2013

Code Engn Basic 14

4.Z320

elttzero@gmail.com

Challenges : Basic 14

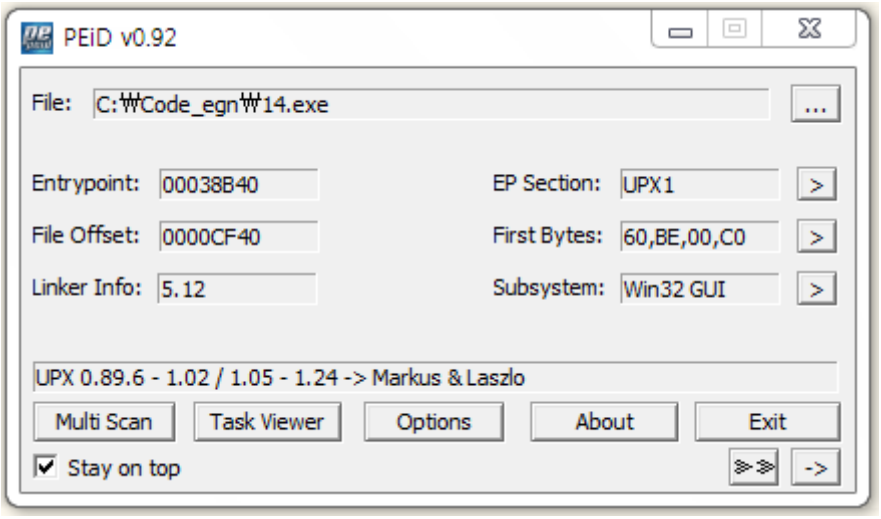
Author : BENGALY

Korea :
Name이 CodeEngn 일때 Serial을 구하시오
(이 문제는 정답이 여러개 나올 수 있는 문제이며 5개의 숫자로 되어있는 정답을 찾아야함, bruteforce 필요)
Ex) 11111

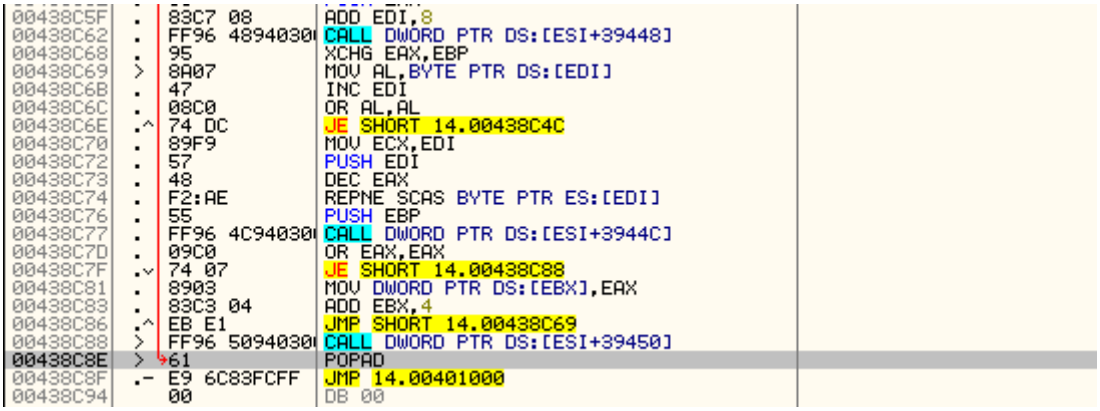
English :
Find the Serial when the Name of CodeEngn
(This problem has several answers, and the answer should be a 5 digit number. Brute forcing is required.)
Ex) 11111

Name 값이 주어졌을때 Serial 을 구하는 문제입니다.

Name 과 Serial 을 받아들인 뒤 Check 를 누르면 확인하고 MessageBox 를 띄우는 프로그램입니다.

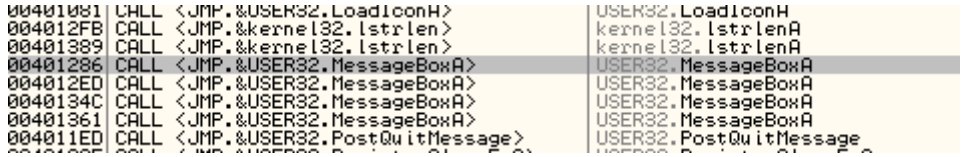


UPX 로 패킹되어 있으므로 Unpack 을 실시합니다.



처음 Entry Point 에서 따라가다 보면 스택값을 복구시키고 Jmp 하는 부분을 찾을 수 있습니다.

점프문으로 넘어간 부분이 OEP 이므로 Dump 를 실시합니다.



Check 버튼을 눌렀을 때 MessageBox 가 나타났으므로 함수를 따라가 보면

```
0040133C  75 15        JNZ SHORT 14__.00401353
0040133E  6A 00        PUSH 0
00401340  68 62344000  PUSH 14__.00403462          ASCII "Key/CrackMe #2   "
00401345  68 B8344000  PUSH 14__.004034B8          ASCII " Good Job, I Wish You the Very Best"
0040134A  6A 00        PUSH 0
0040134C  E8 9D000000  CALL <JMP.&USER32.MessageBoxA>
00401351  EB 13        JMP SHORT 14__.00401366
00401353  6A 00        PUSH 0
00401355  68 62344000  PUSH 14__.00403462          ASCII "Key/CrackMe #2   "
0040135A  68 86344000  PUSH 14__.00403486          ASCII " You Have Enter A Wrong Serial, Pleas
0040135F  6A 00        PUSH 0
00401361  E8 88000000  CALL <JMP.&USER32.MessageBoxA>
00401366  EB 15        JMP SHORT 14__.0040137D
```

이렇게 성공했을때, 실패했을때의 문자열 값이 보이는 것을 알 수 있습니다. 그러므로 이 근처에서 성공과 실패를 나누는 Jmp 문이 있을 것이라고 추측할 수 있으며 0x0040133C 에 있는 JNZ 임을 알 수 있습니다.

```
004012F3  C2 1000        RETN 10
004012F6  68 38304000    PUSH 14__.00403038         ASCII "aaaa"
004012FB  E8 30010000    CALL <JMP.&kernel32.lstrlen>
00401300  33F6           XOR ESI,ESI
00401302  8BC8           MOV ECX,EAX
00401304  B8 01000000    MOV EAX,1
00401309  8B15 38304000  MOV EDX,DWORD PTR DS:[403038]
0040130F  8A90 37304000  MOV DL,BYTE PTR DS:[EAX+403037]
00401315  81E2 FF000000  AND EDX,0FF
0040131B  8BDA           MOV EBX,EDX
0040131D  0FAFDA         IMUL EBX,EDX
00401320  03F3           ADD ESI,EBX
00401322  8BDA           MOV EBX,EDX
00401324  D1FB           SAR EBX,1
00401326  03F3           ADD ESI,EBX
00401328  2BF2           SUB ESI,EDX
0040132A  40             INC EAX
0040132B  49             DEC ECX
0040132C  75 DB          JNZ SHORT 14__.00401309
0040132E  56             PUSH ESI
0040132F  68 38314000    PUSH 14__.00403138         ASCII "1234"
00401334  E8 4A000000    CALL 14__.00401383
00401339  5E             POP ESI
0040133A  3BC6           CMP EAX,ESI
```

그 위쪽을 보게되면 Name 값인 aaaa 가 0x00403038 에, Serial 인 1234 가 0x00401309 에 저장되는것을 볼 수 있습니다. 그리고 우선 Name 값으로 여러가지 연산을 하는것을 확인할 수 있습니다.

```
00401383  55             PUSH EBP
00401384  8BEC           MOV EBP,ESP
00401386  FF75 08        PUSH DWORD PTR SS:[EBP+8]
00401389  E8 A2000000    CALL <JMP.&kernel32.lstrlen>
0040138E  53             PUSH EBX
0040138F  33DB           XOR EBX,EBX
00401391  8BC8           MOV ECX,EAX
00401393  8B75 08        MOV ESI,DWORD PTR SS:[EBP+8]
00401396  51             PUSH ECX
00401397  33C0           XOR EAX,EAX
00401399  AC             LODS BYTE PTR DS:[ESI]
0040139A  83E8 30        SUB EAX,30
0040139D  49             DEC ECX
0040139E  74 05          JE SHORT 14__.004013A5
004013A0  6BC0 0A        IMUL EAX,EAX,0A
004013A3  E2 FB          LOOPD SHORT 14__.004013A0
004013A5  03D8           ADD EBX,EAX
004013A7  59             POP ECX
004013A8  E2 EC          LOOPD SHORT 14__.00401396
004013AA  8BC3           MOV EAX,EBX
004013AC  5B             POP EBX
004013AD  C9             LEAVE
004013AE  C2 0400        RETN 4
```

아래 Serial 값을 보면 PUSH 후 이 함수를 CALL 하며 여기서도 LOOP 를 하며 Serial 값으로 연산을 하는것을 확인할 수 있습니다.

```
EAX 000004D2
ECX 00000000
EDX 00403139 ASCII "234"
EBX 00000030
ESP 0012FBEC
EBP 0012FBEC
ESI 00009240
EDI 0012FC68
```

각 두 연산이 끝난 뒤의 레지스터입니다. Name 의 연산은 ESI 에, Serial 의 연산은 EAX 에 저장되며 각 값은 레지스터에 나와있는 그대로입니다. 하지만 여기서 EAX 값을 보면 0x4D2 즉 10 진수로 1234 라는것을 알 수 있고 몇차례 값을 바꿔가며 확인해보면 입력한 값이 그대로 레지스터에 들어가는 것을 알 수 있습니다.

즉 Serial 값은 바뀐 Name 과 같은 것입니다.

```
EBP 0012FBEC
ESI 000129A1
EDI 0012FC68
```

여기서 이번 문제인 CodeEngn 을 입력하게 되면 ESI 값이 0x129A1 즉 76193 이 되는것을 볼 수 있고 이를 Serial 로 입력하게 되면

이렇게 인증되는 것을 확인할 수 있습니다.