

문제 : Unpack을 한 후 Serial을 찾으시오.

정답인증은 OEP + Serial Ex) 00400000PASSWORD

STUD\_PE로 열어보니까 UPX 패킹 되어있음.

UPX -> PUSHAD -> POPAD 찾자.

004298EF	00	DB 00	
004298F0	60	PUSHAD	
004298F1	BE 00404200	MOV ESI, 424000	
004298F6	8DBE 00D0F0FF	LEA EDI, DWORD PTR DS:[ESI+FFFFD000]	
004298FC	57	PUSH EDI	
004298FD	83CD FF	OR EBP, FFFFFFFF	
00429900	EB 10	JMP SHORT 00429912	00429912
00429902	90	NOP	
00429903	90	NOP	
00429904	90	NOP	

일단 여기가 PUSHAD 부분. 저기 아래로 내려볼 필요가 있다.

00429A34	50	PUSH EAX	
00429A35	53	PUSH EBX	
00429A36	57	PUSH EDI	
00429A37	FFD5	CALL EBP	
00429A38	58	POP EAX	
00429A39	61	POP EDI	
00429A3B	804424 80	LEA EAX, DWORD PTR SS:[ESP-80]	
00429A3F	6A 00	PUSH 0	
00429A41	39C4	CMP ESP, EAX	
00429A43	75 FA	JNZ SHORT 00429A3F	00429A3F
00429A45	83EC 80	SUB ESP, -80	
00429A48	E9 1379F0FF	JMP 00401360	00401360
00429A4D	00	DB 00	
00429A4F	00	DB 00	

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.

0040105C	CALL DWORD PTR DS:[4252B0]	USER32.GetDlgItemTextA
00401094	CALL DWORD PTR DS:[4252B4]	apphelp.74511CC0
004010B8	CALL DWORD PTR DS:[4252B4]	apphelp.74511CC0
00401139	CALL DWORD PTR DS:[4252B0]	USER32.DialogBoxParamA
00401151	CALL DWORD PTR DS:[4252AC]	USER32.LoadIconA
004011F0	CALL DWORD PTR DS:[4252B0]	USER32.EndDialog
00401360	PUSH EBP	(Initial CPU selection)
00401386	CALL DWORD PTR DS:[4251B0]	KERNELBA.GetVersion
00401400	CALL DWORD PTR DS:[4251B4]	KERNELBA.GetCommandLineA
0040142F	CALL DWORD PTR DS:[4251B0]	KERNEL32.GetStartupInfoA
00401467	CALL DWORD PTR DS:[425194]	KERNEL32.GetModuleHandleA
00401512	CALL DWORD PTR DS:[4251B0]	KERNEL32.ExitProcess
00401523	CALL DWORD PTR DS:[4251C0]	KERNELBA.DebugBreak
004015CE	CALL DWORD PTR DS:[4251C4]	KERNEL32.GetStdHandle
004015E8	CALL DWORD PTR DS:[4251C4]	KERNEL32.GetStdHandle

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.

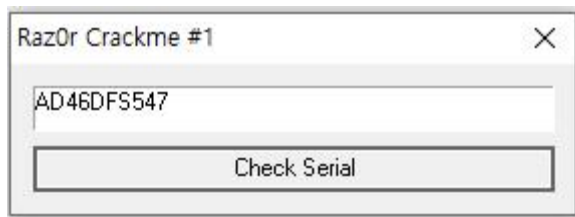
00401062	FF15 B0524200	CALL DWORD PTR DS:[4252B0]	USER32.GetDlgItemTextA
00401064	3BF4	CHP ESI, ESP	
00401066	E8 B7020000	CALL 00401320	00401320
00401068	68 D4354200	PUSH 4235D4	
0040106A	68 302A4200	PUSH 422A30	ASCII "AD46DFS547"
0040106C	E8 18020000	CALL 00401290	00401290
0040106E	83C4 08	ADD ESP, 8	
00401070	85C0	TEST EAX, EAX	
00401072	JNZ SHORT 004010A3		004010A3
00401074	75 24	MOV ESI, ESP	
00401076	8BF4	PUSH 40	
00401078	6A 40	PUSH 420048	ASCII "Good Job!"
0040107A	68 48004200	PUSH 420038	ASCII "You got it ;)"
0040107C	68 38004200	MOV ECX, DWORD PTR DS:[423638]	
0040107E	8B0D 38364200	PUSH ECX	apphelp.74511CC0
00401080	S1	CALL DWORD PTR DS:[4252B4]	
00401082	FF15 B4524200	CHP ESI, ESP	00401320
00401084	3BF4	CALL 00401320	004010C5
00401086	E8 7F020000	JMP SHORT 004010C5	
00401088	EB 22	MOV ESI, ESP	
0040108A	8BF4	PUSH 10	ASCII "ERROR"
0040108C	6A 10	PUSH 420030	ASCII "Wrong serial!!!!"
0040108E	68 30004200	PUSH 42001C	
00401090	68 1C004200	MOV EDX, DWORD PTR DS:[423638]	
00401092	8B15 38364200	PUSH EDX	apphelp.74511CC0
00401094	S2	CALL DWORD PTR DS:[4252B4]	
00401096	FF15 B4524200	CHP ESI, ESP	00401320
00401098	3BF4	CALL 00401320	
0040109A	E8 5B020000	XOR EAX, EAX	
0040109C	33C0	POP EDI	
0040109E	5F	POP ESI	
004010A0	5E	POP EBX	
004010A2	5B	ADD ESP, 40	
004010A4	83C4 40	CHP EBP, ESP	
004010A6	3BEC	CALL 00401320	00401320
004010A8	E8 4C020000	MOV ESP, EBP	
004010AA	8BE5	POP EBP	
004010AC	5D	RET	
004010AE	C3		

이동한 위치이다. 뭔가 굉장히 유용한 정보를 찾아낸 것 같다...

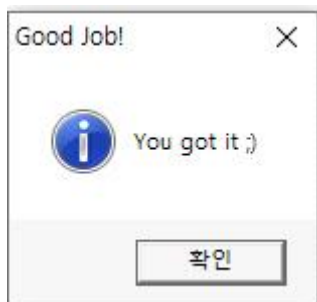
가장 먼저 눈에 띄는 것은 Good Job! 이거랑 Wrong serial!!! 이다.

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
00424058	ASCII "u\$6@hH",0	
00424195	ASCII "BD"	
0042444C	ASCII "n ",0	
00424588	ASCII "lp",0	
0042476E	ASCII "%e",0	
00424821	ASCII "3h",0	
00424C0A	ASCII "0h",0	
00424D03	ASCII "=^_",0	
00425860	ASCII ",plw",0	
00425828	ASCII "]H-2",0	
00425F5E	ASCII "h0fd"	
00426109	ASCII "8w",0	
0042618E	ASCII "?",0	
004262D8	ASCII "uJ",0	
0042632D	ASCII "A-",0	
004264FD	ASCII "s(fk"	
00426698	ASCII "##s#[D",0	
0042689A	ASCII "UI"	
00426978	ASCII "Z"	
004269D3	ASCII "o2"	
00426CA6	ASCII "J\$I^",0	
00426D4C	ASCII "tp",0	
00426FA0	ASCII "@Ut",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 ".XHI/",0	
00428776	ASCII "0+-s#e",0	
004287CE	ASCII "Lo3f"	
00428B83	ASCII "S0"	
00428BD9	ASCII "FFFF"	
00428CB9	ASCII "of ES"	
00428CC6	ASCII "us"	
00428D2A	ASCII "i:rror!l!n@",0	
00428D40	ASCII "%s",0	
00428E1F	ASCII "ho"	
00428F69	ASCII "0"	
00428FFA	ASCII "CRT",0	
004290E7	ASCII "IsU"	
00429317	ASCII "50",0	
00429503	ASCII "( H",0	
00429518	ASCII "27"	
004295FC	ASCII "Ftc",0	
00429602	ASCII "e#",0	
00429619	ASCII "ge-l",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!