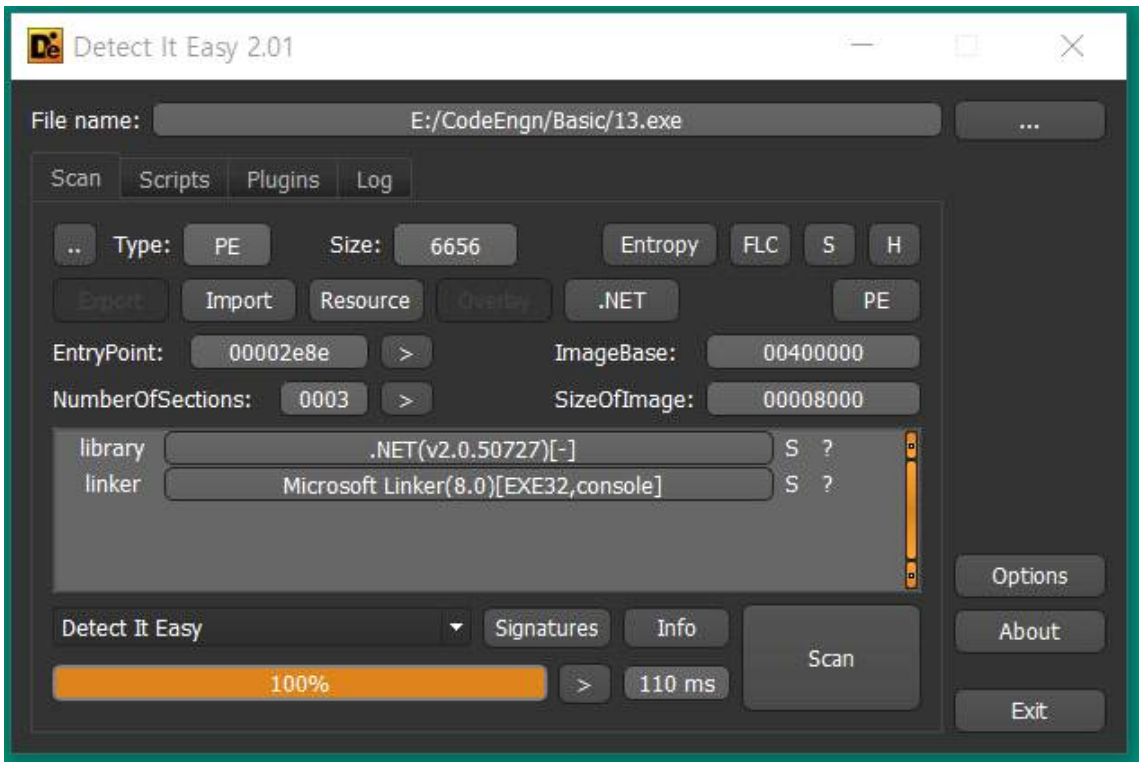
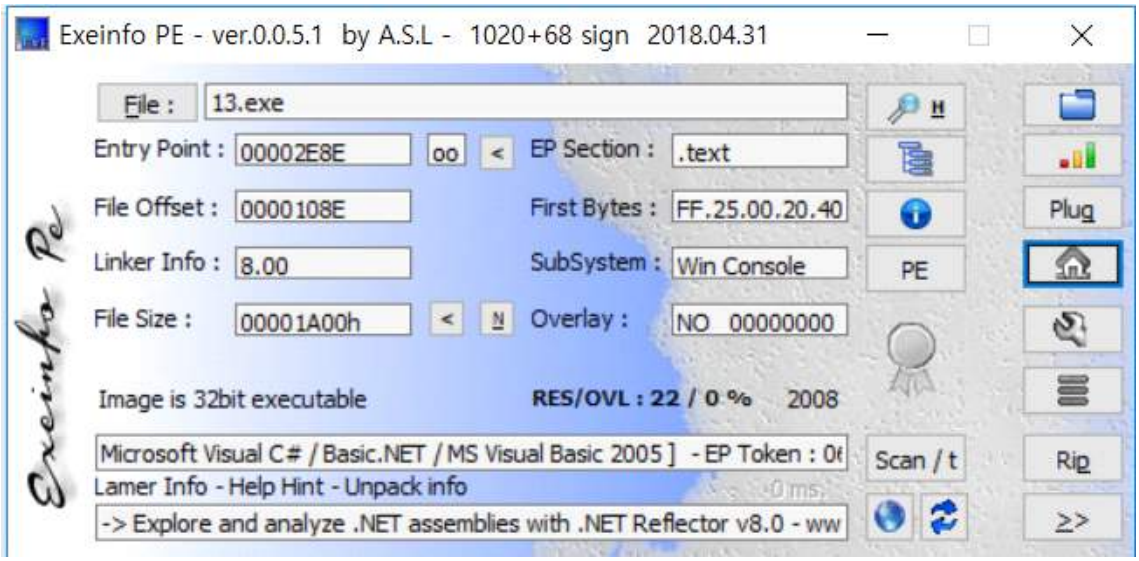
19. 02. 16 CodeEngn Basic RCE L13
Tree to Tree

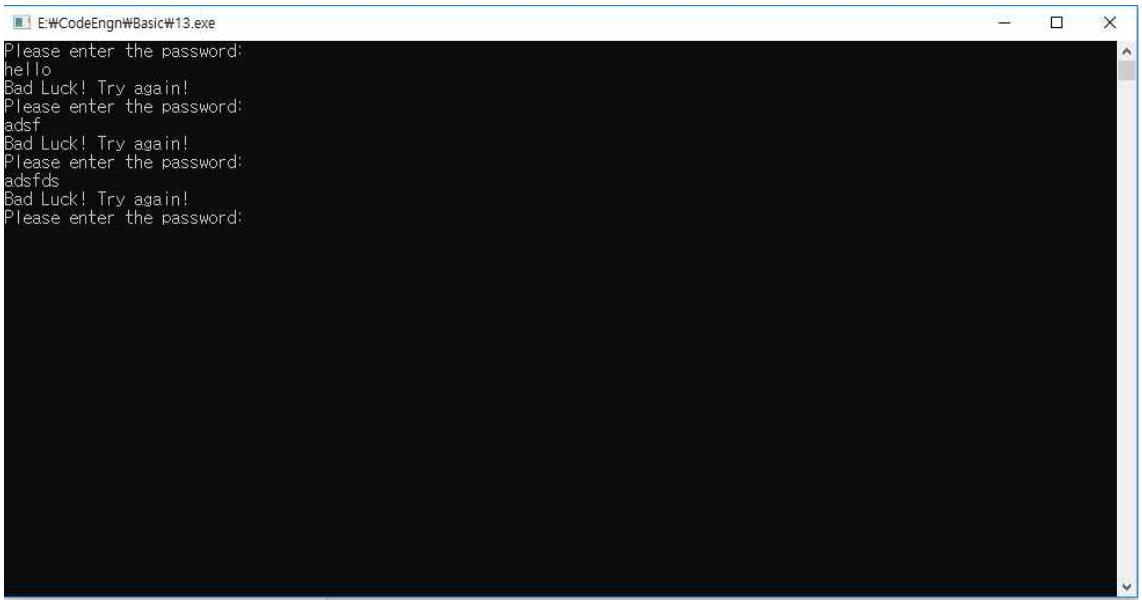
정답은 무엇인가. 심플하다.



문제에서 처음으로 .NET library가 등장.
.NET 프레임워크를 사용하는 언어들로 작성된 소스 코드는 각 언어에 맞는 컴파일러를 거쳐 .NET CLR용 중간 코드인 CIL(Common Intermediate Language)로 컴파일된 후 .exe 파일로 래핑(wrapping)된다. 그리고 .NET CLR은 이 파일을 JIT 컴파일 방식으로 읽어들여 기계어 번역을 수행한다. CIL은 .NET CLR이 설치된 곳이라면 어디서든 컴파일이 가능하며, Java 바이트코드처럼 어셈블리어와 유사한 형태를 띠고 있다.
라는 위키.

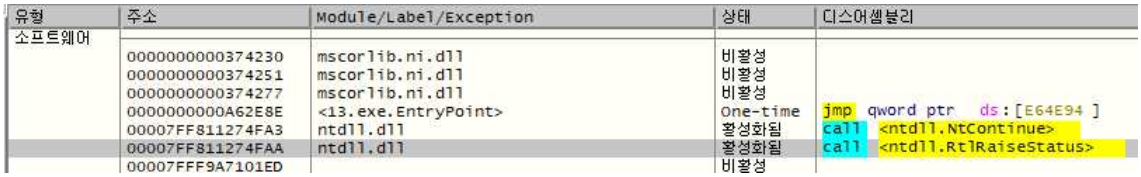


C#을 위한 플랫폼이라고함.



우선 암호를 찾기위한 트레이싱이 시작되었다.
입력한값과 비교하는 곳을 찾기위해 각 함수에 breakpoint를 걸어주고 값을 입력

breakpoint는 이러한 값들에 걸어놓았다.

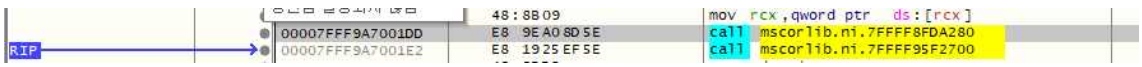


유형	주소	Module/Label/Exception	상태	디스어셈블리
소프트웨어				
	0000000000374230	mscorlib.ni.dll	비활성	
	0000000000374251	mscorlib.ni.dll	비활성	
	0000000000374277	mscorlib.ni.dll	비활성	
	000000000A62E8E	<13.exe.EntryPoint>	One-time	jmp qword ptr ds:[E64E94]
	00007FF811274FA3	ntdll.dll	활성화됨	call <ntdll.NtContinue>
	00007FF811274FAA	ntdll.dll	활성화됨	call <ntdll.RtlRaiseStatus>
	00007FFF9A7101ED		비활성	

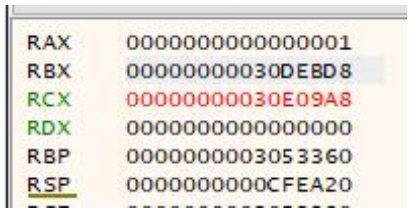
트레이싱하다보니 hello 라는 문자열을 하나씩 읽고 \r\n개행 문자를 만났을 때 넘어가는 어셈블리어 코드를 발견.

\r\n을만났을 때 점프하는곳으로 간 후 부터는 breakpoint를 하드웨어적으로 걸어놔야 멈춘다.

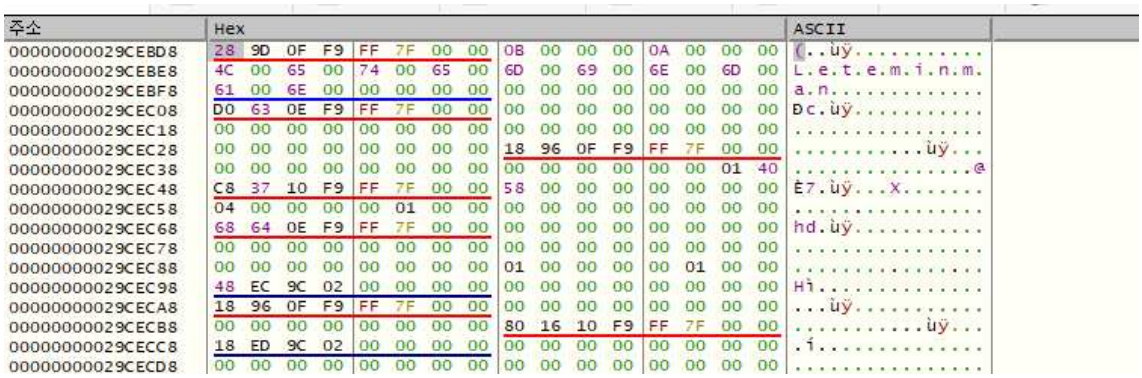
그렇게 계속 트레이싱하다가



위 라인을 만났을 때 Bad Luck! Try again이 실행됨.
유심히 레지스터 변화를 관찰하다가.

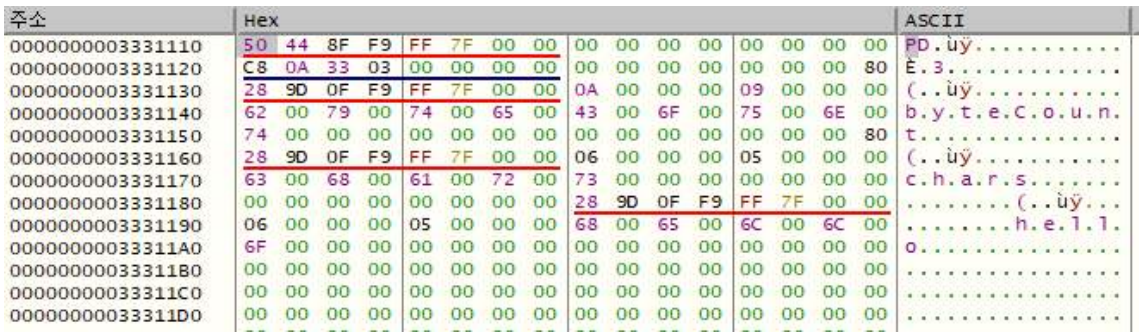


RBX레지스터에 쓰인 주소의 헥사코드를 보니 L.e.t.e.m.i.n.m.a.n이라는 문구 발견



주소	Hex																ASCII
00000000029CEBD8	28	9D	0F	F9	FF	7F	00	00	0B	00	00	00	0A	00	00	00	(..ùÿ.....
00000000029CEBE8	4C	00	65	00	74	00	65	00	6D	00	69	00	6E	00	6D	00	L.e.t.e.m.i.n.m.
00000000029CEBF8	61	00	6E	00	00	00	00	00	00	00	00	00	00	00	00	00	a.n.....
00000000029CEC08	D0	63	0E	F9	FF	7F	00	00	00	00	00	00	00	00	00	00	Đc.ùÿ.....
00000000029CEC18	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000000029CEC28	00	00	00	00	00	00	00	00	18	96	0F	F9	FF	7F	00	00	.....ùÿ...
00000000029CEC38	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01	40	.....@
00000000029CEC48	C8	37	10	F9	FF	7F	00	00	58	00	00	00	00	00	00	00	È7.ùÿ...X.....
00000000029CEC58	04	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	.....
00000000029CEC68	68	64	0E	F9	FF	7F	00	00	00	00	00	00	00	00	00	00	hd.ùÿ.....
00000000029CEC78	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000000029CEC88	00	00	00	00	00	00	00	00	01	00	00	00	00	01	00	00	.....
00000000029CEC98	48	EC	9C	02	00	00	00	00	00	00	00	00	00	00	00	00	Hì.....
00000000029CECA8	18	96	0F	F9	FF	7F	00	00	00	00	00	00	00	00	00	00	...ùÿ.....
00000000029CECB8	00	00	00	00	00	00	00	00	80	16	10	F9	FF	7F	00	00	.....ùÿ...
00000000029CECC8	18	ED	9C	02	00	00	00	00	00	00	00	00	00	00	00	00	.í.....
00000000029CECD8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

그이전에 hello라는 문구가 저런식으로 저장되는 것을 봤었다.
byte수를 카운트하고 h.e.l.l.o 이런식으로 저장됨.



주소	Hex																ASCII
0000000003331110	50	44	8F	F9	FF	7F	00	00	00	00	00	00	00	00	00	00	PD.ùÿ.....
0000000003331120	C8	0A	33	03	00	00	00	00	00	00	00	00	00	00	00	80	È.3.....
0000000003331130	28	9D	0F	F9	FF	7F	00	00	0A	00	00	00	09	00	00	00	(..ùÿ.....
0000000003331140	62	00	79	00	74	00	65	00	43	00	6F	00	75	00	6E	00	b.y.t.e.C.o.u.n.
0000000003331150	74	00	00	00	00	00	00	00	00	00	00	00	00	00	00	80	t.....
0000000003331160	28	9D	0F	F9	FF	7F	00	00	06	00	00	00	05	00	00	00	(..ùÿ.....
0000000003331170	63	00	68	00	61	00	72	00	73	00	00	00	00	00	00	00	c.h.a.r.s.....
0000000003331180	00	00	00	00	00	00	00	00	28	9D	0F	F9	FF	7F	00	00	.....(..ùÿ...
0000000003331190	06	00	00	00	05	00	00	00	68	00	65	00	6C	00	6C	00	.....h.e.l.l.
00000000033311A0	6F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	o.....
00000000033311B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000000033311C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000000033311D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

결과적으로 입력해보니

Leteminman
Clear