



처음 실행화면입니다.

이름이 'CodeEngn' 일때의 패스워드를 찾아야합니다.

CPU - main thread, module 03

Address	Hex dump	ASCII
00401111	6A 20	PUSH 20
00401113	68 38324000	PUSH 03.00403238
00401118	68 EC030000	PUSH 3EC
0040111D	FF75 08	PUSH DWORD PTR SS:[EBP+8]
00401120	E8 B9020000	CALL <JMP.&user32.GetDlgItemTextA>
00401125	83F8 03	CMP EAX,3
00401128	73 18	JNB SHORT 03.00401142
0040112A	6A 10	PUSH 10
0040112C	68 16314000	PUSH 03.00403116
00401131	68 F1304000	PUSH 03.004030F1
00401136	6A 00	PUSH 0
00401138	E8 B3020000	CALL <JMP.&user32.MessageBoxA>
0040113D	E9 8F000000	JMP 03.004011D1
00401142	B8 38324000	MOV EAX,03.00403238
00401147	A3 58324000	MOV DWORD PTR DS:[403258],EAX
0040114C	6A 00	PUSH 0
0040114E	E8 24010000	CALL 03.00401277
00401153	6A 20	PUSH 20
00401155	68 64324000	PUSH 03.00403264
0040115A	68 E0030000	PUSH 3E0
0040115F	FF75 08	PUSH DWORD PTR SS:[EBP+8]
00401162	E8 77020000	CALL <JMP.&user32.GetDlgItemTextA>
00401167	FF35 00304000	TEST DWORD PTR DS:[403000]
0040116D	68 84304000	PUSH 03.00403084
00401172	68 84324000	PUSH 03.00403284
00401177	E8 4A020000	CALL <JMP.&user32.wsprintfA>
0040117C	83C4 0C	ADD ESP,0C

EAX=00000008

Registers (FPU)

Register	Value
EAX	00000008
ECX	759C6F42 user32.759C6F42
EDX	00000030
EBX	00000000
ESP	0012FA9C
EBP	0012FAA4
ESI	00000111
EDI	0012FB20
EIP	00401125 03.00401125
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_SUCCESS (00000)
EFL	00000246 (NO,NB,E,BE,NS,PE,GE)
ST0	empty 0.0
ST1	empty 0.0
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0

Count = 20 (32.)
Buffer = 03.00403238
ControlID = 3EC (1004.)
hWnd = 00000000
GetDlgItemTextA
이름! 3자이상이어야한다
Style = MB_OK|MB_ICONHAND|MB_APPLMODAL
Title = "You failed..."
Text = "No, that is not the right answer :)"
hOwner = NULL
MessageBoxA
ASCII "CodeEngn"
CodeEngn 있는주소값을 넣는다
패스워드만드는거
Count = 20 (32.)
Buffer = 03.00403264
ControlID = 3ED (1005.)
hWnd = 00000000
GetDlgItemTextA
<Xu> = 0
Format = "%u"
s = 03.00403284
wsprintfA
FST 4020 Cond 1 0 0 0 Err 0 0 1

Address	Hex dump	ASCII
00403238	43 6F 64 65 45 6E 67 6E	CodeEngn
00403240	00 00 00 00 00 00 00 00
00403248	00 00 00 00 00 00 00 00
00403250	00 00 00 00 00 00 00 00

0012FA84 0012FAA4
0012FA88 00401125 03.00401125
0012FA8C 001F04EC
0012FA90 000003EC
0012FA94 00403238 ASCII "CodeEngn"

첫 번째 밑줄을 보시면 GetDlgItemTextA를 통해 이름을 받아옵니다.

그 후 이름이 3자이상인지 체크하고 다음으로 넘어갑니다.

두 번째 밑줄은 입력받은 이름을 이용해서 패스워드를 만듭니다.

내부 함수를 분석하고자 했으나 복잡하고 모르는 어셈블리어가 많아서 다음기회에... ㅋㅋ

CPU - main thread, module 03			
00401136	. 6A 00	PUSH 0	hOwner = NULL
00401138	. E8 B3020000	CALL <JMP.&user32.MessageBoxA>	MessageBoxA
0040113D	~ E9 8F000000	JMP 03.004011D1	
00401142	> B8 38324000	MOV EAX,03.00403238	ASCII "CodeEngn"
00401147	. A3 58324000	MOV DWORD PTR DS:[403258],EAX	CodeEngn 있는 주소값을 넣는다
0040114C	. 6A 00	PUSH 0	
0040114E	. E8 24010000	CALL 03.00401277	패스워드만드는거
00401153	. 6A 20	PUSH 20	Count = 20 (32.)
00401155	. 68 64324000	PUSH 03.00403264	Buffer = 03.00403264
0040115A	. 68 ED030000	PUSH 3ED	ControlID = 3ED (1005.)
0040115F	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
00401162	. E8 77020000	CALL <JMP.&user32.GetDlgItemTextA>	GetDlgItemTextA
00401167	. FF35 00304000	PUSH DWORD PTR DS:[403000]	<%u> = C2A776FA (3265754874.)
0040116D	. 68 84304000	PUSH 03.00403084	Format = "%u"
00401172	. 68 84324000	PUSH 03.00403284	s = 03.00403284
00401177	. E8 4A020000	CALL <JMP.&user32.wsprintfA>	wsprintfA
0040117C	. 83C4 0C	ADD ESP,0C	
0040117F	. 33C0	XOR EAX,EAX	
00401181	. A3 00304000	MOV DWORD PTR DS:[403000],EAX	
00401186	. 892D 5C324000	MOV DWORD PTR DS:[40325C],EBP	
0040118C	. 68 64324000	PUSH 03.00403264	String2 = "1234"
00401191	. 68 84324000	PUSH 03.00403284	String1 = "3265754874"
00401196	. E8 25020000	CALL <JMP.&kernel32.lstrcmpA>	lstrcmpA
0040119B	. 99	CDQ	
0040119C	. F7F8	IDIV EAX	
0040119E	. 6A 10	PUSH 10	Style = MB_OK MB_ICONHAND MB_...
004011A0	. 68 16314000	PUSH 03.00403116	Title = "You failed..."
ESP=0012FA90			

만들어진 패스워드를 wsprintf를 통해 00403284번지에 저장합니다.

그 후 입력한 패스워드와 만들어진 패스워드를 lstrcmp를 통해 비교하고 분기합니다.