CodeEngn Basic RCE

9. Level 09

Basic RCE L09

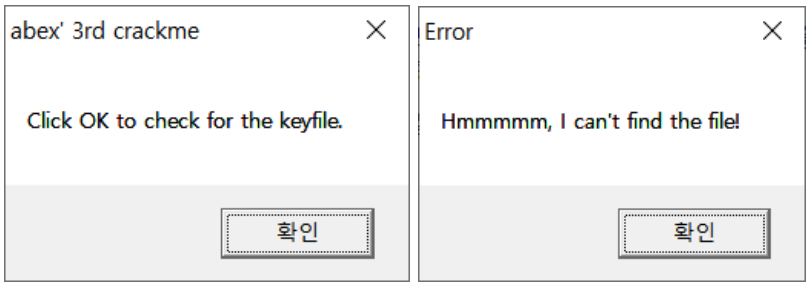
StolenByte를 구하시오 Ex) 75156A0068352040

— Author: abex
— File Password: codeengn

StolenByte란? : 패커가 이동시킨 코드의 윗부분 (보통 OEP에서부터 몇 개의 명령어) 이다.

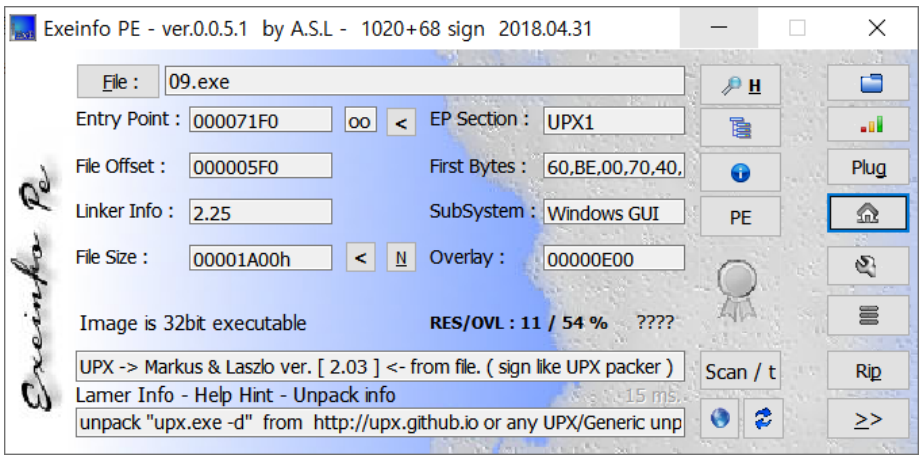
이 경우 언패킹을 할 시에도 정상 실행이 되지 않는다. 하지만 숨겨진 코드를 다시 제자리에 가져다놓고 덤프, IAT 복구를 하면 정상실행이 된다.

UPX에서는 마지막 JMP 전 POPAD 이후 일정 바이트의 코드를 의미함.



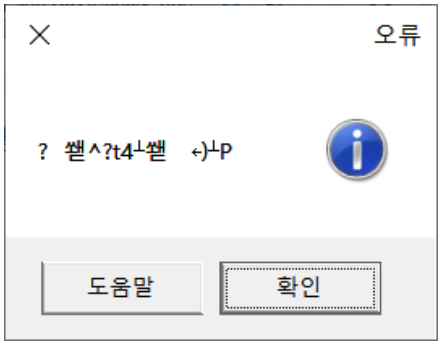
프로그램 실행 화면이다.

exeinfo로 패킹 여부를 확인해보자.



UPX 패킹이 되어있는 것을 확인할 수 있다.

UPX 언패킹 후 x64dbg로 프로그램을 실행해보자.



프로그램 실행 시 오류가 난다.

x64dbg로 프로그램을 까보자.

```
00401000    90          nop                     EntryPoint
00401001    90          nop
00401002    90          nop
00401003    90          nop
00401004    90          nop
00401005    90          nop
00401006    90          nop
00401007    90          nop
00401008    90          nop
00401009    90          nop
0040100A    90          nop
0040100B    90          nop
0040100C    6A 00       push 0
0040100E    E8 8C000000 call 09.40109F
```

OEP 부분(EntryPoint)이 nop으로 채워진 것이 보인다.

언패킹 전 프로그램을 x64dbg로 깐 후

```
0040736E    6A 00       push 0
00407370    68 00204000 push 09 언패킹 전.402000
00407375    68 12204000 push 09 언패킹 전.402012
0040737A    8D4424 80   lea eax,dword ptr ss:[esp-80]
0040737E    6A 00       push 0
```

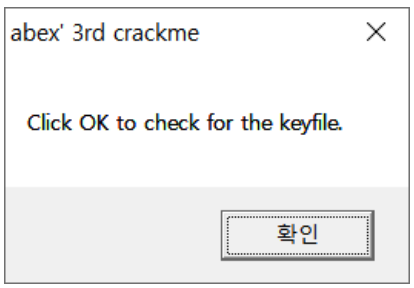
push 값들을 확인해보니 추가로 push해주는 코드가 보인다.

문자열 확인을 위해 push 0까지 실행해보니

```
0040736E    6A 00       push 0
00407370    68 00204000 push 09 언패킹 전.402000    402000:"abex' 3rd crackme"
00407375    68 12204000 push 09 언패킹 전.402012    402012:"Click OK to check for the keyfile."
0040737A    8D4424 80   lea eax,dword ptr ss:[esp-80]
0040737E    6A 00       push 0
```

"abex' 3rd crackme"

"Click OK to check for the keyfile."



프로그램 실행 시 나오는 문자열 확인 가능. 즉, StolenByte는 이 부분인 것을 알 수 있다.

StolenByte는 6A0068002040006812204000