

## 19. 02. 16 CodeEngn Basic RCE L13

Tree to Tree

Basic RCE L13

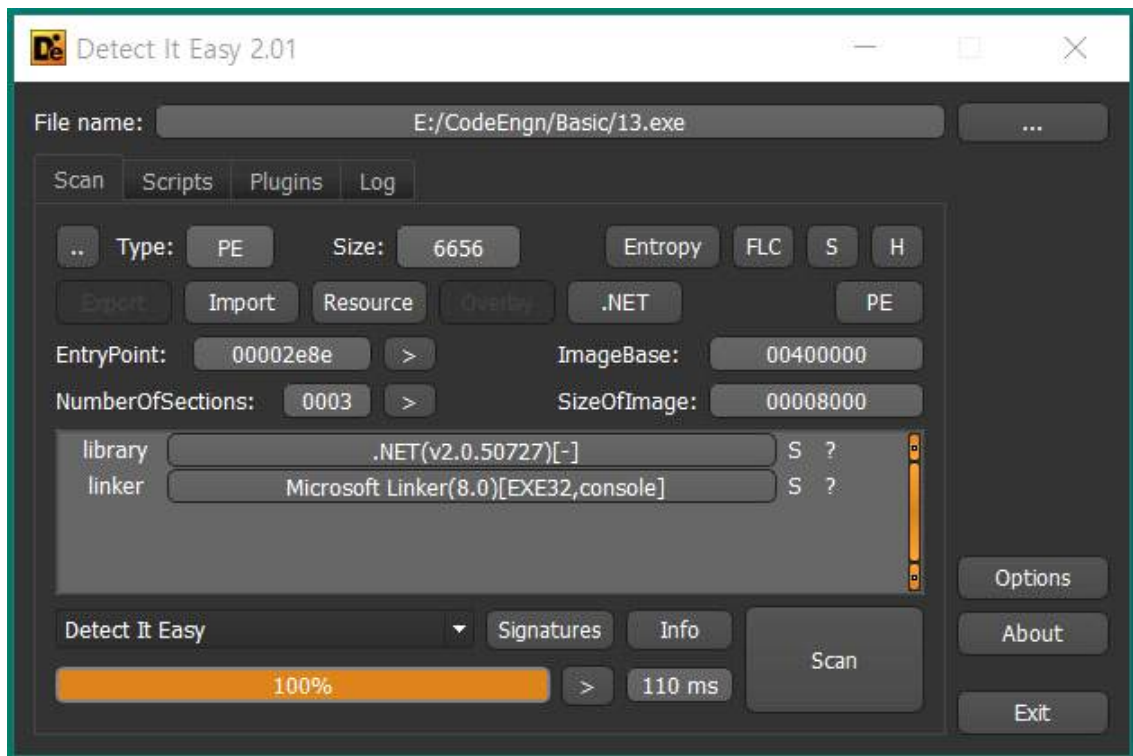
정답은 무엇인가

— Author: Basse 2002

— File Password: codeengn



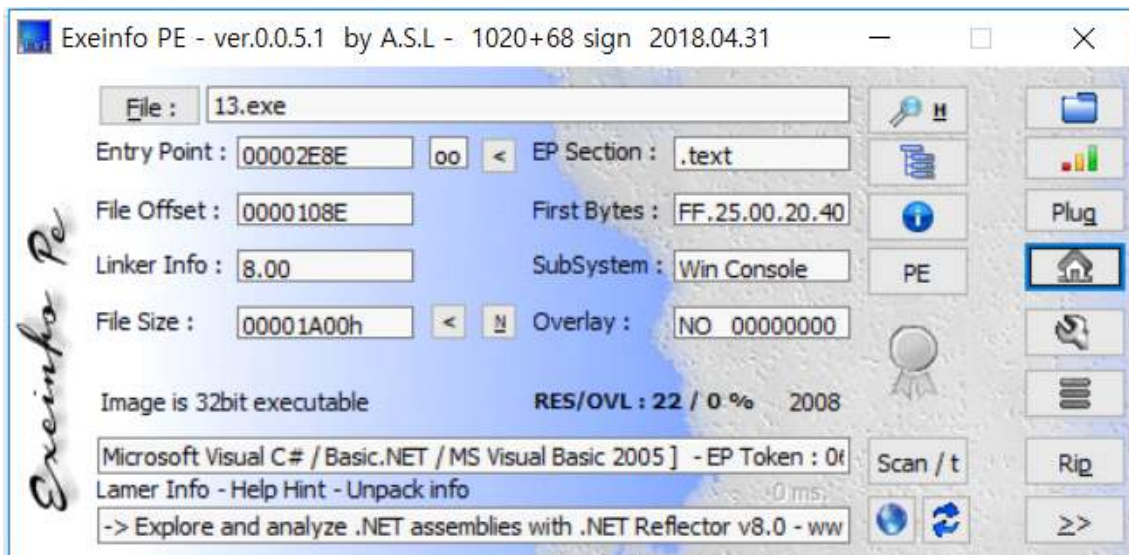
정답은 무엇인가. 심플하다.



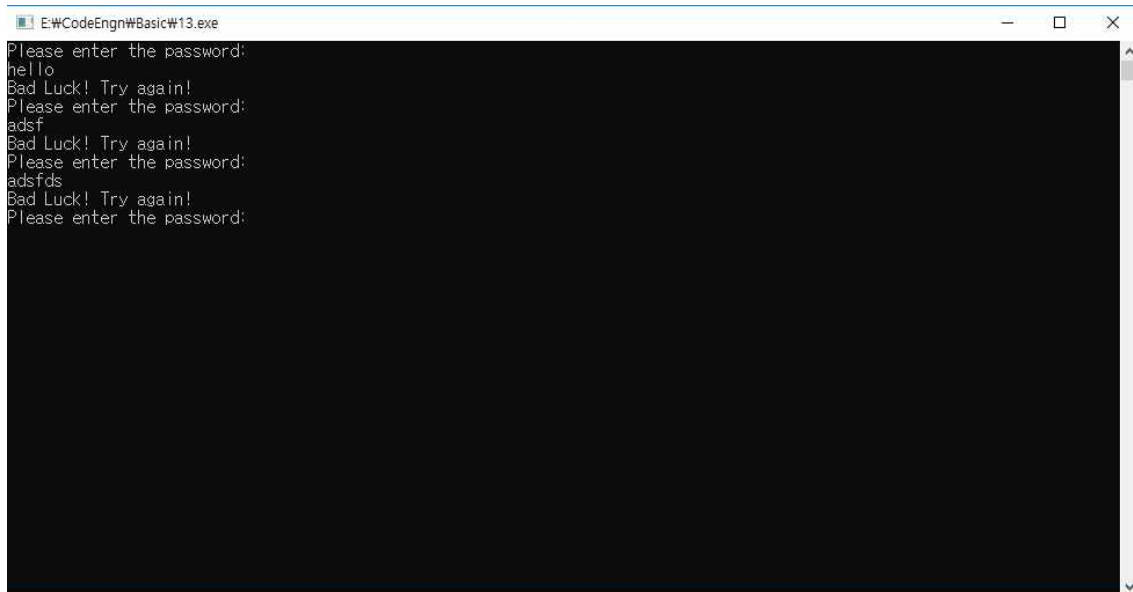
문제에서 처음으로 .NET library가 등장.

.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 읽어들이 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	
	0000000000A62E8E	<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.

```

00007FFF9034236 0F 83 C8 2A 98 00 jae mscorlib.ni.7FFF99E6D07
00007FFF903423C 56 88 5C 5A 10 mov bx,word ptr ds:[rdi+rbx*2+10]
00007FFF9034241 0F B7 C3 movzx eax,bx
00007FFF9034244 83 F8 0D cmp eax,D
00007FFF9034247 74 08 jne mscorlib.ni.7FFF9034251
00007FFF9034249 0F B7 C3 movzx eax,bx
00007FFF903424C 83 F8 0A cmp eax,A
00007FFF903424F 75 C4 jne mscorlib.ni.7FFF9034215
00007FFF9034251 48 85 F6 test rsi,rsi
00007FFF9034254 0F 85 BA 00 00 00 jne mscorlib.ni.7FFF9034314
00007FFF903425A 48 88 57 30 mov rdx,qword ptr ds:[rdi+30]
00007FFF9034262 44 88 47 40 mov r8d,dword ptr ds:[rdi+40]
00007FFF9034265
00007FFF9034267
00007FFF903426C
00007FFF903426F
00007FFF9034270
00007FFF9034274
00007FFF9034277
00007FFF903427C
00007FFF903427F
00007FFF9034282

```

Debugger window shows: Please enter the password: Hello

```

00007FFF9034277 83 F8 0D cmp eax,D
00007FFF903427A 75 36 jne mscorlib.ni.7FFF9034282
00007FFF903427C 88 4F 40 mov ecx,dword ptr ds:[rdi+40]
00007FFF903427F 88 47 44 mov eax,dword ptr ds:[rdi+44]
00007FFF9034282 38 C8 cmp ecx,eax
00007FFF9034284 0F 8D 46 2A 98 00 jae mscorlib.ni.7FFF99E6D00
00007FFF903428A 48 88 57 30 mov rdx,qword ptr ds:[rdi+30]
00007FFF903428E 88 4F 40 mov ecx,dword ptr ds:[rdi+40]
00007FFF9034291 4C 63 C1 movsxd r8,ecx
00007FFF9034294 48 88 42 08 mov rax,qword ptr ds:[rdx+8]
00007FFF9034298 4C 38 C0 cmp r8,rax
00007FFF903429B 0F 83 66 2A 98 00 jae mscorlib.ni.7FFF99E6D07
00007FFF90342A1 42 0F B7 44 42 10 movzx eax,word ptr ds:[rdx+r8*2+10]
00007FFF90342A7 83 F8 0A cmp eax,A
00007FFF90342AA 75 06 jne mscorlib.ni.7FFF9034282
00007FFF90342AC 8D 41 01 lea eax,qword ptr ds:[rcx+1]
00007FFF90342AF 89 47 40 mov dword ptr ds:[rdi+40],eax
00007FFF90342B2 48 88 C6 mov rax,rsi
00007FFF90342B5 48 83 C4 20 add rsp,20
00007FFF90342B9 41 5C pop r12
00007FFF90342BB 5F pop rdi
00007FFF90342BD 5E pop rsi
00007FFF90342BF 5D pop rbp
00007FFF90342C1 58 pop rbx
00007FFF90342C3 F3 C3 ret

```

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

그렇게 계속 트레이싱하다가

```

RIP 00007FFF9A7001DD 48 8B 09 mov rcx,qword ptr ds:[rcx]
00007FFF9A7001E2 E8 9E A0 8D 5E call mscorlib.ni.7FFF99F2700
00007FFF9A7001E7 E8 19 25 EF 5E call mscorlib.ni.7FFF99F2700

```

위 라인을 만났을 때 Bad Luck! Try again이 실행됨.

유심히 레지스터 변화를 관찰하다가.

```

RAX 0000000000000001
RBX 00000000030DEBD8
RCX 00000000030E09A8
RDX 0000000000000000
RBP 0000000003053360
RSP 0000000000CFFA20

```

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	.uy.....
00000000029CEBE8	4C 00 65 00 74 00 65 00	L.e.t.e.m.i.n.m.
00000000029CEBF8	61 00 6E 00 00 00 00 00	a.n.....
00000000029CEC08	D0 63 0E F9 FF 7F 00 00	D.c.uy.....
00000000029CEC18	00 00 00 00 00 00 00 00	.....
00000000029CEC28	00 00 00 00 00 00 00 00	.....uy..
00000000029CEC38	00 00 00 00 00 00 00 00	.....@
00000000029CEC48	C8 37 10 F9 FF 7F 00 00	E7.uy..x.....
00000000029CEC58	04 00 00 00 00 01 00 00	.....
00000000029CEC68	68 64 0E F9 FF 7F 00 00	hd.uy.....
00000000029CEC78	00 00 00 00 00 00 00 00	.....
00000000029CEC88	00 00 00 00 00 00 00 00	.....
00000000029CEC98	48 EC 9C 02 00 00 00 00	Hi.....
00000000029CECA8	18 96 0F F9 FF 7F 00 00	...uy.....
00000000029CECB8	00 00 00 00 00 00 00 00	.....uy..
00000000029CECC8	18 ED 9C 02 00 00 00 00	.i.....
00000000029CECD8	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	P.D.uy.....
0000000003331120	C8 0A 33 03 00 00 00 00	E.3.....
0000000003331130	28 9D 0F F9 FF 7F 00 00	(.uy.....
0000000003331140	62 00 79 00 74 00 65 00	b.y.t.e.C.o.u.n.
0000000003331150	74 00 00 00 00 00 00 00	t.....
0000000003331160	28 9D 0F F9 FF 7F 00 00	(.uy.....
0000000003331170	63 00 68 00 61 00 72 00	c.h.a.r.s.....
0000000003331180	00 00 00 00 00 00 00 00	.....(.uy
0000000003331190	06 00 00 00 05 00 00 00	.....h.e.l.l
00000000033311A0	6F 00 00 00 00 00 00 00	o.....
00000000033311B0	00 00 00 00 00 00 00 00	.....
00000000033311C0	00 00 00 00 00 00 00 00	.....
00000000033311D0	00 00 00 00 00 00 00 00	.....

결과적으로 입력해보니



```
E:\CodeEngn\BasicW13.exe
Please enter the password:
Late
Bad Luck! Try again!
Please enter the password:
Leteminiman
Bad Luck! Try again!
Please enter the password:
Leteminman
Well Done! You cracked it!
```

Leteminman

Clear