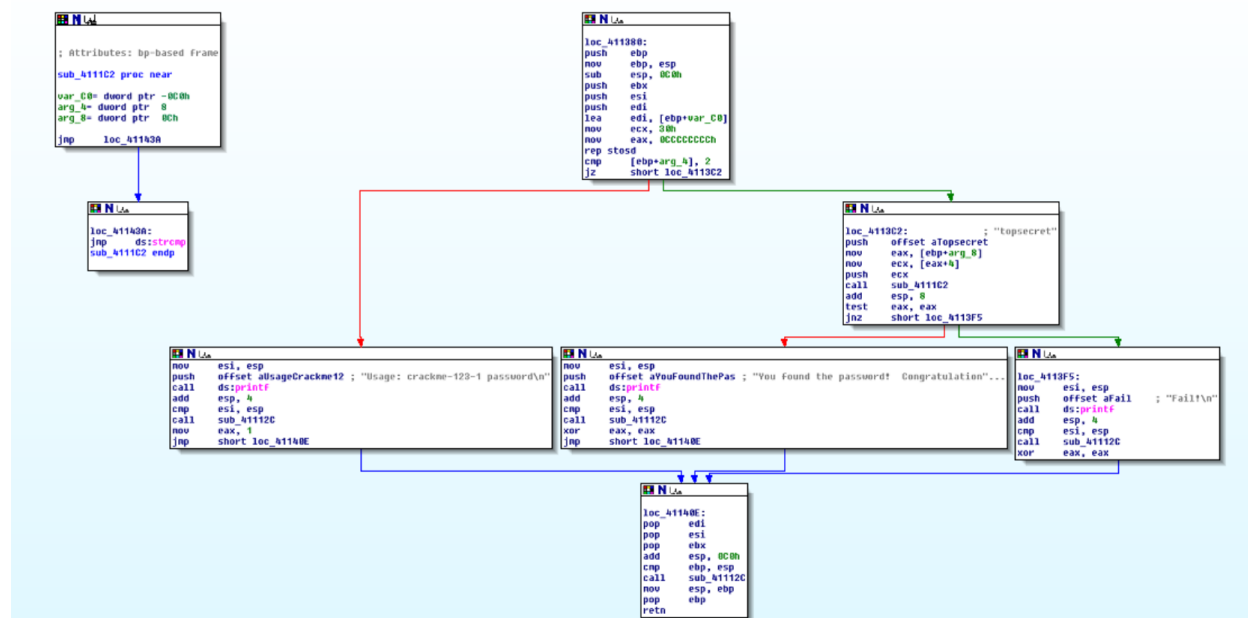


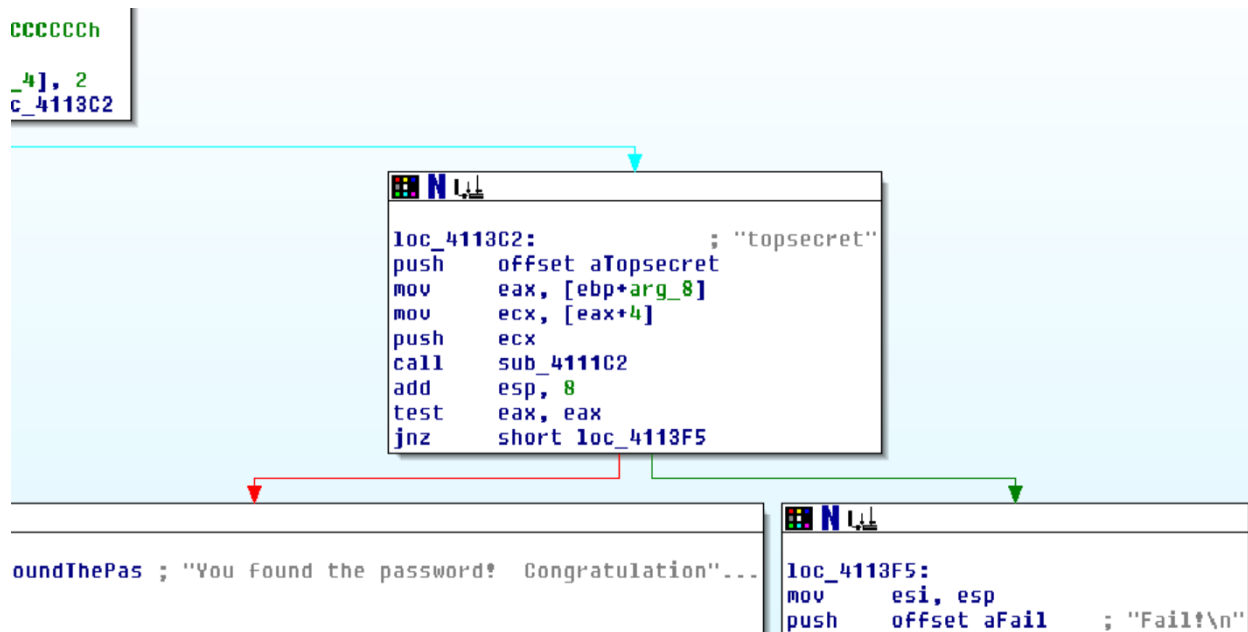
## LAB 15: Reverse Engineering with IDA Pro Freeware

### Viewing Disassembled Code

#### Crackme-121-1.exe:



Drag the "View-A" screen to show module "B", as shown below:



### Running the Executable

```
C:\Users\Lenovo legion 5\Downloads\test>crackme-121-1.exe
Usage: crackme-123-1 password

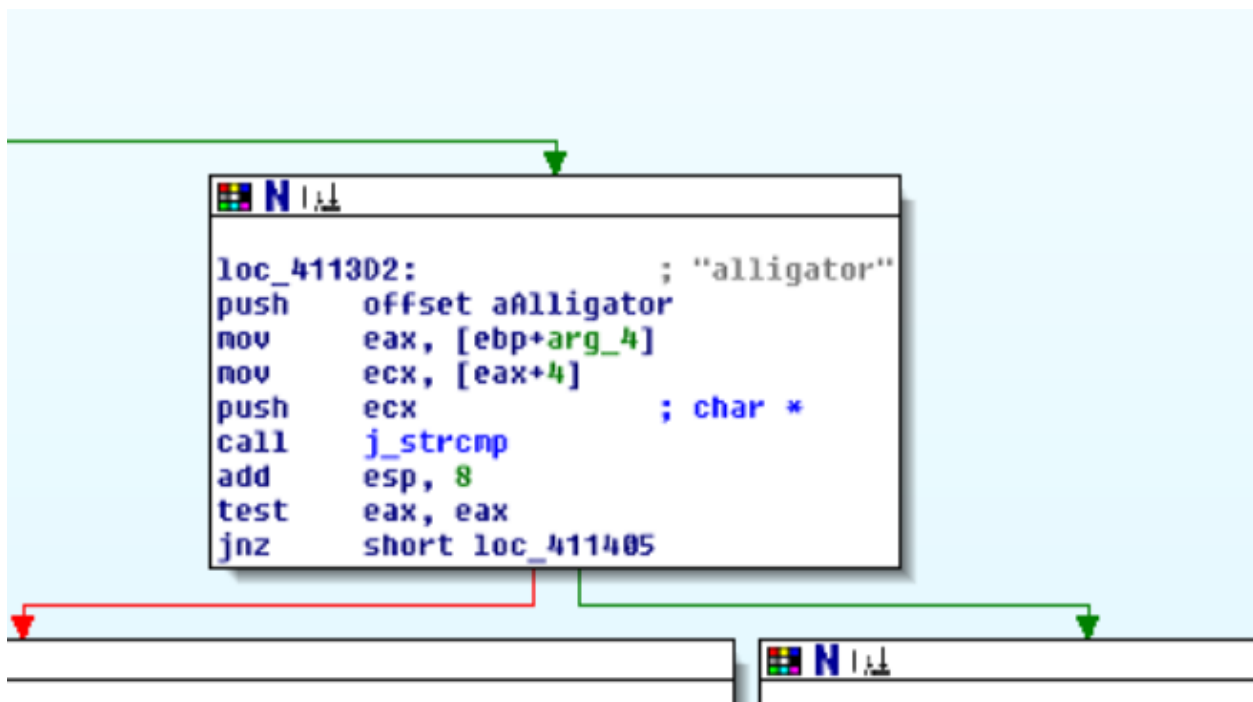
C:\Users\Lenovo legion 5\Downloads\test>crackme-121-1.exe wrongpassword
Fail!

C:\Users\Lenovo legion 5\Downloads\test>crackme-121-1.exe topsecret
You found the password! Congratulations!

C:\Users\Lenovo legion 5\Downloads\test>|
```

### Crackme-121-2.exe:

Load the executable in IDA Pro and Find the module containing the password, and save a screen capture of it:



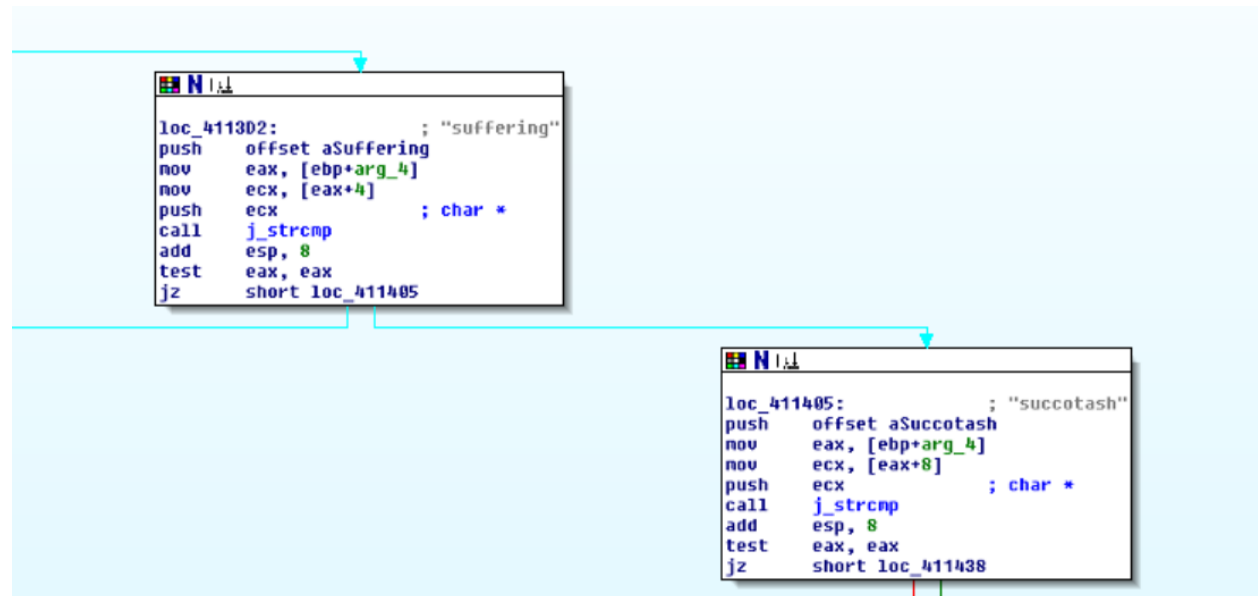
Run the program at a command prompt and save an image of it congratulating you for finding the password:

```
C:\Users\Lenovo legion 5\Downloads\test>crackme-121-2.exe alligator
You found the password! Congratulations!

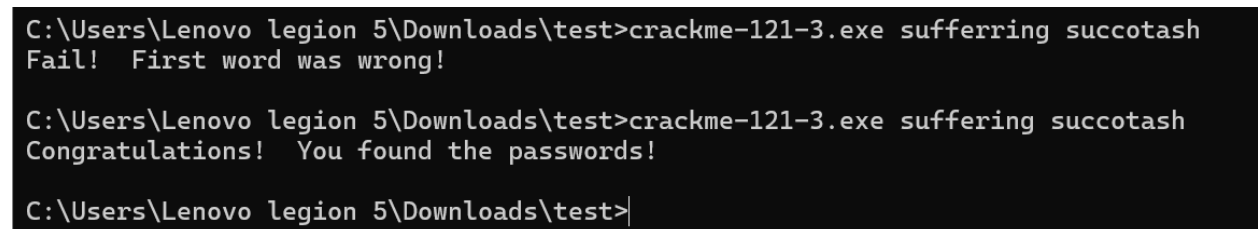
C:\Users\Lenovo legion 5\Downloads\test>|
```

### Crackme-121-3.exe:

Load the executable in IDA Pro and Find the modules containing the passwords, and save a screen capture of them:



Run the program at a command prompt and save an image of it congratulating you for finding the passwords:



**Crackme-121-4.exe:**

Load the executable in IDA Pro and Find the modules that perform string comparisons (strcmp) and try to guess what they are referring to:



Run the program at a command prompt and save an image of it congratulating you for solving the puzzle:

```
C:\Users\Lenovo legion 5\Downloads\test>game3.exe dromedary
Congratulations! You solved the crackme puzzle!

C:\Users\Lenovo legion 5\Downloads\test>|
```

**Note: because crackme-121-4.exe is not match the note of “use age” so I change file name to game3.exe and receive the successful result above.**

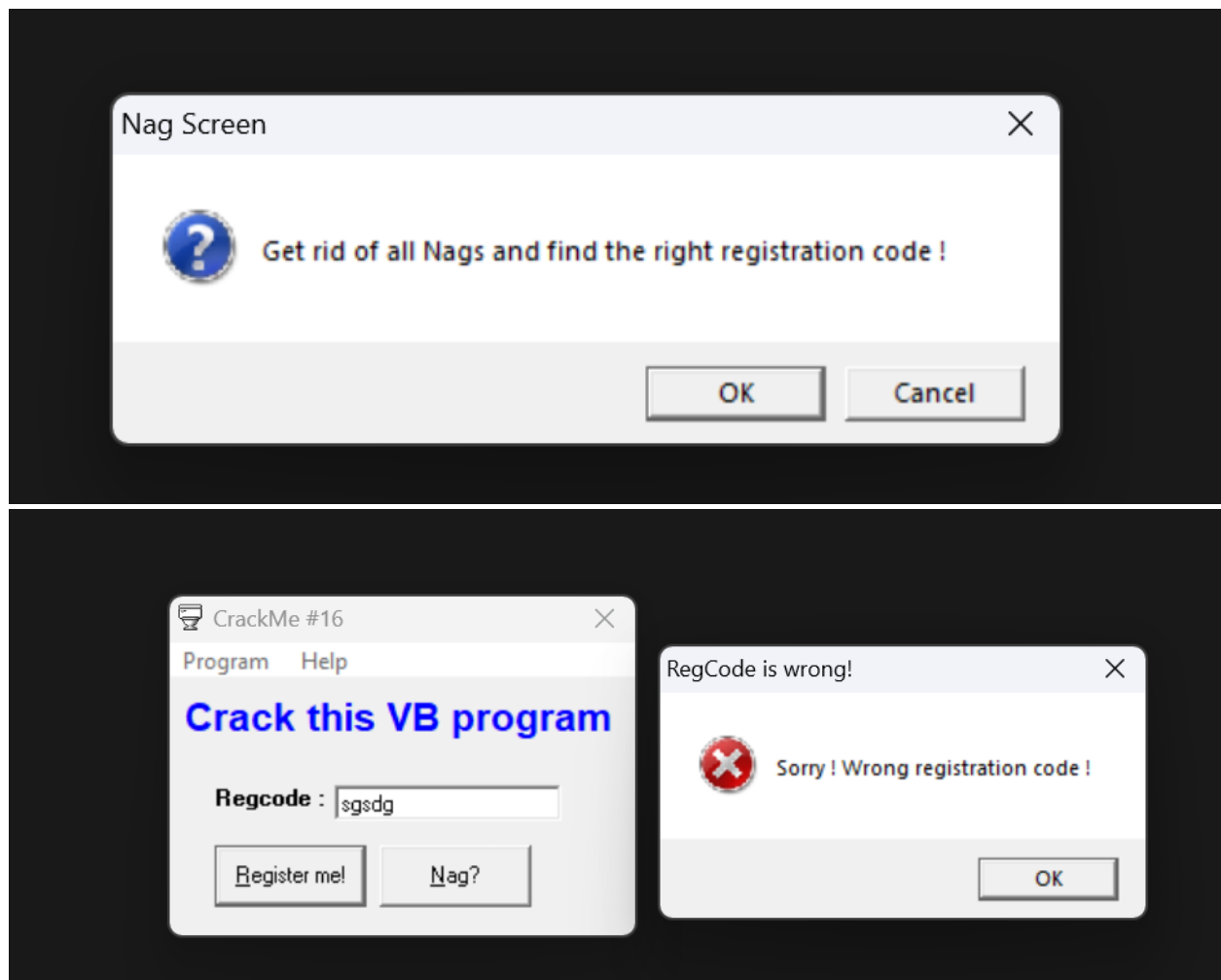
```
C:\Users\Lenovo legion 5\Downloads\test>crackme-121-4.exe
Usage: game3.exe password

C:\Users\Lenovo legion 5\Downloads\test>|
```

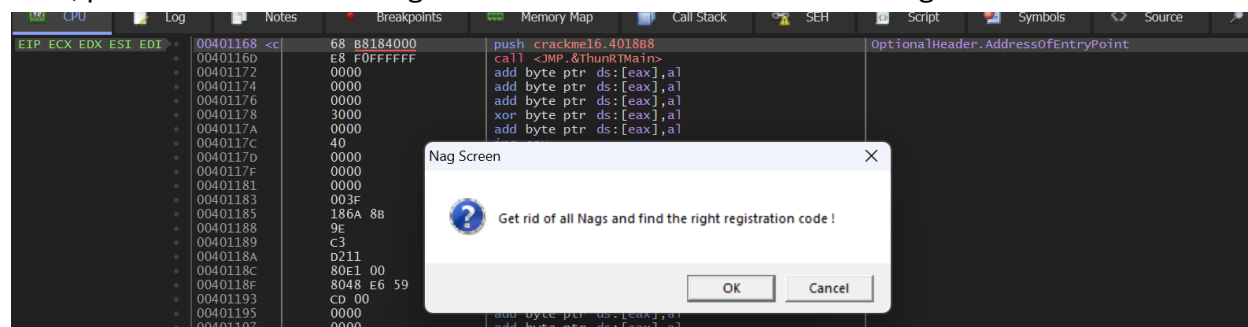
## CRACKME 16

This crackme is created with Visual Basic 5/6 which is prior to the .NET framework. Some programs out there are still written in it. So here's your opportunity to crack such a program. It is a mod of one of Lena's VB programs. Note that this crackme is compiled to native format. The p-code format will be covered in another crackme.

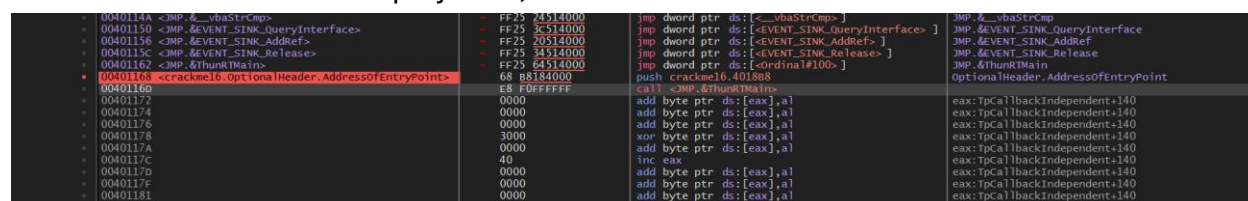
1. Get rid of the Nag screen which pops-up before the above window shows.
2. Crack the Regcode.



First, push the code to x32dbg and run to view the code call the Nag



Notice the Call <JMP.&ThuRTMain>, that line of code call stack to display the NAG screen, because when we run to display NAG, CPU reaches to this code.



View call stack to see the code call the NAG screen, crackme16.00402D03 is the stack part call the NAG screen:

0019F688	7689C126	7689D390	15C	System	user32.MessageBox+6F0
0019F7E4	7689C8CB	7689C126	94	System	user32.DrawState+1A56
0019F878	0F0AE19	7689C8CB	38	User	user32.MessageBoxIndirectA+EB
0019F8B0	0F0AE19	0F0AE19	24	User	msvbvm50.__vbaVarDateVar+4D78D
0019F8D4	0F0AEAF	0F0AE81	30	User	msvbvm50.__vbaVarDateVar+4D5F5
0019F904	0F0A6394	0F0AEFAF	6C	User	msvbvm50.__vbaVarDateVar+4D923
0019F970	0F0D414D	0F0A6394	74	User	msvbvm50.__vbaVarDateVar+44D08
0019F9E4	00402D03	0F0D414D	D4	User	msvbvm50.rtcMsgBox+F3
0019FAB8	0F01E5A9	00402D03	14	User	crackme16.00402D03
0019FACC	0F001AA3	0F01E5A9	8	User	msvbvm50.__vbaErase+2A0
0019FAD4	0F01E583	0F001AA3	1C	User	msvbvm50.0F001AA3
0019FAF0	0F02B781	0F01E583	3C	User	msvbvm50.__vbaErase+27A
0019FB2C	0F01FBC2	0F02B781	E0	User	msvbvm50.__vbaHResultCheckObj+296
0019FC0C	0F01FA4A	0F01FBC2	74	User	msvbvm50.__vbaFreeStrList+21A
0019FC80	0F029A57	0F01FA4A	24	User	msvbvm50.__vbaFreeStrList+A2
0019FCA4	0F0299E9	0F029A57	20	User	msvbvm50.__vbaStrVarMove+3786
0019FCC4	0F05B60C	0F0299E9	38	User	msvbvm50.__vbaStrVarMove+3748
0019FCFC	0F05B4EF	0F05B60C	44	User	msvbvm50.TipGetAddressOfPredeclaredInstance+BC
0019FD40	0F0120C4	0F05B4EF	154	User	msvbvm50.EbResetProjectNormal+246
0019FE94	0F00FF69	0F0120C4	1C	User	msvbvm50.EbLoadRunTime+DF3
0019FEB0	0F00A583	0F00FF69	260154	User	msvbvm50.EbSetContextWorkerThread+104D
00400004	00000000	0F00A583		User	msvbvm50.ThunRTMain+3C4

Click to this code to view detail, address 00402CFE executes to display NAG screen so I fill with nop in this line.

00402CFD	50	push eax	
00402CFE	E8 1DE4FFFF	call <JMP.&rtcMsgBox>	
00402D03	8D95 5CFFFFFF	lea edx,dword ptr ss:[ebp-A4]	
00402D09	8D4D BC	lea ecx,dword ptr ss:[ebp-44]	[ebp-44]:EndDialog+439
00402D0C	8985 64FFFFFF	mov dword ptr ss:[ebp-9C],eax	
00402D12	89BD 5CFFFFFF	mov dword ptr ss:[ebp-A4],edi	
00402D18	E8 09E4FFFF	call <JMP.&__vbaVarMove>	
00402D1D	8D45 8C	lea eax,dword ptr ss:[ebp-74]	
00402CFA	8D45 AC	lea eax,dword ptr ss:[ebp-54]	
00402CFD	50	push eax	
00402CFE	90	nop	
00402CFF	90	nop	
00402D00	90	nop	
00402D01	90	nop	
00402D02	90	nop	
00402D03	8D95 5CFFFFFF	lea edx,dword ptr ss:[ebp-A4]	
00402D09	8D4D BC	lea ecx,dword ptr ss:[ebp-44]	
00402D0C	8985 64FFFFFF	mov dword ptr ss:[ebp-9C],eax	
00402D12	89BD 5CFFFFFF	mov dword ptr ss:[ebp-A4],edi	
00402D18	E8 09E4FFFF	call <JMP.&__vbaVarMove>	

Patch and run again to check, we have problem that the program quit immediately so i we continue to trace below this code to see what happen:

00402D44	8085 7CFFFFFF	lea eax,dword ptr ss:[ebp-84]	eax:TpCallBackIndependent+140
00402D4A	50	push eax	eax:TpCallBackIndependent+140
00402D4B	E8 BEE3FFFF	call <JMP.&__vbaVarTstEq>	
00402D50	66:85C0	test ax,ax	
00402D53	75 05	jne crackme16.402D5A	
00402D55	E8 AEE3FFFF	call <JMP.&__vbaEnd>	
00402D5A	8975 FC	mov dword ptr ss:[ebp-4],esi	
00402D5D	68 982D4000	push crackme16.402D98	
00402D62	E8 13	jmp crackme16.402D77	
00402D64	8D45 8C	lea eax,dword ptr ss:[ebp-74]	eax:TpCallBackIndependent+140

Below the previous NOP, we catch this part of code:

Test ax, ax

Jne crackme16.402D5A => this code not execute because, the previous test is equal.

Call <JMP.&\_\_vbaEnd> => this cause program closed immediately.

So I change to JNE to JMP to pass the Call below.

66:85C0	test ax,ax	
EB 05	jmp crackme16.402D5A	
E8 AEE3FFFF	call <JMP.&__vbaEnd>	
8975 FC	mov dword ptr ss:[ebp-4],esi	
68 982D4000	push crackme16.402D98	
EB 13	jmp crackme16.402D77	
8D45 8C	lea eax,dword ptr ss:[ebp-74]	

This is done of removing NAG.

To crack regcode, we find to string references to see the hint:

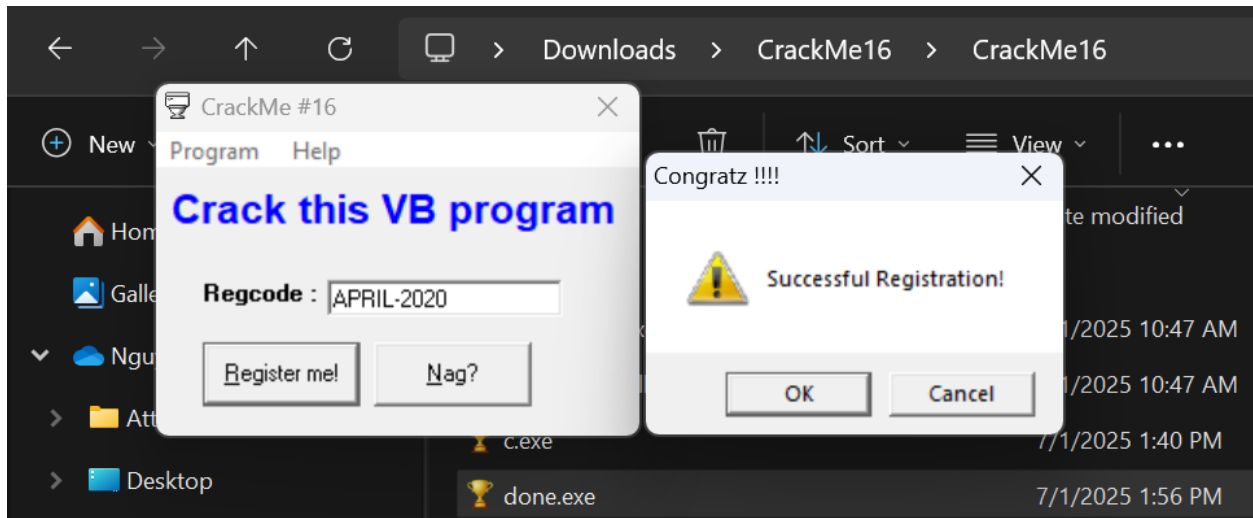
0040511C	&"QWP="
00050040	L"nsi-11-1-0api-ms-win-security-provider-11-1-0api-ms-win-security-sddl-a
00401DDC	L"APRIL-2020"
ackme16.401E08	L"Successful Registration!"
ackme16.401E50	L"Congratz !!!!"
00401DDC	L"APRIL-2020"
ackme16.401E70	L"Sorry ! Wrong registration code !"
ackme16.401EB8	L"RegCode is wrong!"
ackme16.401EF0	L"Get rid of all Nags and find the right registration code !"
ackme16.401F78	L"Nag Screen "
ackme16.401F94	L"by CrackingLessons.com"
00401F9C	L"visible"

We found that string "APRIL-2020", but let inspect to the code to view detail:

004028BD	68 DC1D4000	push crackme16.401DDC	401DDC:L"APRIL-2020"
004028C2	E8 83E8FFFF	call <JMP.&__vbaStrCmp>	
004028C7	8BF8	mov edi,eax	
004028C9	8D4D A8	lea ecx,dword ptr ss:[ebp-58]	
004028CC	F7DF	neg edi	
004028CE	1BFF	sbb edi,edi	
004028D0	47	inc edi	
004028D1	F7DF	neg edi	
004028D3	E8 60E8FFFF	call <JMP.&__vbaFreeStr>	
004028D8	8D4D A4	lea ecx,dword ptr ss:[ebp-5C]	
004028DB	E8 52E8FFFF	call <JMP.&__vbaFreeObj>	
004028E0	66:3BFE	cmp di,si	
004028E3	0F84 F3000000	je crackme16.4029DC	
004028E9	6A 08	push 8	
004028EB	8D95 74FFFFFF	lea edx,dword ptr ss:[ebp-8C]	
004028F1	5E	pop esi	
004028F2	8D4D AC	lea ecx,dword ptr ss:[ebp-54]	
004028F5	C785 7CFFFFFF 081E4000	mov dword ptr ss:[ebp-84],crackme16.401E08	401E08:L"Successful Registration!"
004028FF	89B5 74FFFFFF	mov dword ptr ss:[ebp-8C],esi	
00402905	E8 22E8FFFF	call <JMP.&__vbaVarCopy>	
0040290A	6A 03	push 3	
0040290C	8D95 74FFFFFF	lea edx,dword ptr ss:[ebp-8C]	
00402912	5B	pop ebx	
00402913	8D4D DC	lea ecx,dword ptr ss:[ebp-24]	
00402916	C785 7CFFFFFF 31000000	mov dword ptr ss:[ebp-84],31	31:'1'
00402920	899D 74FFFFFF	mov dword ptr ss:[ebp-8C],ebx	
00402926	E8 FBE7FFFF	call <JMP.&__vbaVarMove>	
0040292B	8D95 74FFFFFF	lea edx,dword ptr ss:[ebp-8C]	
00402931	8D4D CC	lea ecx,dword ptr ss:[ebp-34]	
00402934	C785 7CFFFFFF 501E4000	mov dword ptr ss:[ebp-84],crackme16.401E50	401E50:L"Congratz !!!!"
0040293E	89B5 74FFFFFF	mov dword ptr ss:[ebp-8C],esi	

Below the code of string APRIL-2020, program call the vbaStrCmp, that is the logic comparing our input with regcode (possibly APRIL-2020).

We try to patch the code and run to check:



DONE!!!