

악성코드 분석 보고서

(sand-reversingwithlana-tutorials)

2025.07.23

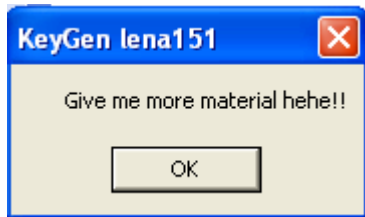
1. 문제



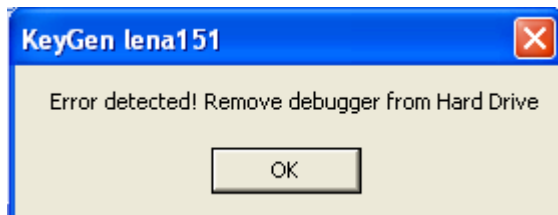
키젠(Key generator) 값 찾기

키젠(Key generator) : 열쇠 생성기. 말 그대로 유료 소프트웨어 또는 유료 서비스를 사용할 수 있는 시리얼 코드 인증을 무력화하는 프로그램

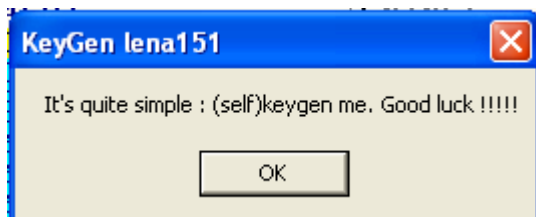
2. 해결 방법



KeygenMe에서 값을 둘 다 입력 안 했을 때와 하나만 입력 했을 때 'Give me more material hehe!!' 해당 문구가 있는 창이 발생한다.

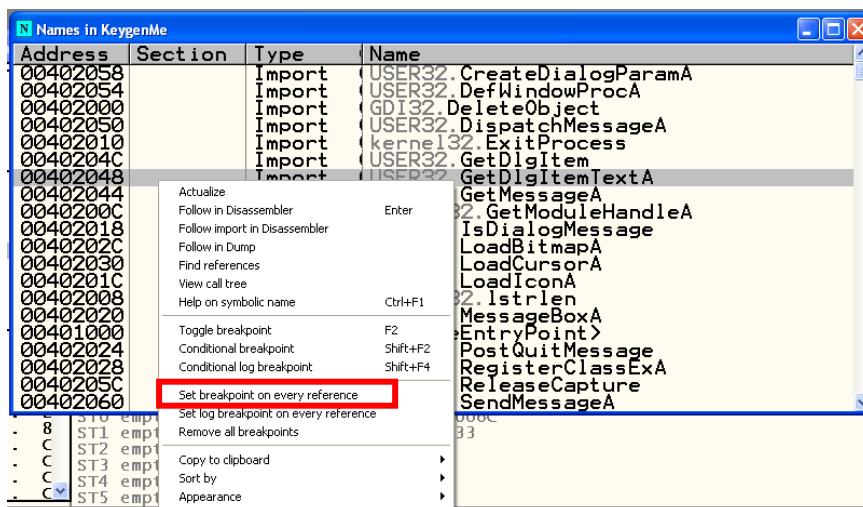


둘 다 입력했을 때 'Error detected! Remove debugger from Hard Drive' 해당 문구가 있는 창이 발생한다.



About을 눌렀을 때 'It's quite simple : (self)keygen me. Good luck !!!!!' 해당 문구가 있는 창이 발생한다.

GetDlgItemTextA API를 사용했을 거라고 생각된다.



'Ctrl+n' 눌러서 GetDlgItemTextA를 사용한 모든 곳에 브레이크를 걸어준다.

총 3개의 박스에 브레이크 걸리는 걸 알 수 있다.

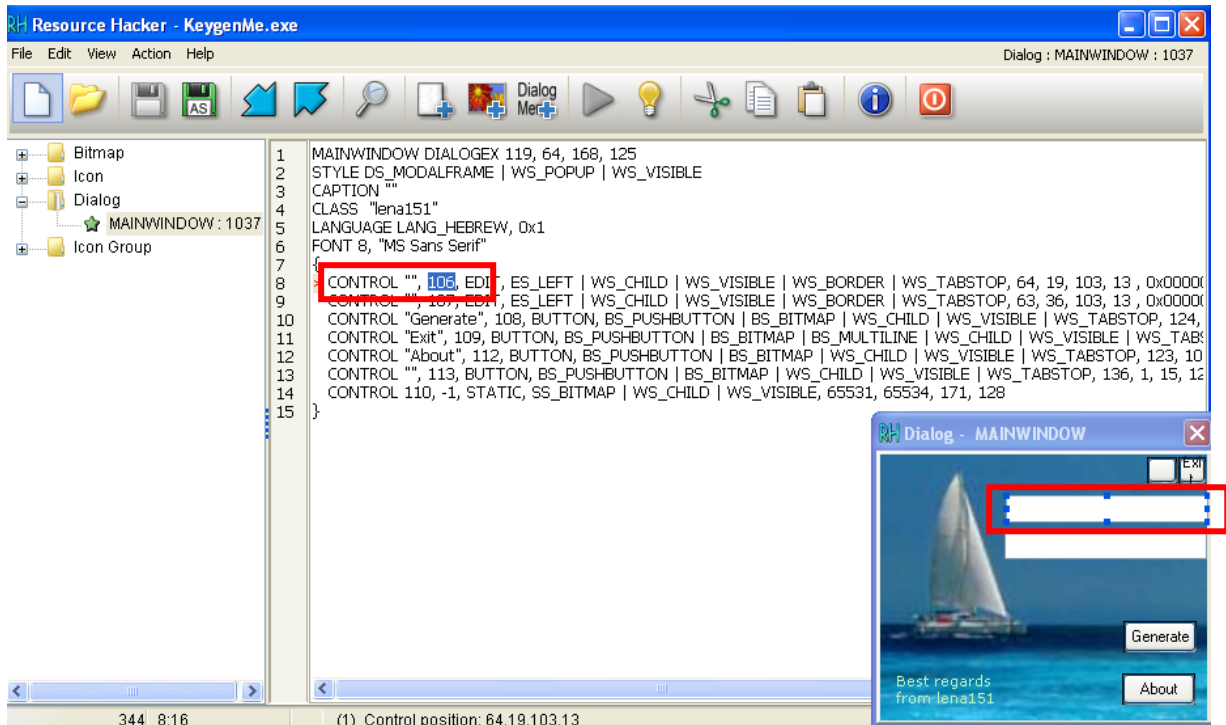


'F9'을 눌러서 실행해주고 값 입력 후 'Check'를 눌러주면 브레이크 걸린 곳에서 멈춘다.

004012A1	~\ 0F85 BF000000	JNZ KeygenMe.00401366	
004012A7	58	POP EAX	
004012A8	83F8 6C	CMP EAX, 6C	
004012AB	~\ 0F85 B5000000	JNZ KeygenMe.00401366	
004012B1	6A 1A	PUSH 1A	
004012B3	68 38304000	PUSH KeygenMe.00403038	
004012B8	6A 6A	PUSH 6A	
004012BA	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
004012BD	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	
004012C9	68 38314000	PUSH KeygenMe.00403138	
004012CE	6A 6B	PUSH 6B	
004012D0	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
004012D3	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	
004012E1	68 62344000	PUSH KeygenMe.00403462	
004012E6	68 00304000	PUSH KeygenMe.00403000	
004012EB	6A 00	PUSH 0	
004012ED	E8 FC000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA

밑에 메시지 박스를 보면 아무것도 입력 안 했을 때 메시지가 보인다.

GetDlgItemTextA이 뭔지 msdn에서 설명을 읽어보면 '대화 상자에서 컨트롤과 연결된 제목 또는 텍스트를 검색'하는 함수라고 나온다. 그리고 반환 값은 '버퍼에 복사된 문자 수를 반환'한다고 나온다.



2개의 함수에서 인자 값인 ControlID를 보면 6A(106.), 6B6A(107.)가 나오는데 Resource Hacker 프로그램을 이용해서 보면 이렇게 ID가 지정하는 곳을 볼 수 있다.

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
004012BD	. E8 08010000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
004012C2	. 83F8 00	CMP EAX,0	

Address	Hex dump	ASCII
00403038	6C 65 65 31 32 33 00 00	tee123..
00403040	00 00 00 00 00 00 00 00
00403048	00 00 00 00 00 00 00 00

첫 번째 GetDlgItemTextA를 실행해보면 Buffer를 0x00403038로 지정한 걸 볼 수 있다. 이 곳으로 가보면 저장된 걸 볼 수 있다. 그리고 반환 값을 버퍼에 복사된 문자 수를 반환한다고 했는데 EAX를 보면 6이 저장된 걸 볼 수 있다.

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
004012BD	. 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
004012D3	. 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	
004012E1	. 68 62344000	PUSH KeygenMe.00403462	Style = MB_OK MB_APPLMODAL
004012E6	. 68 00304000	PUSH KeygenMe.00403000	Title = "KeyGen lena151"
004012EB	. 6A 00	PUSH 0	Text = " Give me more material he
004012ED	. E8 FC000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA

아까 값을 입력했기 때문에 0x004012C5에서 값을 입력하지 않았을 때 나오는 메시지 함수로 넘어가지 않는 걸 볼 수 있다.

Address	Hex dump	ASCII
00403138	31 32 33 34 35 00 00 00	12345..
00403140	00 00 00 00 00 00 00 00
00403148	00 00 00 00 00 00 00 00

두 번째 GetDlgItemTextA를 실행해보면 Buffer를 0x00403138로 지정한 걸 볼 수 있다. 이 곳으로 가보면 저장된 걸 볼 수 있다.

그리고 반환 값을 버퍼에 복사된 문자 수를 반환한다고 했는데 EAX를 보면 5가 저장된 걸 볼 수 있다.

그리고 똑같이 값을 입력하지 않았을 때 나오는 메시지 함수로 넘어가지 않는 걸 볼 수 있다.

004012DB	< 74 02	JE SHORT KeygenMe.004012DF	
004012DD	< EB 17	JMP SHORT KeygenMe.004012F6	
004012DF	> 6A 00	PUSH 0	
004012E1	> 68 62344000	PUSH KeygenMe.00403462	
004012E6	> 68 00304000	PUSH KeygenMe.00403000	
004012E8	> 6A 00	PUSH 0	
004012ED	> E8 FC000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
004012F2	> C9	LEAVE	
004012F3	> C2 1000	RETN 10	
004012F6	> 68 38304000	PUSH KeygenMe.00403038	
004012FB	> E8 30010000	CALL <JMP.&kernel32.lstrlen>	lstrlenA

0x004012F6으로 넘어가는 걸 볼 수 있는 lstrlenA 함수는 msdn에서 '지정된 문자열의 길이 출력' 라고 나오는 걸 볼 수 있다.

Registers (FPU)	
EAX	00000006
ECX	7C80BE86 kernel32.7C80BE86
EDX	00403039 ASCII "ee123"
EBX	00000000

그래서 EAX에 6자리가 저장된다.

004012FB	. E8 30010000	CALL <JMP.&kernel32.lstrlen>
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]

0x00401309 – 0x00401334에서 루프가 실행되는 걸 볼 수 있다. 해석해보면 우선 401300에서 ESI를 0으로 초기화하고 EAX를 1로 바꾸는 걸 볼 수 있다.

401309에서 EDX에 403038의 값을 저장하는데 403038에는 'lee123'이 저장되어 있다.

그럼 EDX=3165656C가 저장된다.(리틀엔디언이라서 거꾸로 저장이된다.)

아마 이 루프에서는 'lee123'의 키를 만드는 것으로 보인다. 왜냐하면 401336에서 cmp를 이용하여 ESI와 403138의 값을 비교하기 때문이다.

Registers (FPU)	
EAX	00000007
ECX	00000000
EDX	00000033
EBX	00000561
ESP	0012FC48
EBP	0012FC48
ESI	001D6B84
EDI	0012FCB0

403138에는 두 번째 상자에 입력했던 '12345'가 있고 ESI에는 루프를 돌면서 나왔던 값이 있다.

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 88344000	PUSH KeygenMe.00403488	Text = "That's right. (Self)keygen me"
0040134A	. 6A 00	PUSH 0	hOwner = NULL
0040134C	. E8 9D000000	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 86344000	PUSH KeygenMe.00403486	Text = "Error detected! Remove debugg"
0040135F	. 6A 00	PUSH 0	hOwner = NULL
00401361	. E8 88000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA

근데 비교했을 때 같지 않으므로 0x00401353으로 가서 에러문을 출력하는 걸 볼 수 있다.

그럼 여기서 첫 번째에 입력한 값이 루프를 이용하여 키를 만들어내고 두 번째 값과 같아야한다는 걸 알 수 있다.

lee123의 키는 846B1D00(16진수) 이걸 아스키로 변환한 값이 k이므로 이걸 입력해보면 원래는 나오는 데 프로그램에서 인식을 못한다.



첫 번째에 a를 입력하면 666F0000(16진수) 이걸 받아온다 아스키로 변환하면 fo이므로 이걸 입력하면 성공하는 걸 볼 수 있다.

문제 해결!