

Codeengn Challenges Advance RCE LEVEL2 풀이

Reverse2 L02 Start

Author : Noble

Korea :

정답은 무엇인가

English :

Find the answer

[Down](#)

PEID로 확인 한 결과 VC++ 6.0에서 만들어진 프로그램인 걸 알 수 있었다.

실행을 시켜보니 입력한 키값이 제대로 된 키값이 아니면 꺼지는 프로그램 인 것 같았다.

분석을 위해 올리로 켜보았다.

그리고 Enter Password : 라는 스트링을 찾고, 그 근처에 BP를 걸고 프로그램을 분석해 보았다.

004012B3	66:AB	STOS WORD PTR ES:[EDI]	
004012B5	68 30424100	PUSH Reverse2,00414230	ASCII "Enter Password: "
004012BA	68 40844100	PUSH Reverse2,00418440	
004012BF	AA	STOS BYTE PTR ES:[EDI]	
004012C0	E8 BB0A0000	CALL Reverse2,00401080	
004012C5	8D8C24 F80300	LEA ECX,DWORD PTR SS:[ESP+3F8]	
004012CC	51	PUSH ECX	
004012CD	68 D0844100	PUSH Reverse2,004184D0	
004012D2	E8 390D0000	CALL Reverse2,00402010	
004012D7	83C4 10	ADD ESP,10	
004012DA	8D9424 880700	LEA EDX,DWORD PTR SS:[ESP+788]	
004012E1	68 D0144000	PUSH Reverse2,004014D0	
004012E6	68 40144000	PUSH Reverse2,00401440	
004012EB	6A 64	PUSH 64	
004012ED	6A 10	PUSH 10	
004012EF	52	PUSH EDX	
004012F0	E8 B0460000	CALL Reverse2,004059A5	
004012F5	C78424 000000	MOV DWORD PTR SS:[ESP+DD0],0	
00401300	8D9C24 8C0700	LEA EBX,DWORD PTR SS:[ESP+78C]	
00401307	C74424 10 64	MOV DWORD PTR SS:[ESP+10],64	
0040130F	8B4C24 000000	MOV ECX,DWORD PTR SS:[ESP+3F8]	

입력받은 후 Step Over로 계속 분석을 해나갔다.

Address	Hex dump	Disassembly	Comment
00401382	8A9414 8C070	MOV DL, BYTE PTR SS:[ESP+EDX+78C]	
00401389	885424 18	MOV BYTE PTR SS:[ESP+18], DL	
0040138D	E8 7E050000	CALL Reverse2,00401910	
00401392	A1 FC104100	MOV EAX, DWORD PTR DS:[4110FC]	
00401397	8D4C24 14	LEA ECX, DWORD PTR SS:[ESP+14]	
00401398	50	PUSH EAX	
0040139C	57	PUSH EDI	
0040139D	56	PUSH ESI	
0040139E	E8 6D030000	CALL Reverse2,00401710	
004013A3	8B4424 18	MOV EAX, DWORD PTR SS:[ESP+18]	
004013A7	C68424 00000	MOV BYTE PTR SS:[ESP+00], 1	
004013AF	3BC7	CMP EAX, EDI	
004013B1	75 05	JNZ SHORT Reverse2,004013B8	
004013B3	B8 F8104100	MOV EAX, Reverse2,004110F8	
004013B8	8D4C24 24	LEA ECX, DWORD PTR SS:[ESP+24]	
004013BC	50	PUSH EAX	
004013BD	51	PUSH ECX	
004013BE	8D9424 EC050	LEA EDX, DWORD PTR SS:[ESP+5EC]	
004013C5	FFD2	CALL EDX	
004013C7	83C4 08	ADD ESP, 8	
004013CA	E8 07EA0000	CALL Reverse2,0040FDD6	
004013CF	8B4424 18	MOV EAX, DWORD PTR SS:[ESP+18]	
004013D3	3BC7	CMP EAX, EDI	
004013D5	74 1D	JE SHORT Reverse2,004013F4	
004013D7	8D48 FF	LEA ECX, DWORD PTR DS:[EAX-1]	
004013DA	8A40 FF	MOV AL, BYTE PTR DS:[EAX-1]	
004013DD	84C0	TEST AL, AL	
004013DF	74 0A	JE SHORT Reverse2,004013EB	
004013E1	3C FF	CMP AL, 0FF	
004013E3	74 06	JE SHORT Reverse2,004013EB	
004013E5	FEC8	DEC AL	
004013E7	8801	MOV BYTE PTR DS:[ECX], AL	
004013E9	EB 09	JMP SHORT Reverse2,004013F4	
004013EB	51	PUSH ECX	
004013EC	E8 9F190000	CALL Reverse2,00402D90	
004013F1	83C4 04	ADD ESP, 4	
004013F3	C9	EXIT	

EAX=005C2919, (ASCII "ASDF")

그리고 내가 입력한 값을 PUSH하고 함수를 CALL하는 것을 볼 수가 있었었다.

그래서 그 함수를 분석해 보았다.

0012F78A	83F9 43	CMP ECX, 43	
0012F78D	0F85 F7000000	JNZ 0012F88A	
0012F793	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	
0012F796	0FBE48 01	MOVSB ECX, BYTE PTR DS:[EAX+1]	
0012F79A	83F9 52	CMP ECX, 52	
0012F79D	0F85 E7000000	JNZ 0012F88A	
0012F7A3	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	
0012F7A6	0FBE48 02	MOVSB ECX, BYTE PTR DS:[EAX+2]	
0012F7AA	83F9 41	CMP ECX, 41	
0012F7AD	0F85 D7000000	JNZ 0012F88A	
0012F7B3	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	
0012F7B6	0FBE48 03	MOVSB ECX, BYTE PTR DS:[EAX+3]	
0012F7BA	83F9 41	CMP ECX, 41	
0012F7BD	0F85 C7000000	JNZ 0012F88A	
0012F7C3	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	
0012F7C6	0FBE48 04	MOVSB ECX, BYTE PTR DS:[EAX+4]	
0012F7CA	83F9 41	CMP ECX, 41	
0012F7CD	0F85 B7000000	JNZ 0012F88A	
0012F7D3	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	
0012F7D6	0FBE48 05	MOVSB ECX, BYTE PTR DS:[EAX+5]	
0012F7DA	83F9 43	CMP ECX, 43	
0012F7DD	0F85 A7000000	JNZ 0012F88A	
0012F7E3	8B45 0C	MOV EAX, DWORD PTR SS:[EBP+C]	

ECX=00000041

함수에서 내가 입력한 값과 위의 ASCII값들과 비교해 주는걸 볼 수 있었었다.

저기에 있는 ASCII code들을 문자형태로 변환후 인증하니까 그것이 답이었다! :D