx
00444D81  8B5424 20      mov edx,dword ptr ss:[esp+20]
00444D82  8B4424 1C      mov eax,dword ptr ss:[esp+1C]
00444D83  8B4C24 18      mov ecx,dword ptr ss:[esp+18]
00444D84  52          push edx
00444D85  8B5424 18      mov edx,dword ptr ss:[esp+18]
00444D86  50          push eax
00444D87  51          push ecx
00444D88  52          push edx
00444D89  FF15 9CD64700  call dword ptr ds:[<MessageBoxW>]
00444D8A  8BF8      mov edi,edx
00444D8B  85F6      test esi,esi
00444D8C  74 17      je 39패치.444D0B
00444D8D  6A FF      push FFFFFFFF
00444D8E  56          push esi

```

Thread List:

번호	ID	진입점	TEB	EIP	일시중지 횟수	우선 순위	대기사유
주요	11E0	00417770	00359000	00417770	0	보통	Executive
4	88C	777B1440	00365000	777CC58C	0	보통	Suspended
1	3438	777B1440	0035C000	777CC58C	0	보통	Suspended
6	4D08	00416243	00368000	777CC58C	0	보통	Suspended
2	398C	777B1440	0035F000	777CC58C	0	보통	Suspended
3	3650	777B1440	00362000	777CC58C	0	보통	Suspended
5	3F94	777B1440	00368000	777CC58C	0	보통	Suspended

6번스레드가 실행된모습

Thread List:

번호	ID	진입점	TEB	EIP	일시중지 횟수	우선 순위	대기사유
주요	11E0	00417770	00359000	745A82A5	0	보통	Executive
4	88C	777B1440	00365000	777CC58C	0	보통	Suspended
1	3438	777B1440	0035C000	777CC58C	0	보통	Suspended
6	4D08	00416243	00368000	777CC58C	0	보통	Suspended
2	398C	777B1440	0035F000	777CC58C	0	보통	Suspended
3	3650	777B1440	00362000	777CC58C	0	보통	Suspended
5	3F94	777B1440	00368000	777CC58C	0	보통	Suspended

Thread Details for Thread 6 (ID 4D08):

번호	ID	진입점	TEB	EIP	일시중지 횟수	우선 순위	대기사유	마지막 오류	사용자 시간	커널 시간	실행 시간
주요	11E0	00417770	00359000	76D5289C	0	보통	WrUserRequest	00000000	00:00:00.0156250	00:00:00.2343750	16:15:32.161
4	88C	777B1440	00365000	777CC58C	0	보통	WrQueue	00000000	00:00:00.0000000	00:00:00.0000000	16:15:35.161
1	3438	777B1440	0035C000	777CC58C	0	보통	WrQueue	00000000	00:00:00.0000000	00:00:00.0000000	16:15:32.161
6	4D08	00416243	00368000	777CC58C	0	보통	DelayExecution	00000000	00:00:00.0000000	00:00:00.0000000	16:18:53.161
2	398C	777B1440	0035F000	777CC58C	0	보통	WrQueue	00000000	00:00:00.0000000	00:00:00.0156250	16:15:32.161
3	3650	777B1440	00362000	777CC58C	0	보통	WrQueue	00000000	00:00:00.0000000	00:00:00.0156250	16:15:32.161
5	3F94	777B1440	00368000	777CC58C	0	보통	WrQueue	000000EA	00:00:00.0000000	00:00:00.0156250	16:15:36.161

이제 MessageBoxW전 명령어들만 보면 된다고 생각하여 레지스터들을 유심히 확인

Assembly code snippet:

```

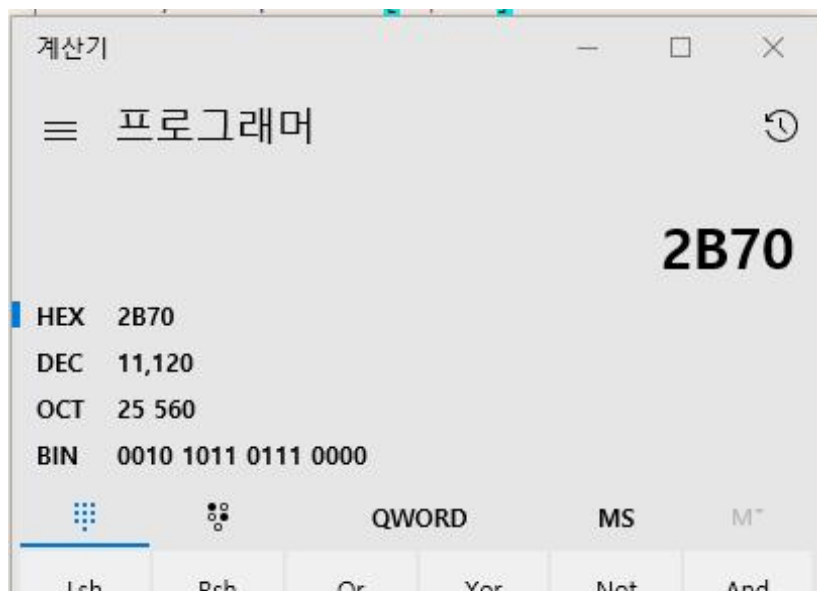
00444D62  8B7C24 24      mov edi,dword ptr ss:[esp+24]
00444D63  33F6      xor esi,esi
00444D64  C605 D2E84800 00  mov byte ptr ds:[48E8D2],0
00444D65  85FF      test edi,edi
00444D66  74 31      je 19패치.444DA4
00444D67  C605 D3E84800 01  mov byte ptr ds:[48E8D3],1
00444D68  FF15 5CD14700  call dword ptr ds:[<GetCurrentThreadId>]

```

mov edi, dword ptr ss:[esp+24] 부분이 밀리세컨드를 저장하는 부분

Register Window:

Register	Value	Comment
EAX	01BCED20	L"CodeEngn Reverse L19"
EBX	00000000	
ECX	01BCEBD8	
EDX	00002870	
EBP	01BCEE10	L"CodeEngn.com by Lee Kang-Seok"
ESP	008AF880	"x\"H"
ESI	00010000	
EDI	00002870	



11120

Clear