

# PE CODE INJECTION ASSIGNMENT

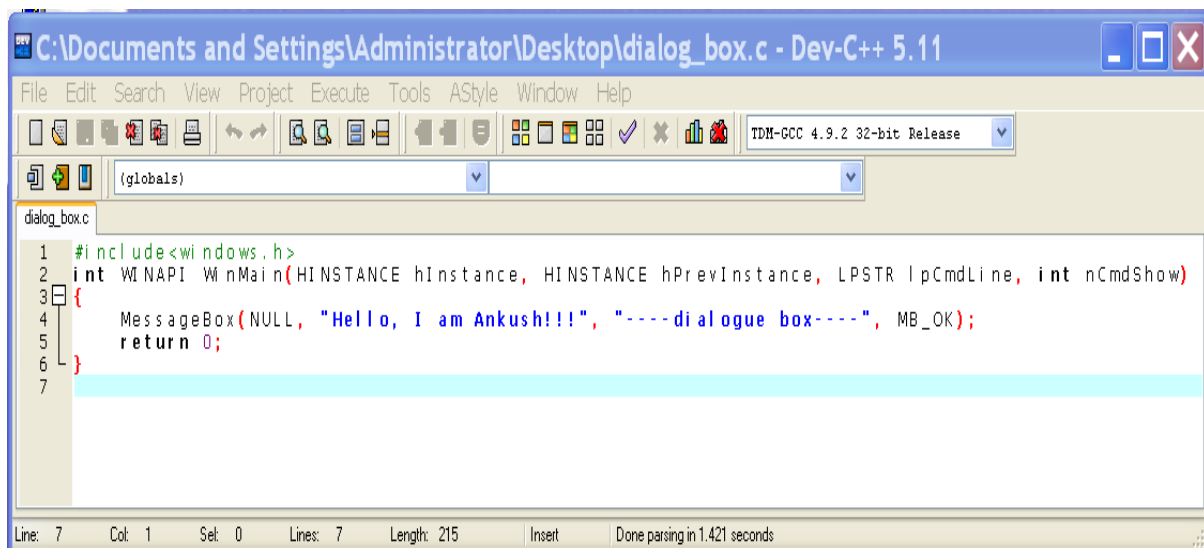
**Ankushdeep Singh**

**Ques:** Implement your own Windows code to display simple Hello Text in the body of the Application. Now hijack control of this program and inject MessageBox code into it, Caption should show your Name\_RollNumber and text in the message box should be "You have been Hacked". After the initial display of this MessageBox, retrieve back the original entry point. Create an answer document listing all the steps with appropriate screen shots.

**Ans:** I used the Lord PE, OLLYDBG, XVI32 for this PE Code Injection experiment and along with that I used “dev c++” to write my own small code that will display the message box and in the message box it shows the content as “Hello, I am Ankush!!!”. As said in the ques that to display the simple Hello Text in the body of the application so to complete this purpose I use the WinMain() function. The WinMain() function is basically the C entry point function of any windows application.

1. **Lord PE** – It is a tool for system programmers which is able to edit or view many parts of portable executable files.
2. **OLLYDBG** – It is basically a debugging tool which is used to analyze binary code.
3. **XVI32** – It is a freeware hexeditor.
4. **dev c++** – dev c++ is a complete IDE that for the C++ language.

This is my code.

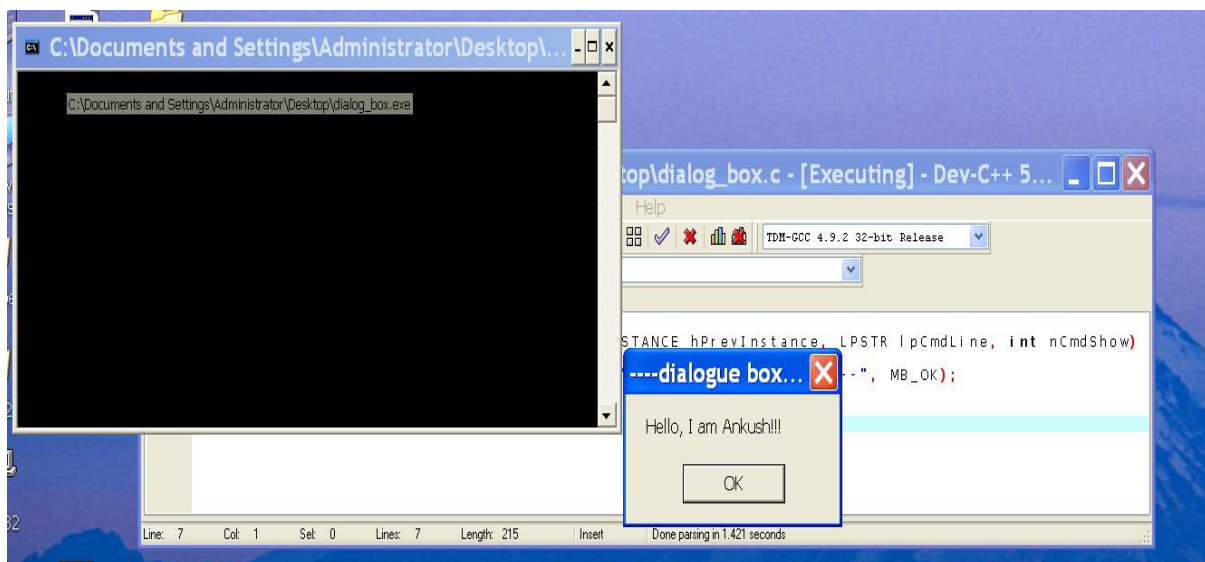


The screenshot shows the Dev-C++ 5.11 IDE with the file 'C:\Documents and Settings\Administrator\Desktop\dialog\_box.c'. The code is as follows:

```
1 #include<windows.h>
2 int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
3 {
4     MessageBox(NULL, "Hello, I am Ankush!!!", "----dialogue box----", MB_OK);
5     return 0;
6 }
7
```

The status bar at the bottom indicates 'Line: 7 Col: 1 Sel: 0 Lines: 7 Length: 215 Insert Done parsing in 1.421 seconds'.

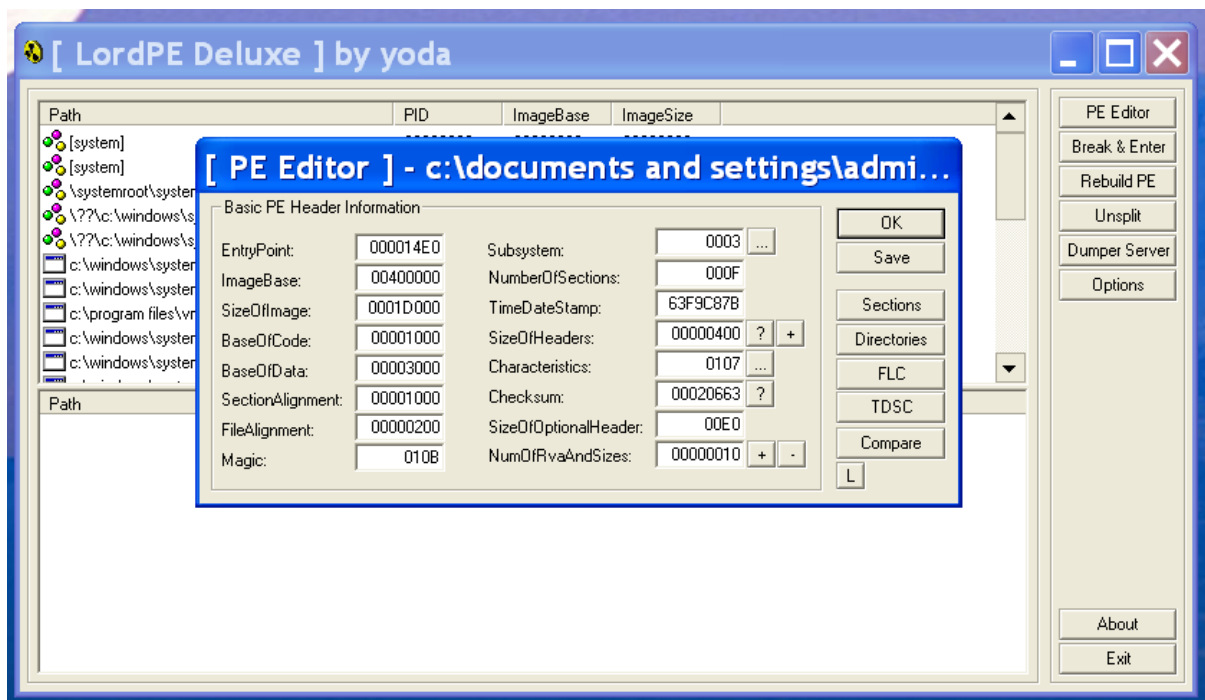
I ran the program to ensure that working as expected.



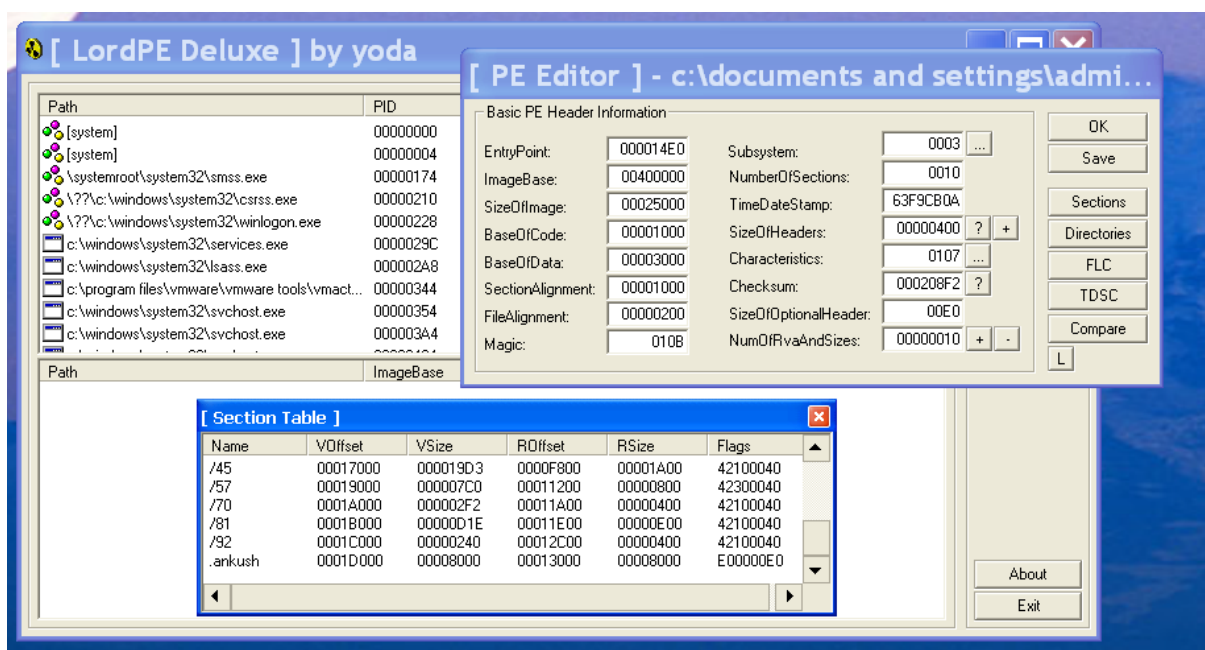
To implement the PE Code Injection I used the .exe file which is created by the “dialog\_box.c” file.



The first step is to add custom header section in the .exe file for the code cave. To add the header, I use the Lord PE editor.



I added my header called .ankush of 8,000 bytes.

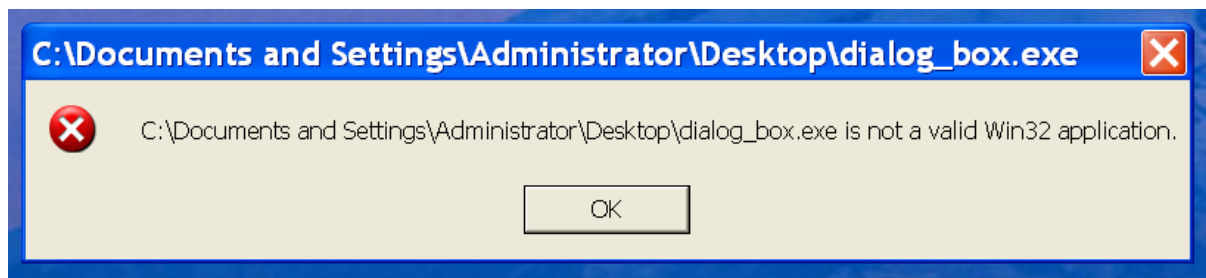


After adding the header, the “SizeOfImage” and NumberOfSections” are changed.

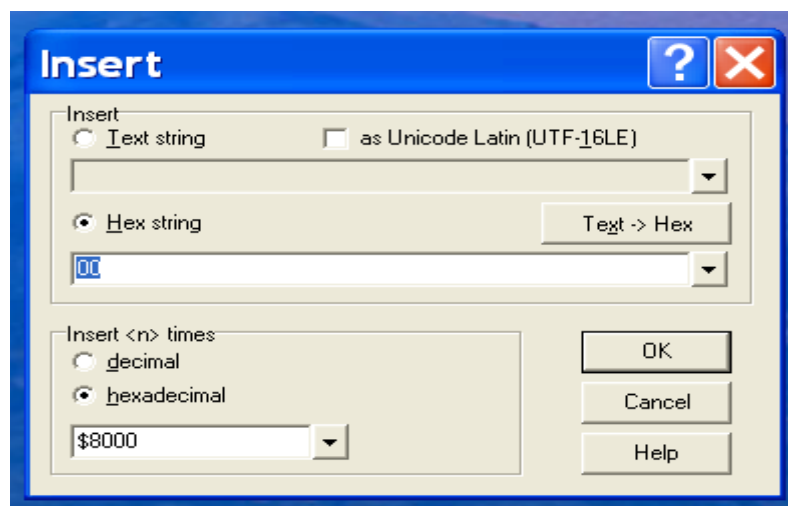
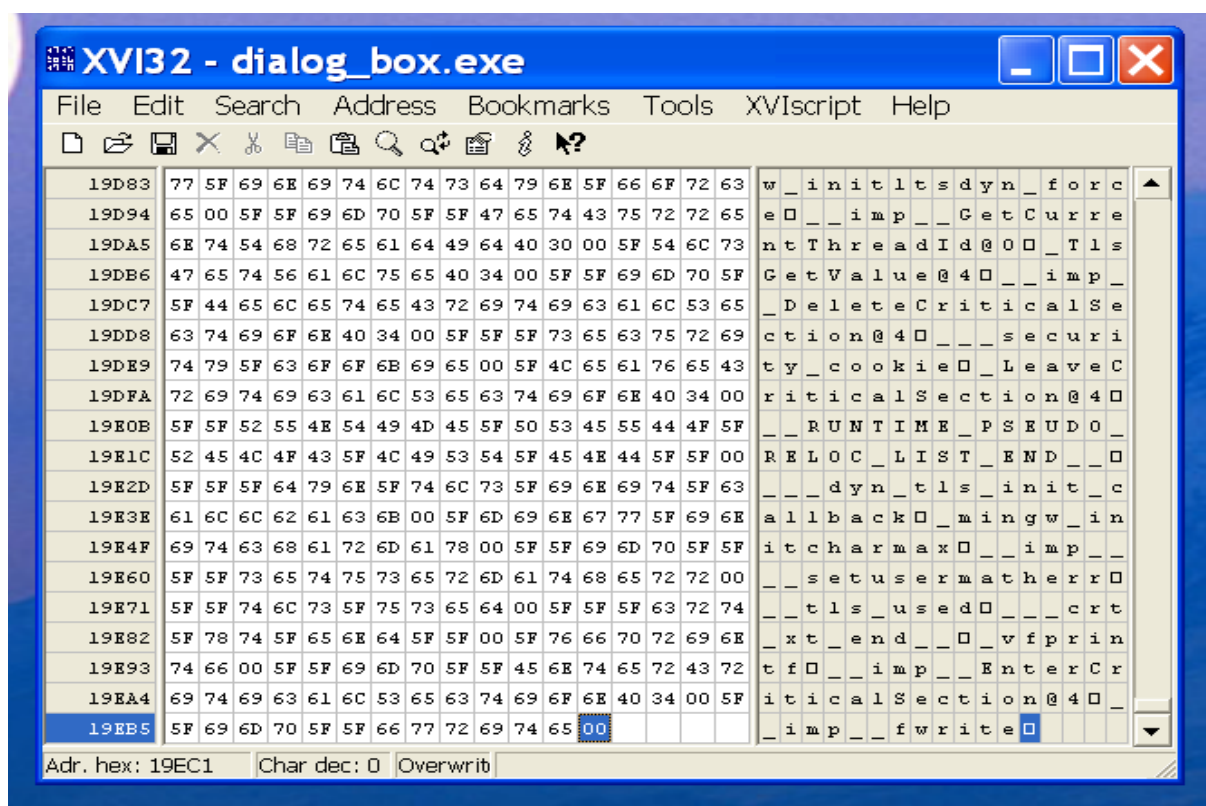
SizeOfImage = 0001D000 is changed to 00025000 after adding the header.

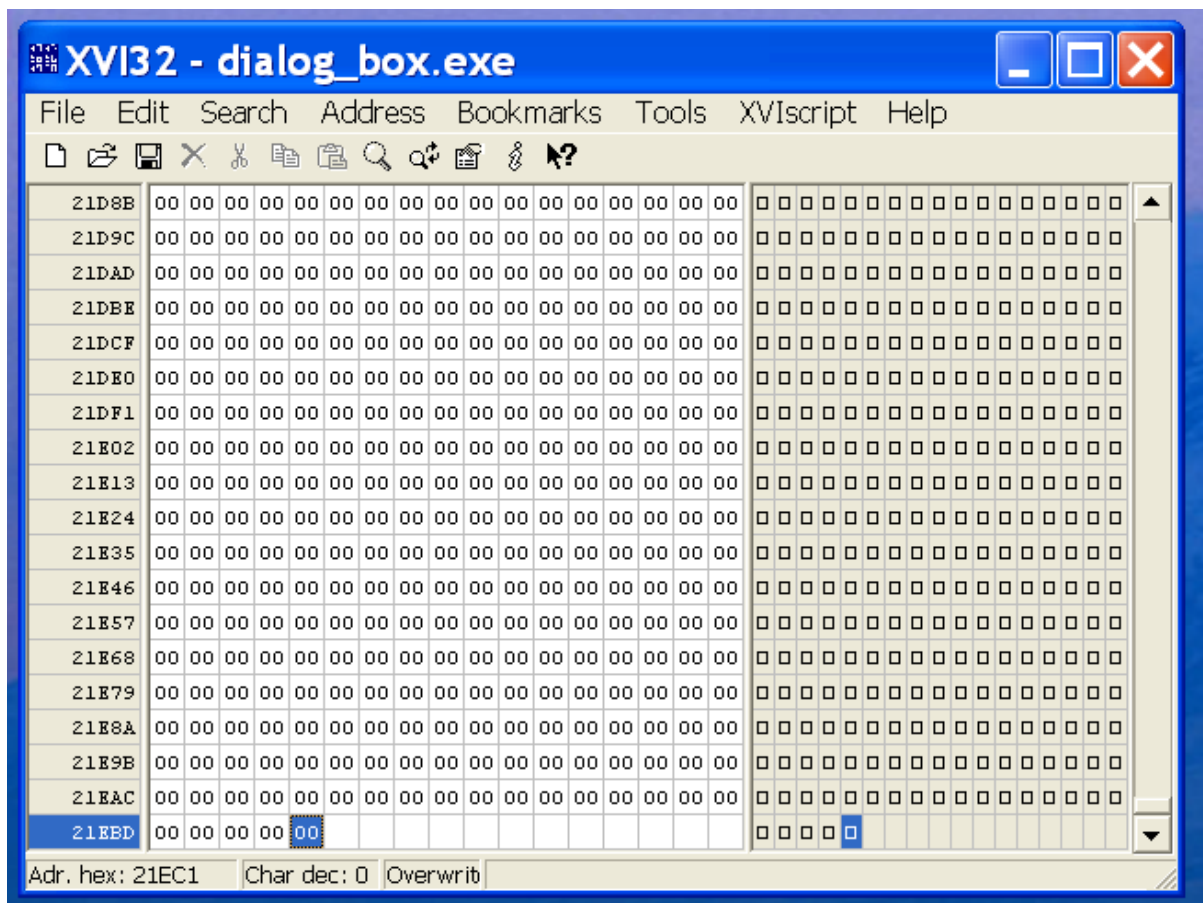
NumberOfSections = 000F is changed to 0010 after adding the header.

So, when I saved and ran the file and it cannot run due to size mismatch.

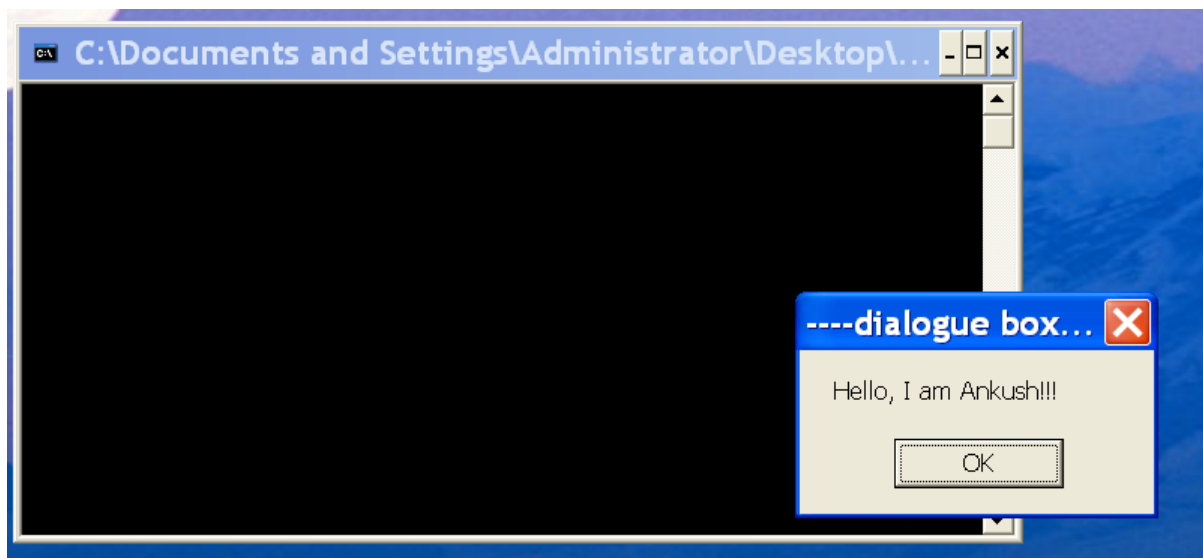


So, to run the .exe file successfully we need match the size of .exe file to the header. To match the size, we need to add the 8,000 bytes to the end of the .exe file, to do this I use XVI32 hexeditor.





After save this, and again run the .exe file to ensure that it is working or not.



Then I use the OLLYDBG to check the address of our code cave.

00400000	00001000	dialog_b	PE header	Imag	R	RWE	
00401000	00002000	dialog_b	code	Imag	R	RWE	
00403000	00001000	dialog_b	data	Imag	R	RWE	
00404000	00001000	dialog_b	.rdata	Imag	R	RWE	
00405000	00001000	dialog_b	.bss	Imag	R	RWE	
00406000	00001000	dialog_b	.idata	Imag	R	RWE	
00407000	00001000	dialog_b	.CRT	Imag	R	RWE	
00408000	00001000	dialog_b	.tls	Imag	R	RWE	
00409000	00001000	dialog_b	/4	Imag	R	RWE	
0040A000	0000B000	dialog_b	/19	Imag	R	RWE	
00415000	00002000	dialog_b	/31	Imag	R	RWE	
00417000	00002000	dialog_b	/45	Imag	R	RWE	
00419000	00001000	dialog_b	/57	Imag	R	RWE	
0041A000	00001000	dialog_b	/70	Imag	R	RWE	
0041B000	00001000	dialog_b	/81	Imag	R	RWE	
0041C000	00001000	dialog_b	/92	Imag	R	RWE	
0041D000	0000B000	dialog_b	.ankush	Imag	R	RWE	
00430000	00041000			Map	R	R	\\Device\\HarddiskVolume1\\WINDOWS\\system32\\sortkey.nls
00480000	00002000			Map	R E	R E	

The address of my code cave is “0041D000”.

After that I go to CPU main thread to copy and save the first few instructions to safe guard the original entry point of the program. Because I want to come back to the original program and run it.

```

*Untitled - Notepad
File Edit Format View Help
ORIGINAL ENTRY POINT:
|
004014E0 > $ 83EC 0C      SUB ESP,0C
004014E3 . C705 34504000 >MOV DWORD PTR DS:[405034],0
004014ED . E8 BE090000      CALL dialog_b.00401EB0
004014F2 . 83C4 0C          ADD ESP,0C
004014F5 . ^E9 86FCFFFF     JMP dialog_b.00401180
004014FA          90              NOP
004014FB          90              NOP

```

After saving the original entry point, we need to add our code cave address to the entry point.

004014E0	-E9 1BB0100	JMP dialog_b.0041D000
004014E5	90	NOP
004014E6	90	NOP
004014E7	90	NOP
004014E8	90	NOP
004014E9	90	NOP
004014EA	90	NOP
004014EB	90	NOP
004014EC	90	NOP
004014ED	. E8 BE090000	CALL dialog_b.00401EB0
004014F2	. 83C4 0C	ADD ESP,0C
004014F5	. ^E9 86FCFFFF	JMP dialog_b.00401180
004014FA	90	NOP
004014FB	90	NOP
004014FC	90	NOP
004014FD	90	NOP
004014FE	90	NOP
004014FF	90	NOP

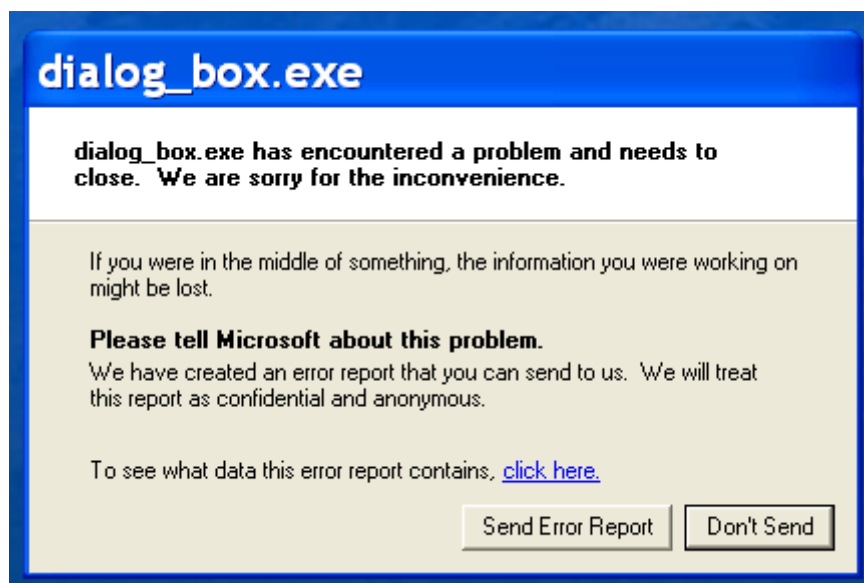
Then press the “fn+f7” to execute the instruction. The execution of this instruction moves us to the code cave address then select some portion to add 00’s. This is the portion where we added string which we want to display and also add the original entry point.



0041D000	0000	ADD BYTE PTR DS:[EAX],AL	
0041D002	0000	ADD BYTE PTR DS:[EAX],AL	
0041D004	0000	ADD BYTE PTR DS:[EAX],AL	
0041D006	0000	ADD BYTE PTR DS:[EAX],AL	
0041D008	0000	ADD BYTE PTR DS:[EAX],AL	
0041D00A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D00C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D00E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D010	0000	ADD BYTE PTR DS:[EAX],AL	
0041D012	0000	ADD BYTE PTR DS:[EAX],AL	
0041D014	0000	ADD BYTE PTR DS:[EAX],AL	
0041D016	0000	ADD BYTE PTR DS:[EAX],AL	
0041D018	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D020	0000	ADD BYTE PTR DS:[EAX],AL	
0041D022	0000	ADD BYTE PTR DS:[EAX],AL	
0041D024	0000	ADD BYTE PTR DS:[EAX],AL	
0041D026	0000	ADD BYTE PTR DS:[EAX],AL	
0041D028	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D030	0000	ADD BYTE PTR DS:[EAX],AL	
0041D032	0000	ADD BYTE PTR DS:[EAX],AL	
0041D034	0000	ADD BYTE PTR DS:[EAX],AL	
0041D036	0000	ADD BYTE PTR DS:[EAX],AL	
0041D038	0000	ADD BYTE PTR DS:[EAX],AL	
0041D03A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D03C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D03E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D040	0000	ADD BYTE PTR DS:[EAX],AL	
0041D042	0000	ADD BYTE PTR DS:[EAX],AL	
0041D044	0000	ADD BYTE PTR DS:[EAX],AL	

0041D114	0000	ADD BYTE PTR DS:[EAX],AL	
0041D116	0000	ADD BYTE PTR DS:[EAX],AL	
0041D118	0000	ADD BYTE PTR DS:[EAX],AL	
0041D11A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D11C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D11E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D120	0000	ADD BYTE PTR DS:[EAX],AL	
0041D122	0000	ADD BYTE PTR DS:[EAX],AL	
0041D124	0000	ADD BYTE PTR DS:[EAX],AL	
0041D126	0000	ADD BYTE PTR DS:[EAX],AL	
0041D128	0000	ADD BYTE PTR DS:[EAX],AL	
0041D12A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D12C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D12E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D130	0000	ADD BYTE PTR DS:[EAX],AL	

We selected the portion from “0041D00” to “0041D130” and fill it with 00’s. So, after saving and executing the .exe file it cannot runs.



Then go to our code cave address and select few spaces because we need to creates those strings and this strings should sits in the memory. The string we want to display is, caption should be your Name\_RollNumber and text in the message box should be “You have been Hacked”.

0041D000	6A 00	PUSH 0	
0041D002	68 4CD04100	PUSH dialog_b.0041D04C	ASCII "Ankushdeep Singh Panesar_102003174"
0041D007	68 2ED04100	PUSH dialog_b.0041D02E	ASCII "You Have Been Hacked.."
0041D00C	5A 00	PUSH 0	
0041D00E	E9 0737037E	CALL USER32.MessageBoxA	
0041D013	90	NOP	
0041D014	0000	ADD BYTE PTR DS:[EAX],AL	
0041D016	0000	ADD BYTE PTR DS:[EAX],AL	
0041D018	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D020	0000	ADD BYTE PTR DS:[EAX],AL	
0041D022	0000	ADD BYTE PTR DS:[EAX],AL	
0041D024	0000	ADD BYTE PTR DS:[EAX],AL	
0041D026	0000	ADD BYTE PTR DS:[EAX],AL	
0041D028	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02E	59	POP ECX	
0041D02F	6F	OUTS DX,DWORD PTR ES:[EDI]	I/O command
0041D030	75 20	JNZ SHORT dialog_b.0041D052	
0041D032	48	DEC EAX	
0041D033	61	POPAD	
0041D034	76 65	JBE SHORT dialog_b.0041D09B	
0041D036	2042 65	AND BYTE PTR DS:[EDX+65],AL	
0041D039	65:6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D03B	2048 61	AND BYTE PTR DS:[EAX+61],CL	
0041D03E	6368 65	ARPL WORD PTR DS:[EBX+65],BP	
0041D041	64:	PREFIX FS:	Superfluous prefix
0041D042	2E:	PREFIX CS:	Superfluous prefix
0041D043	2E:0000	ADD BYTE PTR CS:[EAX],AL	
0041D046	0000	ADD BYTE PTR DS:[EAX],AL	
0041D048	0000	ADD BYTE PTR DS:[EAX],AL	
0041D04A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D04C	41	INC ECX	
0041D04D	6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D04E	6B75 73 68	IMUL ESI,DWORD PTR SS:[EBP+73],68	
0041D052	64:	PREFIX FS:	Superfluous prefix
0041D053	65:	PREFIX GS:	Superfluous prefix
0041D054	65:70 20	JO SHORT dialog_b.0041D077	Superfluous prefix
0041D057	53	PUSH EBX	
0041D058	696E 67 6820506	IMUL EBP,DWORD PTR DS:[ESI+67],61502068	
0041D05F	6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D060	65:73 61	JNB SHORT dialog_b.0041D0C4	Superfluous prefix
0041D063	72 5F	JB SHORT dialog_b.0041D0C4	
0041D065	3130	XOR DWORD PTR DS:[EAX],ESI	
0041D067	3230	XOR DH,BYTE PTR DS:[EAX]	
0041D069	3033	XOR BYTE PTR DS:[EBX],DH	
0041D06B	3137	XOR DWORD PTR DS:[EDI],ESI	
0041D06D	34 00	XOR AL,0	
0041D06F	0000	ADD BYTE PTR DS:[EAX],AL	
0041D071	0000	ADD BYTE PTR DS:[EAX],AL	
0041D073	0000	ADD BYTE PTR DS:[EAX],AL	

From address “0041D000” to “0041D01E” there is a message box code.

From address “0041D02E” to “0041D043” there is a string called “You Have Been Hacked..”.

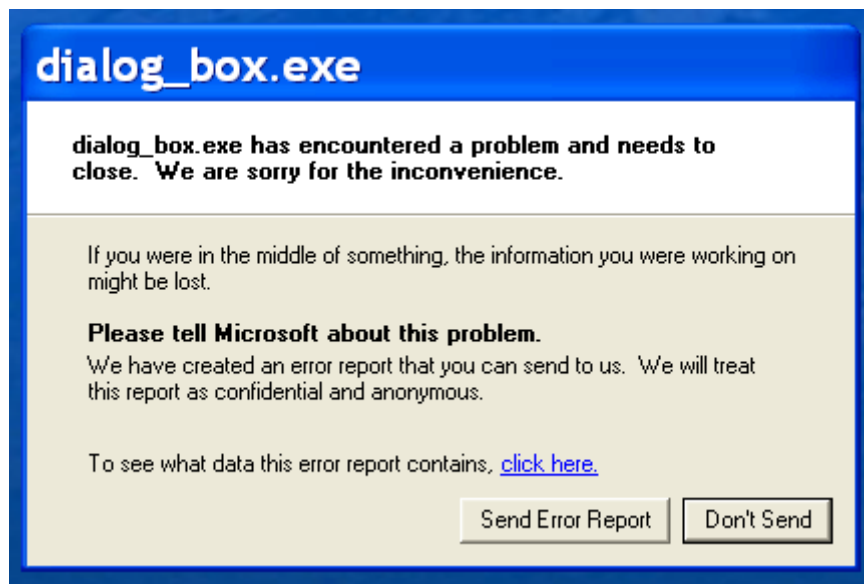
From address “0041D04C” to “0041D06D” there is a string called “Ankushdeep Singh Panesar\_102003174”.

After adding this and run the .exe we get this desired output.





After click on “ok” we get this



This window appears because we only add the string and dialogue box code. So, after clicking on “ok” the program has no instruction to execute and the program crashes.

So, for working of the .exe file properly we need to add the original entry point address.

0041D014	77 61	JMP SHORT dialog_b.0041D077	
0041D016	0000	ADD BYTE PTR DS:[EAX],AL	
0041D018	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D01E	0000	ADD BYTE PTR DS:[EAX],AL	
0041D020	0000	ADD BYTE PTR DS:[EAX],AL	
0041D022	0000	ADD BYTE PTR DS:[EAX],AL	
0041D024	0000	ADD BYTE PTR DS:[EAX],AL	
0041D026	0000	ADD BYTE PTR DS:[EAX],AL	
0041D028	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02C	0000	ADD BYTE PTR DS:[EAX],AL	
0041D02E	59	POP ECX	
0041D02F	6F	OUTS DX,DWORD PTR ES:[EDI]	I/O command
0041D030	75 20	JNZ SHORT dialog_b.0041D052	
0041D032	48	DEC EAX	
0041D033	61	POPAD	
0041D034	76 65	JBE SHORT dialog_b.0041D09B	
0041D036	2042 65	AND BYTE PTR DS:[EDI+65],AL	
0041D039	65:6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D03B	2048 61	AND BYTE PTR DS:[EAX+61],CL	
0041D03E	636B 65	ARPL WORD PTR DS:[EBX+65],BP	
0041D041	64:	PREFIX FS:	Superfluous prefix
0041D042	2E:	PREFIX CS:	Superfluous prefix
0041D043	2E:0000	ADD BYTE PTR CS:[EAX],AL	
0041D046	0000	ADD BYTE PTR DS:[EAX],AL	
0041D048	0000	ADD BYTE PTR DS:[EAX],AL	
0041D04A	0000	ADD BYTE PTR DS:[EAX],AL	
0041D04C	41	INC ECX	
0041D04D	6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D04E	6B75 73 68	IMUL ESI,DWORD PTR SS:[EBP+73],68	
0041D052	64:	PREFIX FS:	Superfluous prefix
0041D053	65:	PREFIX GS:	Superfluous prefix
0041D054	65:70 20	JQ SHORT dialog_b.0041D077	Superfluous prefix
0041D057	53	PUSH EBX	
0041D058	696E 67 6820506	IMUL EBP,DWORD PTR DS:[ESI+67],61502068	
0041D05F	6E	OUTS DX,BYTE PTR ES:[EDI]	I/O command
0041D060	65:73 61	JNB SHORT dialog_b.0041D0C4	Superfluous prefix
0041D063	72 5F	JB SHORT dialog_b.0041D0C4	
0041D065	3130	XOR DWORD PTR DS:[EAX],ESI	
0041D067	3230	XOR DH,BYTE PTR DS:[EAX]	
0041D069	3033	XOR BYTE PTR DS:[EBX],DH	
0041D06B	3137	XOR DWORD PTR DS:[EDI],ESI	
0041D06D	34 00	XOR AL,0	
0041D06F	0000	ADD BYTE PTR DS:[EAX],AL	
0041D071	0000	ADD BYTE PTR DS:[EAX],AL	
0041D073	0000	ADD BYTE PTR DS:[EAX],AL	
0041D075	0000	ADD BYTE PTR DS:[EAX],AL	
0041D077	E8 344EFEFF	CALL dialog_b.00401EB0	
0041D07C	E9 7144FEFF	JMP dialog_b.004014F2	
0041D081	0000	ADD BYTE PTR DS:[EAX],AL	
0041D083	0000	ADD BYTE PTR DS:[EAX],AL	

At address “0041D077” I add the original entry point address and at address “0041D07C” I add the next address of the original entry point.

At address “0041D014” I add the address where the original entry point address is called.

So, after saving and run the .exe file it runs properly.

First window:



After pressing “ok”:



After pressing “ok” the program will close.