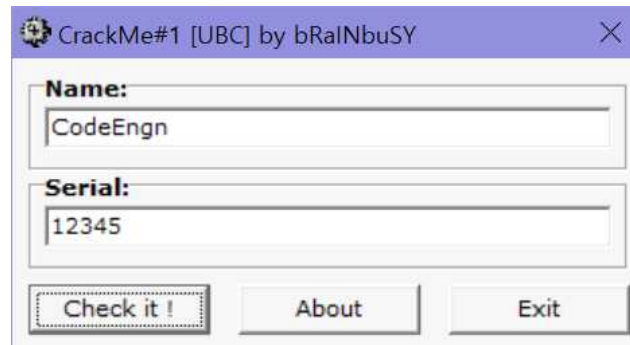


CodeEngn Basic RCE L15

: Name이 CodeEngn일때 Serial을 구하시오



CodeEngn일 때의 시리얼 넘버를 찾는 것이기 때문에 시리얼 넘버에 아무거나 넣어주었고, Try again!이라는 에러메세지가 났다. 올리디버거로 열어주었더니 따로 패킹이 되어 있지는 않아서, 오류 메세지를 찾기 위해 all text string을 찾아주었다.

Address	Disassembly	Text string
00401000	MOV EAX, 15.00401000	(Initial CPU selection)
00401001	MOV ECX, 15.00401001	ASCII "JPEG Image File"
00401002	MOV EDI, 15.00401002	ASCII "jpeg"
00401003	MOV EAX, 15.00401003	ASCII "JPEG Image File"
00401004	MOV EAX, 15.00401004	ASCII "jpg"
00401005	ASCII "JPEG Image File",0	
00401006	ASCII "jpeg",0	
00401007	ASCII "jpg",0	
00401008	DD 15.00401008	ASCII 06,"TForm2"
00401009	ASCII "Panel1"	
0040100A	ASCII "Label1"	
0040100B	ASCII "Label2"	
0040100C	ASCII "Memo1"	
0040100D	ASCII "Panel2"	
0040100E	ASCII "Image1"	
0040100F	ASCII "TForm2"	
00401010	ASCII "TForm2"	
00401011	ASCII "Unit2"	
00401012	DD 15.00401012	ASCII 06,"TForm1"
00401013	ASCII "GroupBox1"	
00401014	ASCII "GroupBox2"	
00401015	ASCII "Edit1"	
00401016	ASCII "Edit2"	
00401017	ASCII "Button1"	
00401018	ASCII "Button2"	
00401019	ASCII "Button3"	
0040101A	ASCII "Button1Click"	
0040101B	ASCII "Button3Click"	
0040101C	ASCII "Button2Click"	
0040101D	ASCII "TForm1"	
0040101E	ASCII "TForm1"	
0040101F	ASCII "Unit1"	
00401020	MOV EBX, 15.00401020	ASCII ""a"
00401021	MOV EAX, 15.00401021	ASCII "You cracked the UBC CrackMe#1 ! Please ser
00401022	MOV EDI, 15.00401022	ASCII "CRACKED"
00401023	MOV EAX, 15.00401023	ASCII "Try Again !"
00401024	ASCII "You cracked the "	
00401025	ASCII "UBC CrackMe#1 ! "	
00401026	ASCII "Please send your"	
00401027	ASCII " solution to ubc"	
00401028	ASCII "racker@hotmail."	
00401029	ASCII "com !",0	
0040102A	ASCII "CRACKED",0	
0040102B	ASCII "Try Again !"	

you cracked the UBC ~~~ 는 성공메세지로 추측된다. 해당 주소로 이동한다.

00458816	. E8 45FFFFFF	CALL 15.00458760	
00458818	. 8055 FC	LEA EDX,DWORD PTR SS:[EBP-4]	
0045881E	. 8B83 00020000	MOV EAX,DWORD PTR DS:[EBX+200]	
00458824	. E8 97C0FCFF	CALL 15.004255C0	
00458829	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	
0045882C	. E8 43EFAFF	CALL 15.00407774	
00458831	. 3B05 44B84500	CMP EAX,DWORD PTR DS:[45B844]	
00458837	. 75 1B	JNZ SHORT 15.00458854	
00458839	. B8 88884500	MOV EAX,15.00458888	ASCII "You cracked the UBC CrackMe#1 ! Please sen
0045883E	. E8 29C1FEFF	CALL 15.0044496C	ASCII "CRACKED"
00458843	. BA E8884500	MOV EDX,15.004588E8	
00458848	. A1 3CB84500	MOV EAX,DWORD PTR DS:[45B83C]	
0045884D	. E8 9EC0FCFF	CALL 15.004255F0	
00458852	. EB 0A	JMP SHORT 15.0045885E	
00458854	. B8 F8884500	MOV EAX,15.004588F8	ASCII "Try Again !"
00458859	. E8 0EC1FEFF	CALL 15.0044496C	
0045885E	. 33C0	XOR EAX,EAX	
00458860	. 5A	POP EDX	
00458861	. 59	POP ECX	
00458862	. 59	POP ECX	
00458863	. 64:8910	MOV DWORD PTR FS:[EAX],EDX	
00458866	. 68 7B884500	PUSH 15.0045887B	
0045886B	. 8D45 FC	LEA EAX,DWORD PTR SS:[EBP-4]	
0045886F	. E8 7E0CFCFF	CALL 15.00407774	

성공메세지 위쪽을 보면

CMP EAX, PTR ~~

JNZ 00458854가 나오는데, 14번 문제와 마찬가지로 EAX와 PTR ~~~를 비교해서 0이 아니라면 00458854로 이동해서 TRY AGAIN이라는 에러 메시지를 출력한다.

즉 EAX와 PTR ~~~의 값이 같아야 한다. 따라서 CMP에 BP를 걸고 다시 실행해 주었다.

**** DWORD PTR DS:[45B844] ****

45B844 주소의 데이터 영역의 값->해당 줄 선택하여 FOLLOW IN DUMP를 수행해주어야함.

Registers (FPU)	
EAX	00003039
ECX	0019F058
EDX	00000000
EBX	02204B80
ESP	0019F060
EBP	0019F074
ESI	0019F1EC
EDI	0019F1EC
EIP	00458831 15.00458831
C 0	ES 002B 32bit 0(FFFFFFFF)
P 1	CS 0023 32bit 0(FFFFFFFF)
A 0	SS 002B 32bit 0(FFFFFFFF)

EAX의 값을 10진수로 바꾸어주면 12345이고, EAX 레지스터가 사용자의 입력값을 저장한다는 것을 알 수 있다. 그럼 앞서 말한것처럼 덤프따라가기로 밑 헥스 데이터를 확인해 보았다.

Address	Hex dump	ASCII
00458844	60 61 00 00 09 00 00 00 FF FF FF FF 00 00 00 00	'a'.....
00458854	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00458864	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00458874	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00458884	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00458894	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004588A4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

DWORD PTR DS:[45B844]는 45B844부터 4바이트 만큼의 값이기 때문에 00006160이다. (리틀 엔디안이라서 반대로 보아야함) 마찬가지로 10진수로 바꾸어 주면 24928이고, 이를 대입해보면



정답이다!!