

두개의 값을 입력받고, 검사결과에 따라 성공메시지를 출력한다.

이렇게 값을 잘못 입력하면, 에러 메시지를 보여준다.

004010AA	PUSH KeygenMe.00403021	ASCII "MainWindow"
00401179	PUSH KeygenMe.004034DC	ASCII " Tut selfkeygenMe "
0040127A	PUSH KeygenMe.00403462	ASCII "KeyGen lena151 "
0040127F	PUSH KeygenMe.0040323C	ASCII "It's quite simple : (self)keygen me. Good luck !!!!!
004012B3	PUSH KeygenMe.00403038	ASCII "asdf"
004012C9	PUSH KeygenMe.00403138	ASCII "1234"
004012E1	PUSH KeygenMe.00403462	ASCII "KeyGen lena151 "
004012E6	PUSH KeygenMe.00403000	ASCII " Give me more material hehe!?"
004012F6	PUSH KeygenMe.00403038	ASCII "asdf"
00401330	JNZ SHORT KeygenMe.00401353	(Initial CPU selection)
00401340	PUSH KeygenMe.00403462	ASCII "KeyGen lena151 "
00401345	PUSH KeygenMe.004034B8	ASCII " That's right. (Self)keygen me now!"
00401355	PUSH KeygenMe.00403462	ASCII "KeyGen lena151 "
0040135A	PUSH KeygenMe.004034B6	ASCII " Error detected! Remove debugger from Hard Drive "
00403000	ASCII " Give me more"	
00403010	ASCII " material hehe!?"	
00403020	ASCII "	

아까 본 문자열을 검색해서 이 문자열을 보여주는 루틴으로 이동한다.

메시지 출력 위쪽에 비교문이 있고, 분기문기문이 있다. [403138]에는 내가 입력한 시리얼 값이 있고, 그 전에는 반복문으로 내가 입력한 이름에 따라 시리얼을 생성해주는 코드가 있다.

00401330	. 03F6	ADD ESI,ESI	
00401332	. 40	INC EAX	
00401333	. 49	DEC ECX	
00401334	. 75 D3	JNZ SHORT KeygenMe.00401309	
00401336	. 3B35 38314000	CMP ESI,DWORD PTR DS:[403138]	
0040133C	. 75 15	JNZ SHORT KeygenMe.00401353	
0040133E	. 6A 00	PUSH 0	Style = MB_OK MB_APPLMODAL
00401340	. 68 62344000	PUSH KeygenMe.00403462	Title = "KeyGen lena151 "
00401345	. 68 B8344000	PUSH KeygenMe.004034B8	Text = " That's right. (Self)keygen me now!"
0040134A	. 6A 00	PUSH 0	hOwner = NULL
0040134D	. E8 90000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
00401351	. EB 13	JMP SHORT KeygenMe.00401366	
00401353	. 6A 00	PUSH 0	Style = MB_OK MB_APPLMODAL
00401355	. 68 62344000	PUSH KeygenMe.00403462	Title = "KeyGen lena151 "
0040135A	. 68 B8344000	PUSH KeygenMe.004034B6	Text = " Error detected! Remove debugger from Hard Drive "
0040135F	. 6A 00	PUSH 0	hOwner = NULL
00401361	. E8 88000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
00401366	. EB 15	JMP SHORT KeygenMe.0040137D	
00401368	. FF75 14	PUSH DWORD PTR SS:[EBP+14]	r lParam

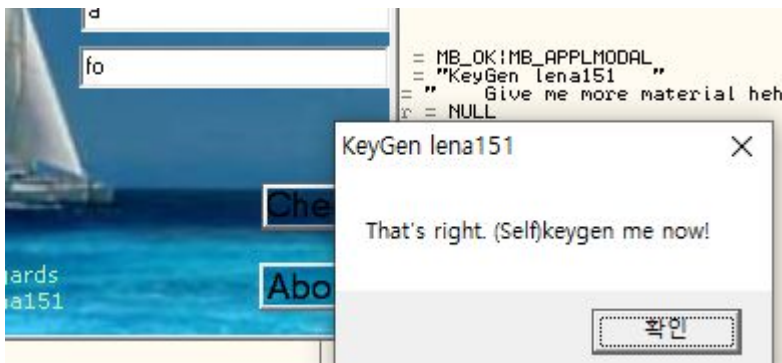
004012AB	· 0F85 B5000000	JNZ KeygenMe.00401366	
004012B1	· 6A 1A	PUSH 1A	[Count = 1A (26.)
004012B3	· 68 38304000	PUSH KeygenMe.00403038	Buffer = KeygenMe.00403038
004012B8	· 6A 6A	PUSH 6A	ControlID = 6A (106.)
004012BA	· FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
004012BC	· E8 08010000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
004012C2	· 83F8 00	CMP EAX,0	
004012C5	· 74 18	JE SHORT KeygenMe.004012DF	
004012C7	· 6A 1A	PUSH 1A	[Count = 1A (26.)
004012C9	· 68 38314000	PUSH KeygenMe.00403138	Buffer = KeygenMe.00403138
004012CE	· 6A 6B	PUSH 6B	ControlID = 6B (107.)
004012D0	· FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
004012D2	· E8 F2000000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
004012D8	· 83F8 00	CMP EAX,0	
004012DB	· 74 02	JE SHORT KeygenMe.004012DF	
004012DD	· EB 17	JMP SHORT KeygenMe.004012F6	
004012DF	> 6A 00	PUSH 0	[Style = MB_OK!MB_APPLMODAL
004012E1	· 68 62344000	PUSH KeygenMe.00403462	Title = "KeyGen lena151"
004012E6	· 68 00304000	PUSH KeygenMe.00403000	Text = "Give me more material hehe!!"
004012EB	· 6A 00	PUSH 0	hOwner = NULL
004012ED	· E8 FC000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
004012F2	· C9	LEAVE	
004012F3	· C2 1000	RETN 10	
004012F6	> 68 38304000	PUSH KeygenMe.00403038	[String = "a"
004012FB	· E8 30010000	CALL <JMP.&kernel32.lstrlen>	lstrlenA
00401300	· 33F6	XOR ESI,ESI	

GetDlgItemTextA API를 통해
사용자에게 두개의 입력을 받는다.
그리고 0과 비교해 같으면 “Give
me more material hehe!!” 값을 더
입력하라는 오류 메시지를
출력한다.

즉 EAX는 사용자가 입력한 문자
개수를 의미한다.

004012F3	. C2 1000	RETN 10	
004012F6	> 68 38304000	PUSH KeygenMe.00403038	[String = "a"
004012FB	. E8 30010000	CALL <JMP.&kernel32.lstrlen>	lstrlenA
00401300	. 33F6	XOR ESI,ESI	
00401302	. 8BC8	MOV ECX,EAX	
00401304	. B8 01000000	MOV EAX,1	
00401309	> 8B15 38304000	MOV EDX,DWORD PTR DS:[403038]	
0040130F	. 8A90 37304000	MOV DL,BYTE PTR DS:[EAX+403037]	
00401315	. 81E2 FF000000	AND EDX,0FF	
00401318	. 8BDA	MOV EBX,EDX	
0040131D	. 0FAFDA	IMUL EBX,EDX	
00401320	. 03F3	ADD ESI,EBX	
00401322	. 8BDA	MOV EBX,EDX	
00401324	. D1FB	SAR EBX,1	
00401326	. 83C3 03	ADD EBX,3	
00401329	. 0FAFDA	IMUL EBX,EDX	
0040132C	. 2BDA	SUB EBX,EDX	
0040132E	. 03F3	ADD ESI,EBX	
00401330	. 03F6	ADD ESI,ESI	
00401332	. 40	INC EAX	
00401333	. 49	DEC ECX	
00401334	. ^75 D3	JNZ SHORT KeygenMe.00401309	
00401336	. 3B35 38314000	CMP ESI,DWORD PTR DS:[403138]	
0040133F	. ^75 1C	JNZ SHORT KeygenMe.00401353	

00401340	. E8 90000000	
DS:[00403138]=00000031		
ESI=00006F66		
Address	Hex dump	



입력을 다 했으면, 내가 입력한 a값을 통해 반복문으로 시리얼 값을 생성한다. 그리고 ESI에는 a에맞는 시리얼값과 403138에는 내가 입력한 시리얼 값이 들어있다. 이 값이 일치하면 성공 메시지를 출력한다.

ESI에는 6F66이라는 값이 들어있었고, 이 값을 문자로 표현하면 'fo'이다.