## codeengn-basic-L11 풀이

리버싱 문제풀이 / Wonlf / 2022. 4. 2. 22:32



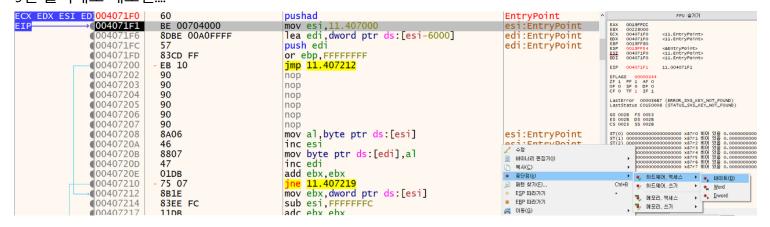
이번 문제는 9번 문제와 완벽히 동일하다. 먼저 9번 링크를 참고하겠다.

2022.03.27 - [리버싱 문제 풀이/CodeEngn.com] - codeengn-basic-L09 풀이

9번과 동일하지만 포인트로 짚고 넘어가야 할 것이 한 개 추가 되었다. 내려가면서 확인해보도록 하자.

문제는 OEP와 Stolenbyte를 원하고 있다.

9번 풀이대로 해보면...



pushad이후 하드웨어 브레이크 포인트를 걸어주고,

popad에서 멈추고 보게되면...

```
0040736D0040736E00407370004073750040737A
                                                     popad
                                                     push 0
                    68 00204000
                                                     push 11.402000
push 11.402012
                                                                                                                402000:"abex' 3rd crackme"
402012:"Click OK to check for the keyfile.'
                     68
                        12204000
                     8D4424 80
                                                     lea eax,dword ptr ss:[esp-80]
  0040737E
                     6A 00
                                                     push 0
                                                     cmp esp,eax
jne 11.40737E
00407380
                     39C4
                     75 FA
                                                     sub esp,FFFFFF80
jmp 11.40100C
                    E9 809CFFFF
```

popad이후 stolenbyte를 복구하는 구문과 OEP로 추정되는 것이 보인다.

그래서 Auth에 "40100C6A0068002040006812204000"

이대로 입력해보니 incorrect가 뜨길래 맞는데 뭐가 문제지 했는데

지금 저 40100C로 보이는 부분이 OEP로 보일 수도 있겠지만, stolenbyte바이트를 복구하지 않은 상태의 주소이다.

그러니까 40100C위에 stolenbyte가 복구 되었을 때의 EP가 OEP가 되는 것이다.

현재 확인한 stolenbyte는 "6A0068002040006812204000" 로 총 12바이트이다.

```
■00401000 |r.
                90
                                                                                         sub_401000
                                          nop
 00401001
                90
                                          nop
 00401002
                90
                                          nop
 00401003
                90
                                          nop
  중단점 설정되지 않음
                90
                                          nop
                90
                                          nop
                90
 00401006
                                          nop
                90
 00401007
                                          nop
                90
 00401008
                                          nop
                90
 00401009
                                          nop
                90
 0040100A
                                          nop
                90
 0040100B
                                          nop
■ 0040100c
                6A
                   00
                                          push
```

nop로 되어 있는 부분도 총 12바이트. 그러니 stolenbyte를 복구 해줬을 때를 보면

딱 맞아 떨어지고 00401000이 EP가 되었다.

당은 "004010006A0068002040006812204000"

더 쉽게 보면 0040100C에 stolenbyte(12byte)를 뺐을 때 실제 OEP가 나오게 된다.

>>> hex(0x0040100C - 12)
'0x401000'

이번 문제는 stolenbyte가 있을 때 stolenbyte를 생각해서 OEP를 찾아야 되는 것을 알려주는 문제였다.