



처음 실행화면입니다. 그냥 체크버튼을 눌러보니 에러메세지가 발생합니다.

올바른 시리얼을 찾기 위해서 프로그램 분석을 해보도록 하겠습니다.

CPU - main thread, module 07

Address	Hex dump	ASCII
00402324	61 62 63 64 00 00 00 00	abcd....

보다 확실하게 하기 위해 입력값을 abcd로 입력하도록 하겠습니다.

break point가 설정 된 00401078 번지를 보면 GetDlgItemTextA라는 함수를 호출합니다.

이는 프로그램에서 텍스트박스에 입력한 값을 가져오는 함수입니다.

호출 뒤에 밑에 보시면 abcd가 들어간 것을 볼 수 있습니다.

이 함수를 통해 입력 값을 가져오는 것을 확인했습니다.

CPU - main thread, module 07			
00401073	6A 68	PUSH 68	ControlID = 68 (104.)
00401075	FF75 08	PUSH DWORD PTR SS:[EBP+8]	hWnd
00401078	E8 F4000000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
0040107D	6A 00	PUSH 0	pFileSystemNameSize = NULL
0040107F	6A 00	PUSH 0	pFileSystemNameBuffer = NULL
00401081	68 C8204000	PUSH 07.004020C8	pFileSystemFlags = 07.004020C8
00401086	68 90214000	PUSH 07.00402190	pMaxFilenameLength = 07.00402190
00401088	68 94214000	PUSH 07.00402194	pVolumeSerialNumber = 07.00402194
00401090	6A 32	PUSH 32	MaxVolumeNameSize = 32 (50.)
00401092	68 5C224000	PUSH 07.0040225C	VolumeNameBuffer = 07.0040225C
00401097	6A 00	PUSH 0	RootPathName = NULL
00401099	E8 B5000000	CALL <JMP.&KERNEL32.GetVolumeInformationA>	GetVolumeInformationA
0040109E	68 F3234000	PUSH 07.004023F3	StringToAdd = "4562-ABEX"
004010A3	68 5C224000	PUSH 07.0040225C	ConcatString = ""
004010A8	E8 94000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010AD	B2 02	MOV DL,2	
004010AF	> 8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010C8	FECA	DEC DL	
004010CD	75 E0	JNZ SHORT 07.004010AF	
004010CF	68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010DE	68 5C224000	PUSH 07.0040225C	StringToAdd = ""
004010E3	68 00204000	PUSH 07.00402000	ConcatString = ""
00401153=<JMP.&KERNEL32.GetVolumeInformationA>			
Address	Hex dump	ASCII	
0040225C	00 00 00 00 00 00 00 00	

두 번째 break point인 GetVolumeInformation 함수는 드라이버의 정보를 가져올 때 사용하는 함수입니다.

밑줄과 박스로 표시한 것은 드라이버의 이름을 얻어오는 버퍼입니다.

그리고 밑줄 바로 밑에 RootPathName=NULL이라고 되어 있는데 이 값이 NULL이면 C:\를 가리키게 됩니다.

지금 빈 칸인 이유는 로컬 디스크라는 이름이 비어있는 이름을 뜻하기 때문입니다.

CPU - main thread, module 07			
00401078	E8 F4000000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
0040107D	6A 00	PUSH 0	pFileSystemNameSize = NULL
0040107F	6A 00	PUSH 0	pFileSystemNameBuffer = NULL
00401081	68 C8204000	PUSH 07.004020C8	pFileSystemFlags = 07.004020C8
00401086	68 90214000	PUSH 07.00402190	pMaxFilenameLength = 07.00402190
0040108B	68 94214000	PUSH 07.00402194	pVolumeSerialNumber = 07.00402194
00401090	6A 32	PUSH 32	MaxVolumeNameSize = 32 (50.)
00401092	68 5C224000	PUSH 07.0040225C	VolumeNameBuffer = 07.0040225C
00401097	6A 00	PUSH 0	RootPathName = NULL
00401099	E8 B5000000	CALL <JMP.&KERNEL32.GetVolumeInformationA>	GetVolumeInformationA
0040109E	68 F3234000	PUSH 07.004023F3	StringToAdd = "4562-ABEX"
004010A3	68 5C224000	PUSH 07.0040225C	ConcatString = "4562-ABEX"
004010A8	E8 94000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010AD	B2 02	MOV DL,2	
004010AF	> 8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010CB	FECA	DEC DL	
004010CD	75 E0	JNZ SHORT 07.004010AF	
004010CF	68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010DE	68 5C224000	PUSH 07.0040225C	StringToAdd = "4562-ABEX"
004010E3	68 00204000	PUSH 07.00402000	ConcatString = ""
004010E8	E8 54000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010ED	68 24234000	PUSH 07.00402324	String2 = "abcd"
DL=0A (Line Feed)			
Address	Hex dump	ASCII	
0040225C	34 35 36 32 2D 41 42 45	4562-ABE	
00402264	58 00 00 00 00 00 00 00	X.....	

현재 라인을 보시면 위에 두 변수를 가지고 strcat 함수를 호출하는 것을 알 수 있습니다.

strcat 함수는 문자열을 붙이는 함수인데 인자값으로 4562-ABEX와 07.0040225C 값을 가지고 옵니다.

07.0040225C는 드라이버의 이름인 빈 값이므로 strcat 함수를 지나도 4562-ABEX라는 값이 됩니다.

004010AD	B2 02	MOV DL,2	
004010AF	> 8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010CB	FECA	DEC DL	
004010CD	75 E0	JNZ SHORT 07.004010AF	
004010CF	68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	68 00204000	PUSH 07.00402000	ConcatString = "L2C-57816784-ABEX"
004010D9	E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010DE	68 5C224000	PUSH 07.0040225C	StringToAdd = "6784-ABEX"
004010E3	68 00204000	PUSH 07.00402000	ConcatString = "L2C-57816784-ABEX"
004010E8	E8 54000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010ED	68 24234000	PUSH 07.00402324	String2 = "abcd"
004010F2	68 00204000	PUSH 07.00402000	String1 = "L2C-57816784-ABEX"
004010F7	E8 51000000	CALL <JMP.&KERNEL32.lstrcmpA>	lstrcmpA
004010FC	83F8 00	CMP EAX,0	

DL에 2를 넣고 두 번 반복하는데 0040225C의 값 앞의 4자리 각각 1씩 더합니다.

4562 + 1 1 1 1 = 6784-ABEX 값이 되고 그 후에 L2C-5781 + 연산한 값을 한 것이 시리얼 값입니다.

004010FC	83F8 00	CMP EAX,0	
004010FF	74 16	JE SHORT 07.00401117	
00401101	6A 00	PUSH 0	
00401103	68 34244000	PUSH 07.00402434	
00401108	68 3B244000	PUSH 07.0040243B	
0040110D	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
00401110	E8 56000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
00401115	EB 16	JMP SHORT 07.0040112D	
00401117	6A 00	PUSH 0	
00401119	68 06244000	PUSH 07.00402406	
0040111E	68 11244000	PUSH 07.00402411	
00401123	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
00401126	E8 40000000	CALL <JMP.&USER32.MessageBoxA>	MessageBoxA
0040112B	EB 00	JMP SHORT 07.0040112D	
0040112D	6A 00	PUSH 0	
0040112F	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
00401132	E8 22000000	CALL <JMP.&USER32.EndDialog>	EndDialog
00401137	C9	LEAVE	
00401138	C2 1000	RETN 10	

L2C-57816784-ABEX와 abcd를 비교하고 그 결과를 메세지박스로 출력해줍니다.

←

→

컴퓨터

구성

시스템 속성

프로그램 제거 또는 변경

네트워크 드라이브 연결

즐거찾기

다운로드

바탕 화면

최근 위치

하드 디스크 드라이브 (1)

CodeEngn (C:)

59.9GB 중 52.1GB 사용 가능

CPU - main thread, module 07

00401078	E8 F4000000	CALL <JMP.&USER32.GetDlgItemTextA>	GetDlgItemTextA
0040107D	6A 00	PUSH 0	pFileSystemNameSize = NULL
0040107F	6A 00	PUSH 0	pFileSystemNameBuffer = NULL
00401081	68 C8204000	PUSH 07.004020C8	pFileSystemFlags = 07.004020C8
00401086	68 90214000	PUSH 07.00402190	pMaxFilenameLength = 07.00402190
00401088	68 94214000	PUSH 07.00402194	pVolumeSerialNumber = 07.00402194
00401090	6A 32	PUSH 32	MaxVolumeNameSize = 32 (50.)
00401092	68 5C224000	PUSH 07.0040225C	VolumeNameBuffer = 07.0040225C
00401097	6A 00	PUSH 0	RootPathName = NULL
00401099	E8 B5000000	CALL <JMP.&KERNEL32.GetVolumeInformationA>	GetVolumeInformationA
0040109E	68 F3234000	PUSH 07.004023F3	StringToAdd = "4562-ABEX"
004010A3	68 5C224000	PUSH 07.0040225C	ConcatString = "CodeEngn"
004010A8	E8 94000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010AD	B2 02	MOV DL,2	
004010AF	8305 5C224000	ADD DWORD PTR DS:[40225C],1	
004010B6	8305 5D224000	ADD DWORD PTR DS:[40225D],1	
004010BD	8305 5E224000	ADD DWORD PTR DS:[40225E],1	
004010C4	8305 5F224000	ADD DWORD PTR DS:[40225F],1	
004010CB	FECA	DEC DL	
004010CD	75 E0	JNZ SHORT 07.004010AF	
004010CF	68 FD234000	PUSH 07.004023FD	StringToAdd = "L2C-5781"
004010D4	68 00204000	PUSH 07.00402000	ConcatString = ""
004010D9	E8 63000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010DE	68 5C224000	PUSH 07.0040225C	StringToAdd = "CodeEngn"
004010E3	68 00204000	PUSH 07.00402000	ConcatString = ""
004010E8	E8 54000000	CALL <JMP.&KERNEL32.lstrcatA>	lstrcatA
004010ED	68 24234000	PUSH 07.00402324	String2 = "abcd"

004023F3=07.004023F3 (ASCII "4562-ABEX")

Address	Hex dump	ASCII
0040225C	43 6F 64 65 45 6E 67 6E	CodeEngn

아까 빈칸이었던 0040225C 번지에 CodeEngn값이 들어갔습니다.

이를 이용해서 연산하게 되면 원하는 결과를 얻을 수 있습니다.