

Code Engn Basic 15

4.Z320

eltzero@gmail.com

Challenges : Basic 15

Author : uBc - bRaINbuSY

Korea :

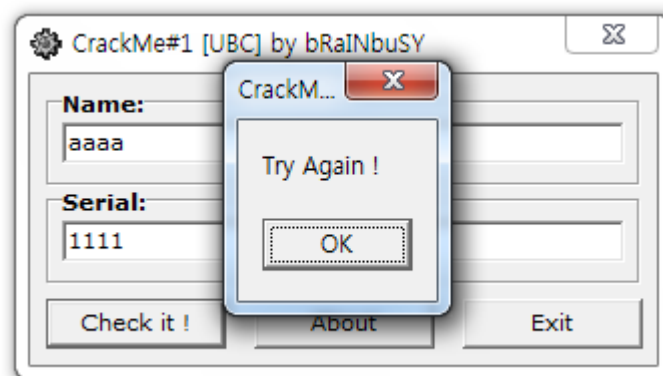
Name이 CodeEngn일때 Serial을 구하시오

English :

Find the Serial when the Name is CodeEngn

[Download](#)

Name 에 따른 Serial 을 구하는 문제입니다.



아무 값이나 넣고 'Check it!'을 누르면 'Try Again!'이란 메시지박스가 나옵니다.

OllyDbg 에서 이 문자열을 따라가게 되면

0045882C	. E8 43EFAFF	CALL 15.00407774	
00458831	. 3B05 44B8450	CMP EAX,DWORD PTR DS:[45B844]	
00458837	. 75 1B	JNZ SHORT 15.00458854	
00458839	. B8 88884500	MOV EAX,15.00458888	ASCII "You cracked the
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 9ECDFCFF	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	. 5A	POP EAX	

이런 부분을 찾을 수가 있으며 JNZ 에서 성공과 실패가 나뉘게 될 것임을 짐작할 수 있습니다.

```

00458816 . E8 45FFFFFF CALL 15.00458760
0045881B . 8D55 FC LEA EDX,DWORD PTR SS:[EBP-4]
0045881E . 8B83 D0020000 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

```

JNZ 구문 위에는 총 3 개의 함수가 있습니다. 그중에 첫번째 함수로 들어가게 되면 아래와 같은 내용이 나옵니다.

```

00458768 . BB 44B84500 MOV EBX,15.00458844
0045876D . BE 48B84500 MOV ESI,15.00458848
00458772 . BF 40B84500 MOV EDI,15.00458840

```

우선 3 개의 변수에 EBX, ESI, EDI 레지스터를 할당합니다.

```

004587A8 . E8 B7B2FAFF CALL 15.00403A64
004587AD . 85C0 TEST EAX,EAX
004587AF . 7E 19 JLE SHORT 15.004587CA
004587B1 . C706 01000000 MOV DWORD PTR DS:[ESI],1
004587B7 . 8B17 MOV EDX,DWORD PTR DS:[EDI]
004587B9 . 8B0E MOV ECX,DWORD PTR DS:[ESI]
004587BB . 0FB6540A FF MOVZX EDX,BYTE PTR DS:[EDX+ECX-1]
004587C0 . C1E2 03 SHL EDX,3
004587C3 . 0113 ADD DWORD PTR DS:[EBX],EDX
004587C5 . FF06 INC DWORD PTR DS:[ESI]
004587C7 . 48 DEC EAX
004587C8 . 75 ED JNZ SHORT 15.004587B7
004587CA . 8B07 MOV EAX,DWORD PTR DS:[EDI]

```

그후 Name 값만큼 루프를 돌게 되는데, EDX 엔 NAME, ESI 에는 1 을 할당하게 되고 EDX 에

Name 값에서 첫번째 글자를 가져오고 이를 SHL 3 실시한 뒤에 EBX 에 저장합니다.

이렇게 각 Name 의 한글자 한글자가 Shift 연산된 값이 EBX 에 저장되고 나면 루프가 끝나게 되고 Name 값은 EAX 에 들어갑니다.

```

004587C8 . 75 ED JNZ SHORT 15.004587B7
004587CA . 8B07 MOV EAX,DWORD PTR DS:[EDI]
004587CC . E8 93B2FAFF CALL 15.00403A64
004587D1 . C1E0 03 SHL EAX,3
004587D4 . 0103 ADD DWORD PTR DS:[EBX],EAX
004587D6 . 8B03 MOV EAX,DWORD PTR DS:[EBX]
004587D8 . C1E0 02 SHL EAX,2
004587DB . 8903 MOV DWORD PTR DS:[EBX],EAX
004587DD . 33C0 XOR EAX,EAX
004587DF . 5A POP EDX
004587E0 . 59 POP ECX
004587E1 . 59 POP ECX
004587E2 . 64:8910 MOV DWORD PTR FS:[EAX],EDX
004587E5 . 68 FA874500 PUSH 15.004587FA
004587EA . 8D45 FC LEA EAX,DWORD PTR SS:[EBP-4]
004587ED . E8 F6AFAFF CALL 15.004037E8
004587F2 . C3 RETN

```

그리고 다시 문자열 길이를 구하여 SHL 3 을 실시한 뒤 EBX 에 넣고 그 값을 SHL 2 회 실시하며 이 함수는 종료됩니다.

정리하자면 Name 의 각 글자를 각각 8 씩 곱한 뒤에 더하고, 이에 문자열 길이에 8 을 곱한 값을 더한 뒤 총 결과값에다가 4 를 곱한다고 보면 됩니다.

(Shift 연산이긴 하지만 문자 하나하나만 보기에 DWORD 범위를 넘어갈리는 없습니다.)

그리고 마지막 함수에 들어가게 되면

0040778B	64:FF30	PUSH DWORD PTR FS:[EAX]	
0040778E	64:8920	MOV DWORD PTR FS:[EAX],ESP	
00407791	8D55 FC	LEA EDX,DWORD PTR SS:[EBP-4]	
00407794	8BC3	MOV EAX,EBX	
00407796	E8 09B1FFFF	CALL 15.00402974	
0040779B	8BF0	MOV ESI,EAX	
0040779D	837D FC 00	CMP DWORD PTR SS:[EBP-4],0	
004077A1	74 23	JE SHORT 15.004077C6	
004077A3	8D55 F8	LEA EDX,DWORD PTR SS:[EBP-8]	
004077A6	A1 90A44500	MOV EAX,DWORD PTR DS:[45A490]	

EAX 에 사용자가 입력한 Serial 값을 넣고 함수를 호출하여 String 을 Integer 값으로 바꿔줍니다.

그리고 이 값을 ESI 에 저장하게 되고

004077DB	C3	RET	
004077DC	E9 C7BAFFFF	JMP 15.004032A8	
004077E1	EB F0	JMP SHORT 15.004077D3	
004077E3	8BC6	MOV EAX,ESI	
004077E5	5E	POP ESI	0012F504
004077E6	5B	POP EBX	
004077E7	8BE5	MOV ESP,EBP	
004077E9	5D	POP EBP	
004077EA	C3	RET	

함수의 마지막에 EAX 에 넣고 리턴하게 됩니다.

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 L
0045883E	E8 29C1FEFF	CALL 15.0044496C	ASCII "CRACKED"
00458843	B8 E8884500	MOV EDX,15.004588E8	
00458848	A1 3CB84500	MOV EAX,DWORD PTR DS:[45B83C]	
0045884D	E8 9ECDFCFF	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	

이렇게 리턴된 EAX 와 변조된 Name 과 비교하게 되므로 즉 Serial 은 숫자로 변경되기만 한다는 뜻이 됩니다.

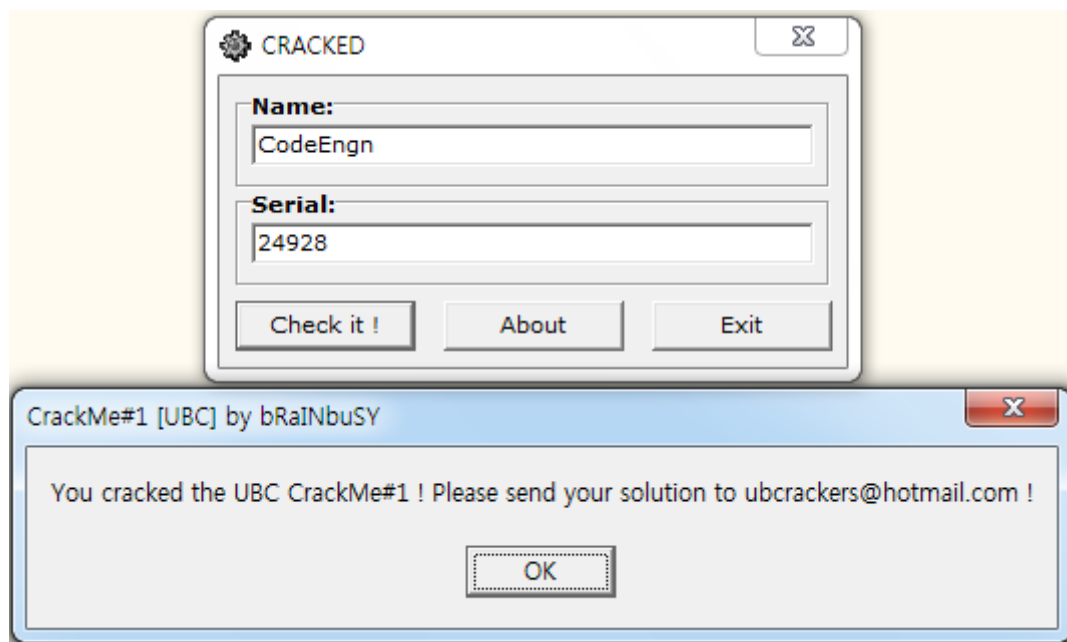
Address	Hex dump	ASCII
0045B844	00 31 00 00 05 00 00 00 FF FF FF FF 00 00 00 00	.1..+...
0045B854	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B864	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B874	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B884	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B894	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8A4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8B4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8C4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8D4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8E4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

그러므로 입력한 aaaa 에 맞는 Serial 은 변조된 Name 값이 저장되는 0x0045B844 에 저장되어 있는 0x3100 이므로 이를 Decimal 값으로 바꾼 12544 가 됩니다.

이번 Code Engn 의 문제는 CodeEngn 에 맞는 Serial 값을 찾는 것이었으므로 이 값을 넣은 뒤 프로그램을 따라가다 보면

Address	Hex dump	ASCII
0045B844	60 61 00 00 09 00 00 00 FF FF FF FF 00 00 00 00	a.....
0045B854	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B864	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B874	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B884	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B894	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8A4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8B4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8C4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8D4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0045B8E4	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

변조된 값에 0x6160 이 저장되어 있음을 확인할 수 있고 이를 입력하게 되면



이렇게 맞는 값이라고 나오게 됩니다.

즉 이번 코드엔진의 답은 24928 입니다.