

코드 엔진 Challenges: Basic 12

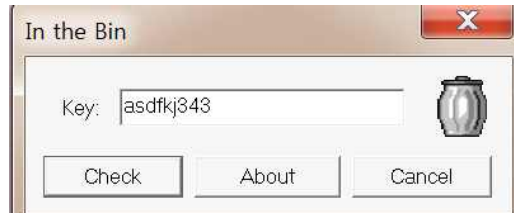
Author: Basse 2002

Korean: Key를 구한 후 입력하게 되면 성공메시지를 볼 수 있다.

이때 성공메시지 대신 key 값이 MessageBox에 출력되도록 하려면 파일을 HexEdit로 오픈한 다음 0x????~0x???? 영역에 Key값을 overwrite 하면 된다.

문제 : key값+주소영역을 찾으시오 .

문제를 확인했으니 파일을 다운로드 받아서 실행해보자.



확인을 위해 아무값을 넣어봤으나 어떠한 메시지 창도 뜨지않았다. 옳은 값을 넣어야 성공 메시지가 나오고 이 성공메시지를 수정하는 것이 이번 문제의 목표다. key를 구하기 위해 파일을 분석해보자. 일단 파일에 패킹여부를 DE를 통해 확인해봤으나 패킹은 되어있지않다. 올리디버거를 이용해 파일을 열어보자.

```
CPU - main thread, module 12
00401000 6A 00 PUSH 0
00401002 E8 AB010000 CALL <JMP.&KERNEL32.GetModuleHandleA>
00401007 A3 64364000 MOV DWORD PTR DS:[403664],EAX
0040100C 6A 00 PUSH 0
0040100E 68 29104000 PUSH 12.00401029
00401013 6A 00 PUSH 0
00401015 6A 65 PUSH 65
00401017 FF35 64364000 PUSH DWORD PTR DS:[403664]
0040101D E8 60010000 CALL <JMP.&USER32.MessageBoxParamA>
00401022 6A 00 PUSH 0
00401024 E8 83010000 CALL <JMP.&KERNEL32.ExitProcess>
00401029 55 PUSH EBP
0040102A 8BEC MOV EBP,ESP
0040102C 8B45 0C MOV EAX,DWORD PTR SS:[EBP+0C]
0040102F 3D 11010000 CMP EAX,111
00401034 0F85 97000000 JNZ 12.004010D1
00401039 8B55 10 MOV EDI,DWORD PTR SS:[EBP+10]
```

```
[pModule = NULL
~GetModuleHandleA
IParam = NULL
DlgProc = 12.00401029
hOwner = NULL
pTemplate = 65
hInst = NULL
DialogBoxParamA
ExitCode = 0
ExitProcess
```

사용된 문자열을 확인해서 키 값을 찾아보자 .

Address	Disassembly	Text string
00401000	PUSH 0	(Initial CPU selection)
00401063	MOV ESI,12.00403000	ASCII "0qiqb4EhW/4jISMjlzQf6kp6QwLrG+GEIV4bPc0JL/jWBfNLejmbme3ga"
00401086	PUSH 12.00403530	ASCII "In the Bin"
0040108B	PUSH 12.0040353B	ASCII "Congratulation, you found the right key"

키 값이 맞을 경우 나올 것 같은 문자열인 "Congratulation~ " 문자열을 클릭 해 해당 주소로 이동해보자.

00401043	0F85 B4000000	JNZ 12.004010FD	
00401049	8B45 10	MOV EAX,DWORD PTR SS:[EBP+10]	
0040104C	66:83F8 01	CMP AX,1	
00401050	75 4A	JNZ SHORT 12.0040109C	
00401052	6A 00	PUSH 0	
00401054	6A 00	PUSH 0	
00401056	68 B90B0000	PUSH 0BB9	
0040105B	FF75 08	PUSH DWORD PTR SS:[EBP+8]	
0040105E	E8 31010000	CALL <JMP.8USER32.GetDlgItemInt>	IsSigned = FALSE pSuccess = NULL ControlID = BB9 (3001.) hWnd
00401063	BE 00304000	MOV ESI,12.00403000	GetDlgItemInt
00401068	833E 00	CMP DWORD PTR DS:[ESI],0	ASCII "0qiqb4EhM/4jISMjlzQf6kpGQwLrG+
0040106B	75 04	JNZ SHORT 12.00401071	
0040106D	EB 0E	JMP SHORT 12.0040107D	
0040106F	EB 0C	JMP SHORT 12.0040107D	
00401071	8B1E	MOV EBX,DWORD PTR DS:[ESI]	
00401073	E8 97000000	CALL 12.0040110F	
00401078	83C6 04	ADD ESI,4	
0040107B	EB EB	JMP SHORT 12.00401068	
0040107D	3D BF96287A	CMP EAX,7A2896BF	
00401082	75 14	JNZ SHORT 12.00401098	
00401084	6A 40	PUSH 40	
00401086	68 30354000	PUSH 12.00403530	Style = MB_OK MB_ICONASTERISK MB_APPL
0040108B	68 3B354000	PUSH 12.0040353B	Title = "In the Bin"
00401090	FF75 08	PUSH DWORD PTR SS:[EBP+8]	Text = "Congratulation, you found the hOwner

이동한 주소 주변에 수상한 값과 GetDlgItemInt함수가 보인다. 일단 함수에 대해 알아보고 이 함수가 어떤 역할을 하는지 확인해보자.

#GetDlgItemInt 함수

//대화상자의 지정된 컨트롤의 텍스트를 정수 값으로 변환하는 함수

UINT GetDlgItemInt(

HWND hDlg, //관심 컨트롤을 포함하는 대화 상자의 핸들

int nIDDlgItem, //텍스트를 번역할 컨트롤의 식별자

BOOL *lpTranslated, //성공 또는 실패를 나타낸다.(True는 성공.False는 실패 ,NULL 어떠한 정보도 반환하지 않음)

BOOL bSigned //함수가 처음에 빼기 부호에 대한 텍스트를 검사하고 부호있는 정수 값을 찾으면 이를 반환해야하는지를 나타낸다.

);

#반환값

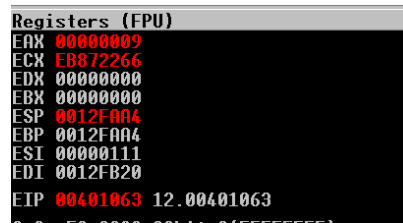
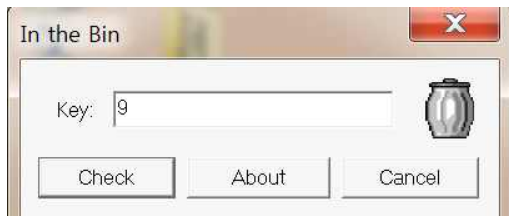
-함수가 성공하면 lpTranslated 가 가리키는 변수는 TRUE 로 설정되고 반환 값은 컨트롤 텍스트의 변환 된 값입니다.

-함수가 실패하면 lpTranslated 가 가리키는 변수는 FALSE 로 설정되고 반환 값은 0입니다. 0은 변환 가능한 값이므로 0을 반환하면 자체적으로 실패를 나타내지는 않습니다.

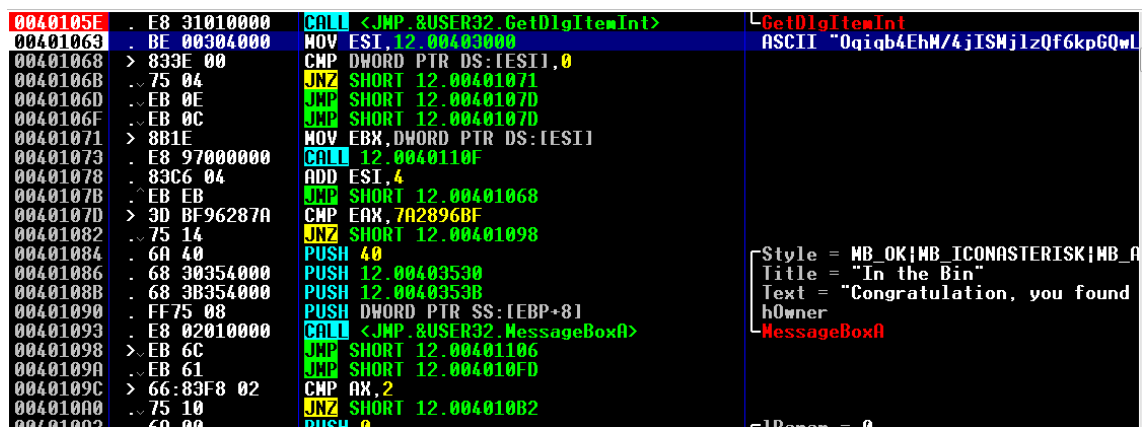
-경우, 그 주 bSigned의 매개 변수가 TRUE (-) 텍스트의 시작 부분 및 빼기 기호가 GetDlgItemInt는 부호있는 정수 값에 텍스트를 변환합니다. 그렇지 않으면 함수는 부호없는 정수 값을 만듭니다. 이 경우 적절한 값을 얻으려면 리턴 값을 int 유형으로 변환하십시오.

GetDlgItemInt함수를 이용해서 사용자가 입력한 값을 가져와 키 값과 비교하는 방식임을 지금까지의 문제풀이를 통해서 예측할 수 있다. 그러므로 GetDlgItemInt함수에 BP를 걸어 사용자가 입력값이 어디에 저장되는지 확인하면 보다 쉽게 키 값을 찾을 수 있을 것이다.

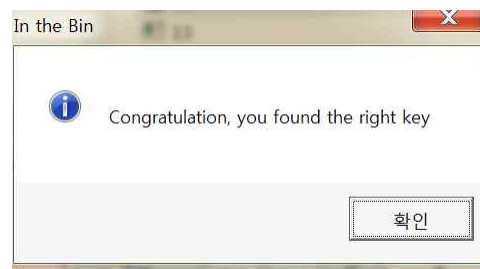
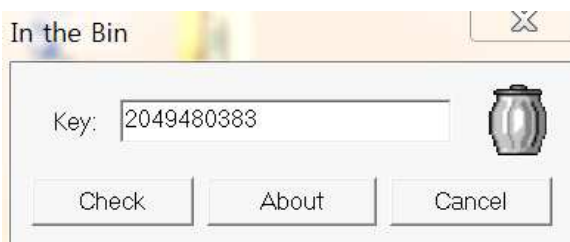
*JNZ 0 이 아니면 점프



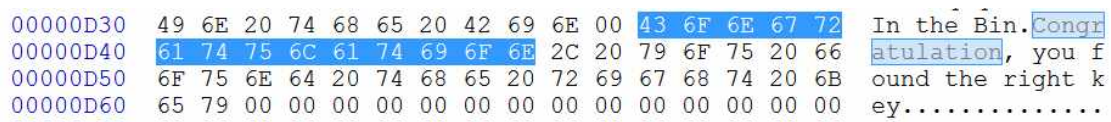
프로그램의 시리얼 입력칸에 임의의 값을 입력하고 실행시켜보니 EAX값에 저장되는 것을 확인 할 수 있다. 사용자 입력값이 EAX에 저장되는 것을 확인했으니 EAX와 시리얼 값을 비교해서 성공메시지를 출력하는 곳으로 분기하는 분기점을 찾아보자.



ESI 값에 이상한 값을 넣고 그 값에서 4 바이트씩 연산을 해서 EBX에 넣고 비교하는 패턴이다. 실행시켜보니 00401068~00401078을 반복하다가 결국 EAX값과 7A2896BF를 비교해서 분기하는 코드였다. 위의 알고리즘은 일종의 낚시 코드이고 결국 이 프로그램의 키 값은 7A2896B의 10 진수값인 2049480383이 된다. 이 값을 넣어 확인해보자.



키 값을 찾았으니 이제 성공메시지를 수정하기위해 HexEditor를 사용해 열어보도록 하자.



0D3B~0D62까지 문자열이 위치하고 있다. 이 주소에 키 값을 overwrite해서 수정해보자.

62	34	45	68	4F	71	69	71	00	00	00	00	78	56	34	12	b4EhOqĩq...xV4.
49	6E	20	74	68	65	20	42	69	6E	00	32	30	34	39	34	In the Bin.20494
38	30	33	38	33	00	00	00	00	00	00	00	00	00	00	00	80383.....
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

문제의 답은 20494803830D3B0D45

#키 값을 성공메시지대신 출력하는 방법에는 키값이 담긴 주소로 점프하게끔 할 수도 있다.