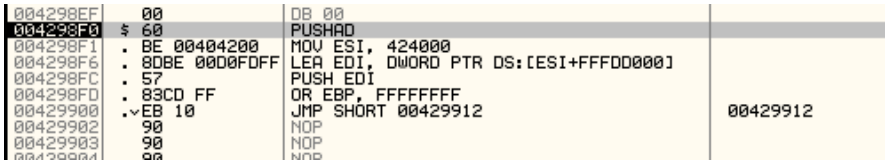
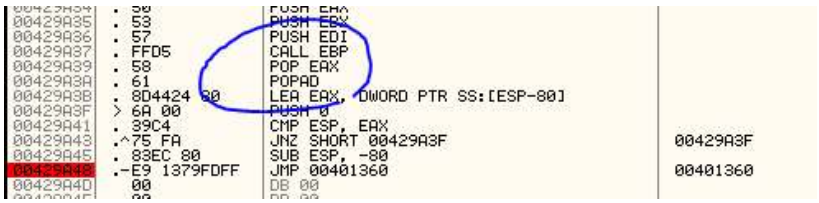
문제 : Unpack을 한 후 Serial을 찾으시오.
정답인증은 OEP + Serial Ex) 00400000PASSWORD

STUD_PE로 열어보니까 UPX 패킹 되어있음.
UPX -> PUSHAD -> POPAD 찾자.

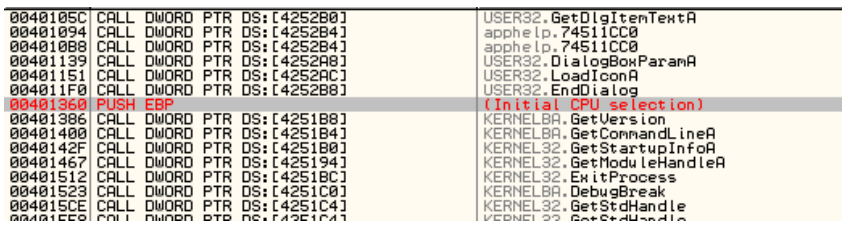


일단 여기가 PUSHAD 부분. 저기 아래로 내려볼 필요가 있다.



POPAD 찾음. 그다음에 나오는 JMP [addr]에서 addr가 OEP. -> 00401360 (JMP 00401360)
저기에 BP걸고 F9 run함. 그리고 F7로 step into.

이제 시리얼 찾아야 됨. 그래서 해 본 방법이 2가지.
1. 시리얼 틀리면 Wrong serial!!!출력 -> Search for -> All referenced text strings에서 저 문장 찾기
근데 1번은 나오지 않았다... 못 찾은 것 일수도 있음.
그래서 2번 방법 실행
2. Search for -> All intermodular calls 로 문자열 비교 함수라던가 도움될 만한 함수 찾기.



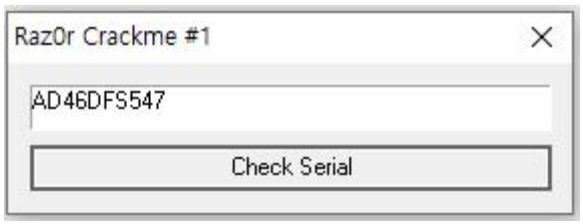
뭔가 엄청 많은데, 딱 눈에 띄는게 있다. Abex Crackme #1에서 본 GetDlgItemTextA() 함수.
이 함수는 다이얼로그 박스 안에 있는 문자열을 가져오는 함수.
(Initial CPU selection) -> 이거는 이제 실행할 위치를 말함(00401360)

저 GetDlgItemTextA()를 더블클릭해서 해당 위치로 이동하자.

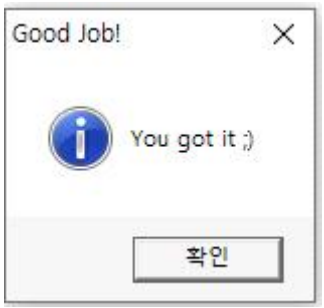
```
0040105C  FF15 B0524200   CALL DWORD PTR DS:[4252B0]      USER32.GetDlgItemTextA
00401062  3BF4            CMP ESI, ESP
00401064  E8 B7020000     CALL 00401320                   00401320
00401069  68 D4354200     PUSH 4235D4
0040106E  68 302A4200     PUSH 422A30                     ASCII "AD46DFS547"
00401073  E8 18020000     CALL 00401290                   00401290
00401078  83C4 08         ADD ESP, 8
0040107B  85C0            TEST EAX, EAX
0040107D  75 24           JNZ SHORT 004010A3              004010A3
0040107F  8BF4            MOV ESI, ESP
00401081  6A 40           PUSH 40
00401083  68 48004200     PUSH 420048                     ASCII "Good Job!"
00401088  68 38004200     PUSH 420038                     ASCII "You got it ;)"
0040108D  8B0D 38364200   MOV ECX, DWORD PTR DS:[423638]
00401093  51              PUSH ECX
00401094  FF15 B4524200   CALL DWORD PTR DS:[4252B4]      apphelp.74511CC0
0040109A  3BF4            CMP ESI, ESP
0040109C  E8 7F020000     CALL 00401320                   00401320
004010A1  EB 22           JMP SHORT 004010C5              004010C5
004010A3  8BF4            MOV ESI, ESP
004010A5  6A 10           PUSH 10
004010A7  68 30004200     PUSH 420030                     ASCII "ERROR"
004010AC  68 1C004200     PUSH 42001C                     ASCII "Wrong serial!!!"
004010B1  8B15 38364200   MOV EDX, DWORD PTR DS:[423638]
004010B7  52              PUSH EDX
004010B8  FF15 B4524200   CALL DWORD PTR DS:[4252B4]      apphelp.74511CC0
004010BE  3BF4            CMP ESI, ESP
004010C0  E8 5B020000     CALL 00401320                   00401320
004010C5  33C0            XOR EAX, EAX
004010C7  5F              POP EDI
004010C8  5E              POP ESI
004010C9  5B              POP EBX
004010CA  83C4 40         ADD ESP, 40
004010CD  3BEC            CMP EBP, ESP
004010CF  E8 4C020000     CALL 00401320                   00401320
004010D4  8BE5            MOV ESP, EBP
004010D6  5D              POP EBP
004010D7  C9              RETN
```

이동한 위치이다. 뭔가 굉장히 유용한 정보를 찾아낸 것 같다...
가장 먼저 눈에 띄는 것은 Good Job! 이거랑 Wrong serial!!! 이다.
그리고 두 번째로 눈에 띄는 것은 5번째 줄에 있는 ASCII “AD46DFS547” 이다.

진짜 딱 봐도 Serial Number같이 생겨서 바로 프로그램에 넣어보았다.



결과는?



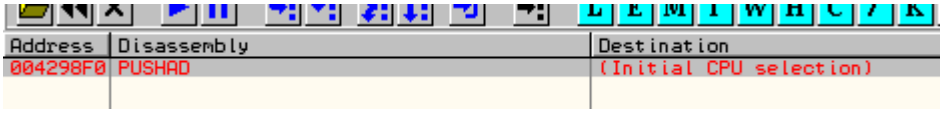
시리얼 번호도 찾았다. -> “AD46DFS547”

6번 문제의 답은 OEP + Serial Ex) 00400000PASSWORD 방식으로 작성해야 하므로

정답은 00401360AD46DFS547 일 것이다. 바로 Auth 넣자.

팁! 몰랐던 사실인데, UPX 패킹되어있는 파일은 무조건 OEP를 찾아야 Search for 라던가 뭔가를 해 볼수 있다. 만약 OEP가 아닌 PUSHAD ~ POPAD 부분에서 뭔가를 찾으려고 한다면... 아무것도 얻을 수 없다.

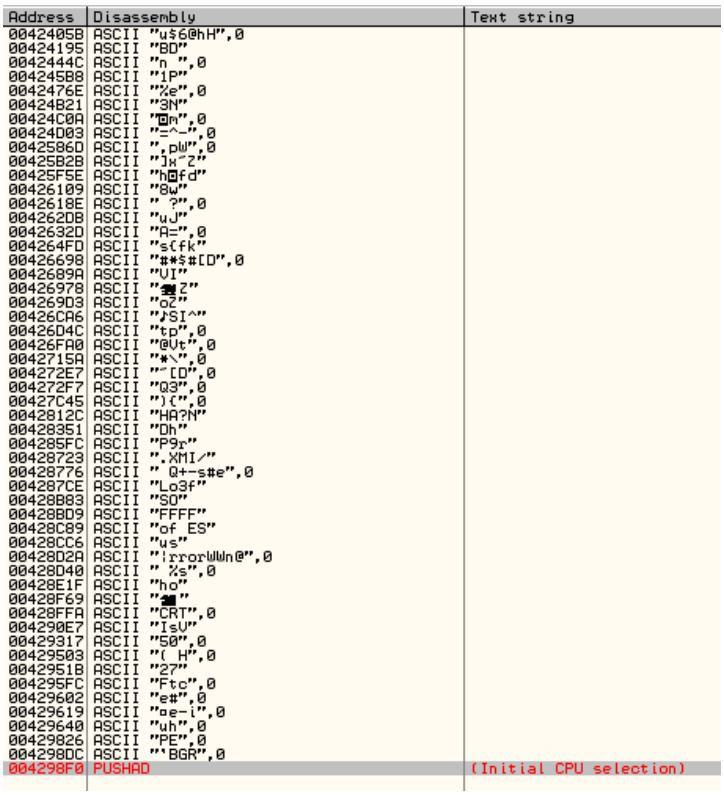
아래는 UPX 패킹된 상태에서 Search for All intermodular calls 본 결과.



Address	Disassembly	Destination
004298F0	PUSHAD	(Initial CPU selection)

이거 한 줄이 끝.

그리고 Search for All referenced text strings 본 결과.



Address	Disassembly	Text string
0042405B	ASCII "u$6@hH",0	
00424195	ASCII "BD"	
0042444C	ASCII "n ",0	
004245B8	ASCII "1P"	
0042476E	ASCII "%e",0	
00424B21	ASCII "3N"	
00424C0A	ASCII "m",0	
00424D03	ASCII "=^-",0	
0042586D	ASCII ",pW",0	
00425B2B	ASCII "]w~Z"	
00425F5E	ASCII "hfd"	
00426109	ASCII "8w"	
0042618E	ASCII " ?",0	
004262DB	ASCII "uJ"	
0042632D	ASCII "A=",0	
004264FD	ASCII "s{fk"	
00426698	ASCII "#*$#D",0	
0042689A	ASCII "VI"	
00426978	ASCII "Z"	
004269D3	ASCII "oZ"	
00426CA6	ASCII "SI^"	
00426D4C	ASCII "tp",0	
00426FA0	ASCII "@Vt",0	
0042715A	ASCII "*\",0	
004272E7	ASCII "~D",0	
004272F7	ASCII "Q3",0	
00427C45	ASCII "){",0	
0042812C	ASCII "HA?N"	
00428351	ASCII "Dh"	
004285FC	ASCII "P9r"	
00428723	ASCII ".XMI"	
00428776	ASCII " Q+~s#e",0	
004287CE	ASCII "Lo3f"	
00428B83	ASCII "SO"	
00428BD9	ASCII "FFFF"	
00428C89	ASCII "of ES"	
00428CC6	ASCII "us"	
00428D2A	ASCII "!rrorWWn@",0	
00428D40	ASCII " %s",0	
00428E1F	ASCII "ho"	
00428F69	ASCII ""	
00428FFA	ASCII "CRT",0	
004290E7	ASCII "IsU"	
00429317	ASCII "50",0	
00429503	ASCII "(H",0	
0042951B	ASCII "27"	
004295FC	ASCII "Ftc",0	
00429602	ASCII "e#",0	
00429619	ASCII "ae-i",0	
00429640	ASCII "uh",0	
00429826	ASCII "PE",0	
004298DC	ASCII "`BGR",0	
004298F0	PUSHAD	(Initial CPU selection)

역시 좋은 정보가 없다.

UPX는 어떤 일이 있어도 PUSHAD -> POPAD -> OEP 찾는 것으로 해야 하는구나!