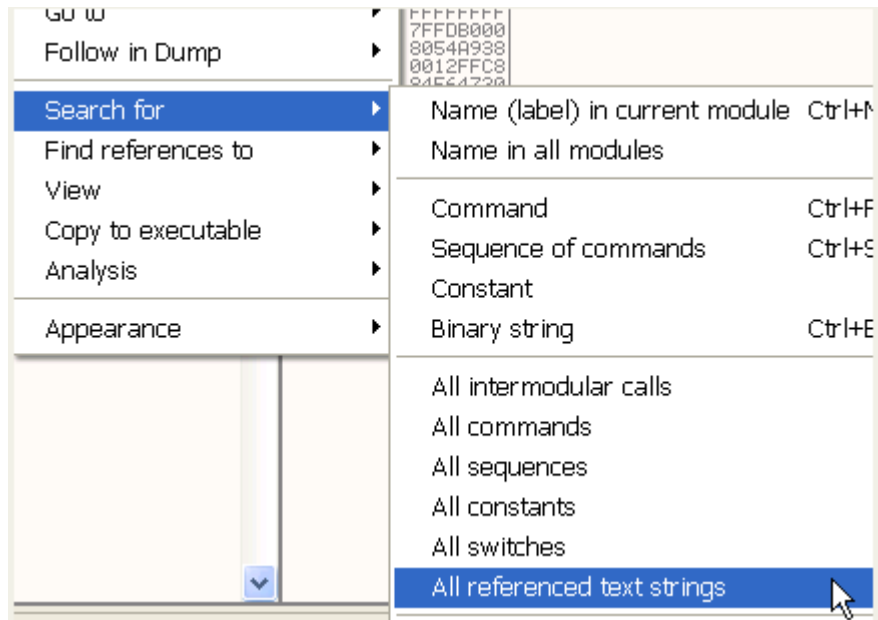


## 第十七章-序列号生成算法分析-Part2

首先我们来解决上一章留下的 mexcrk1.zip 这个 CrackMe,很简单。首先用 OD 加载它,断在入口点处。

Address	Disassembly	Comment
00420728	PUSH EBP	
00420729	MOV EBP,ESP	
0042072B	ADD ESP,-0C	
0042072E	MOV EAX,CRACK1.00420640	
00420733	CALL CRACK1.00404F58	
00420738	MOV EAX,DWORD PTR DS:[42EA10]	
0042073D	MOV EAX,DWORD PTR DS:[EAX]	
0042073F	CALL CRACK1.00428800	
00420744	MOV ECX,DWORD PTR DS:[42EA88]	CRACK1.0042F748
0042074A	MOV EAX,DWORD PTR DS:[42EA10]	
0042074F	MOV EAX,DWORD PTR DS:[EAX]	
00420751	MOV EDX,DWORD PTR DS:[42D35C]	CRACK1.0042D39C
00420757	CALL CRACK1.00428818	
0042075C	MOV EAX,DWORD PTR DS:[42EA10]	
00420761	MOV EAX,DWORD PTR DS:[EAX]	
00420763	CALL CRACK1.004288A4	
00420768	CALL CRACK1.00403414	
0042076D	LEA EAX,DWORD PTR DS:[EAX]	
00420770	ADD BYTE PTR DS:[EAX],AL	
00420772	ADD BYTE PTR DS:[EAX],AL	
00420774	ADD BYTE PTR DS:[EAX],AL	

首先看看程序中使用了哪些字符串,单击鼠标右键:



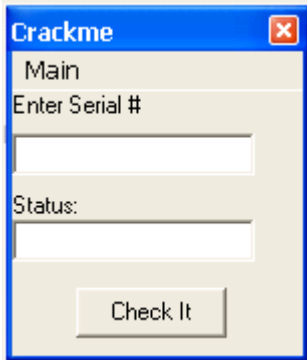
可以看到以下字符串。

Address	Disassembly	String
00420402	ASCII "TForm1"	
004204F6	ASCII "crackme"	
00420507	MOV EDX,CRACK1.0042D590	ASCII "Benadryl"
00420543	MOV EDX,CRACK1.0042D5A4	ASCII "Wrong Code DUDE"
00420555	MOV EDX,CRACK1.0042D5BC	ASCII "Thanks you made it"
00420590	ASCII "Benadryl",0	
004205A4	ASCII "Wrong Code DUDE",0	

我们可以在正确字符串上面或者错误字符串上面双击来关键代码附近。

Address	Disassembly	String
00420526	LEA EDX,DWORD PTR SS:[EBP-4]	
00420529	MOV EAX,DWORD PTR DS:[EBX+10C]	
0042052F	CALL CRACK1.0041A188	
00420534	MOV EAX,DWORD PTR SS:[EBP-4]	ASCII "Benadryl"
00420537	MOV EDX,CRACK1.0042D590	
0042053C	CALL CRACK1.00403800	
00420541	JE SHORT CRACK1.0042D555	
00420543	MOV EDX,CRACK1.0042D5A4	ASCII "Wrong Code DUDE"
00420548	MOV EAX,DWORD PTR DS:[EBX+1E8]	
0042054E	CALL CRACK1.0041A188	
00420553	JMP SHORT CRACK1.0042D565	
00420555	MOV EDX,CRACK1.0042D5BC	ASCII "Thanks you made it"
0042055A	MOV EAX,DWORD PTR DS:[EBX+1E8]	
00420560	CALL CRACK1.0041A188	
00420565	XOR EAX,EAX	
00420567	POP EDX	

我们看到这里有一个 CALL 指令决定后面的 JE 指令是跳转到”Thanks you made it”代码处,还是直接显示后面的”Wrong Code DUDE”。我们在 42D534 地址处设置一个断点,然后运行起来。



我们在 Enter Serial#下面的编辑框中随便输入一个序列号。



这里我输入 Narvajita。

0042D52F	. 8B83 00010000	MOV EAX,DWORD PTR DS:[EBX+100]	
0042D52F	. E8 54CCFEFF	CALL CRACK1.0041A188	
0042D534	. 8B45 FC	MOV EAX,DWORD PTR SS:[EBP-4]	
0042D537	. BA 90D54200	MOV EDI,CRACK1.0042D590	ASCII "Benadryl"
0042D53C	. E8 8F63FDFF	CALL CRACK1.004038D0	
0042D541	. 74 12	JE SHORT CRACK1.0042D555	
0042D543	. BA A4D54200	MOV EDI,CRACK1.0042D5A4	ASCII "Wrong Code DUDE"
0042D548	. 8B83 E8010000	MOV EAX,DWORD PTR DS:[EBX+1E8]	
0042D54E	. E8 65CCFEFF	CALL CRACK1.0041A188	
0042D553	. EB 10	JMP SHORT CRACK1.0042D565	
0042D555	. BA BCD54200	MOV EDI,CRACK1.0042D5BC	ASCII "Thanks you made it"
0042D55A	. 8B83 E8010000	MOV EAX,DWORD PTR DS:[EBX+1E8]	
0042D560	. E8 53CCFEFF	CALL CRACK1.0041A188	
0042D565	. 33C0	XOR EAX,EAX	
0042D567	. F3	DB EAX	

如果你跟进这个 CALL,你会发生时我们输入的错误序列号”Narvajita”在于”Benadryl”进行比较。

004038F9	> 8B0E	MOV ECX,DWORD PTR DS:[ESI]
004038FB	. 8B1F	MOV EBX,DWORD PTR DS:[EDI]
004038FD	. 39D9	CMP ECX,EBX
004038FF	. 75 58	JNZ SHORT CRACK1.00403959
EBP	0012F9B0	
ESI	00925948	ASCII "Narvajita"
EDI	0042D590	ASCII "Benadryl"
EIP	004038F9	CRACK1.004038F9

好了,我们现在达到了比较指令处,”Benadryl”就是正确的序列号,我们删除之前设置的断点,运行程序。

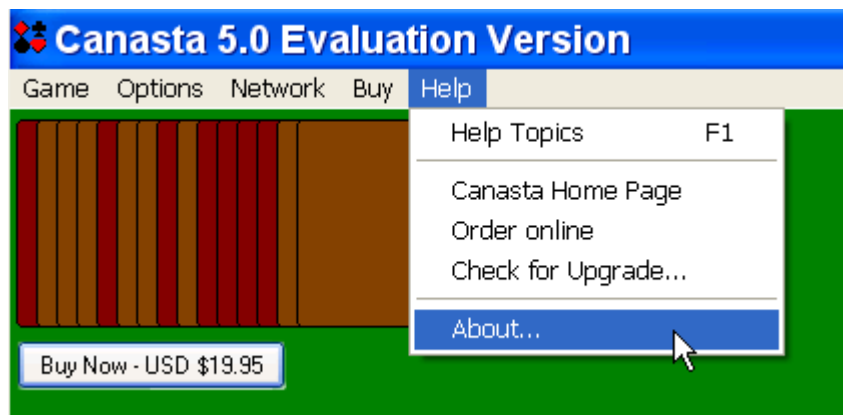


这里我们可以看到,显示”Thanks you made it”提示序列号正确的字样。

好了,接下来我们分析 Canasta 5.0 这个 CrackMe。

该 CrackMe 属于那种 OK 按钮开始不可用,当用户输入的用户名与序列号相匹配的时候才可以单击的例子。

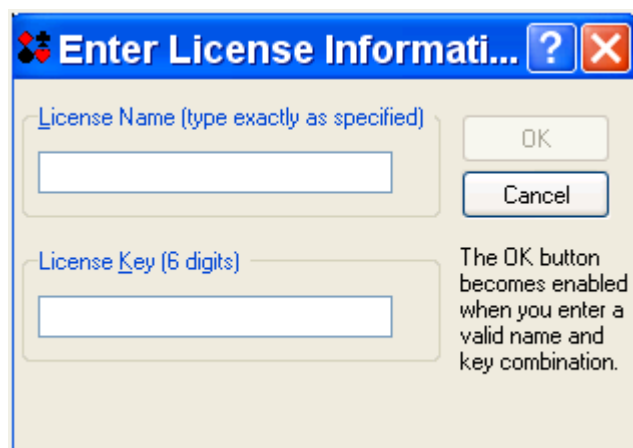
我们首先安装程序,然后打开 About 对话框。



提示 20 美元,确切是说是 19.95 美元。

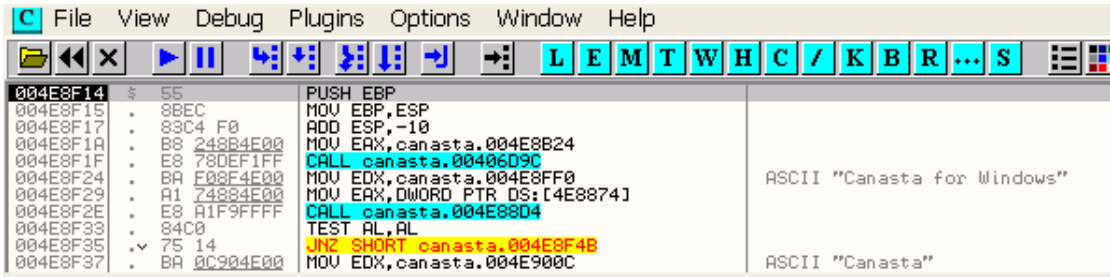


我们单击 Enter License 按钮。



提示说只有当用户输入的用户名和序列号组合正确时 OK 按钮才会变为可用。

用 OD 加载该程序。



我们断在了入口点处。

Address	Section	Type	Name	Comment
004F7850	.idata	Import	user32.ActivateKeyboardLayout	
004F784C	.idata	Import	user32.AdjustWindowRectEx	
004F7834	.idata	Import	user32.BeginDeferWindowPos	
004F7830	.idata	Import	user32.BeginPaint	
004F7598	.idata	Import	gdi32.BitBlt	
004F782C	.idata	Import	user32.CallNextHookEx	
004F7828	.idata	Import	user32.CallWindowProcA	
004F7840	.idata	Import	user32.CharLowerA	
004F783C	.idata	Import	user32.CharLowerBuffA	
004F7298	.idata	Import	user32.CharNextA	
004F7838	.idata	Import	user32.CharNextA	
004F7848	.idata	Import	user32.CharToOemA	
004F7844	.idata	Import	user32.CharUpperBuffA	
004F7824	.idata	Import	user32.CheckMenuItem	
004F7820	.idata	Import	user32.ChildWindowFromPoint	
004F7944	.idata	Import	comdlg32.ChooseColorA	
004F7940	.idata	Import	comdlg32.ChooseFontA	
004F781C	.idata	Import	user32.ClientToScreen	
004F7818	.idata	Import	user32.CloseClipboard	
004F7284	.idata	Import	kernel32.CloseHandle	
004F744C	.idata	Import	kernel32.CloseHandle	
004F7918	.idata	Import	winspool.ClosePrinter	
004F788C	.idata	Import	ole32.CoCreateInstance	
004F7894	.idata	Import	ole32.CoInitialize	
004F7448	.idata	Import	kernel32.CompareStringA	
004F7594	.idata	Import	gdi32.CopyEnhMetaFileA	
004F7888	.idata	Import	ole32.CoTaskMemAlloc	
004F7884	.idata	Import	ole32.CoTaskMemFree	
004F7890	.idata	Import	ole32.CoUninitialize	
004F7590	.idata	Import	gdi32.CreateBitmap	
004F758C	.idata	Import	gdi32.CreateBrushIndirect	
004F7588	.idata	Import	gdi32.CreateCompatibleBitmap	
004F7584	.idata	Import	gdi32.CreateCompatibleDC	
004F7580	.idata	Import	gdi32.CreateDCA	
004F7578	.idata	Import	gdi32.CreateDIBitmap	
004F757C	.idata	Import	gdi32.CreateDIBSection	
004F7444	.idata	Import	kernel32.CreateEventA	
004F7280	.idata	Import	kernel32.CreateFileA	
004F7440	.idata	Import	kernel32.CreateFileA	
004F7574	.idata	Import	gdi32.CreateFontIndirectA	
004F7570	.idata	Import	gdi32.CreateHalftonePalette	
004F756C	.idata	Import	gdi32.CreateIcon	

好,我们来看看 API 函数列表以及字符串列表。

Address	Disassembly	Text string
004B40E5	ASCII "als/RTD]"	
004B40F1	ASCII "Error"	
004B4130	DD canasta.004B417C	ASCII 00,"TLicense"
004B4150	DD canasta.004B417C	ASCII 00,"TLicense"
004B417D	ASCII "TLicense"	
004B4188	DD canasta.004B41D4	ASCII 0F,"TLicenseManager"
004B41A8	DD canasta.004B41D4	ASCII 0F,"TLicenseManager"
004B41D5	ASCII "TLicenseManager"	
004B4204	MOV ESI,canasta.004B423C	ASCII 11,"Unregistered copy"
004B423D	ASCII "Unregistered cop"	
004B424D	ASCII "y"	
004B42A3	MOV EDX,canasta.004B4370	ASCII "Settings\"
004B42DF	MOV EDX,canasta.004B4370	ASCII "Settings\"
004B4370	ASCII "Settings\",0	
004B43CF	MOV EDX,canasta.004B446C	ASCII "Settings\"
004B4400	MOV EDX,canasta.004B446C	ASCII "Settings\"
004B446C	ASCII "Settings\",0	
004B4484	MOV ESI,canasta.004B44A4	ASCII 11,"Unregistered copy"
004B44A5	ASCII "Unregistered cop"	
004B44B5	ASCII "y"	
004B462D	MOV EDX,canasta.004B4690	ASCII "Unregistered copy"
004B4690	ASCII "Unregistered cop"	
004B46A0	ASCII "y",0	
004B46CF	MOV EDX,canasta.004B46E8	ASCII "Unregistered copy"
004B46E8	ASCII "Unregistered cop"	
004B46F8	ASCII "y",0	
004B47EB	MOV EDX,canasta.004B4800	ASCII "Settings\DateFormat"
004B4804	PUSH canasta.004B489C	ASCII "Registration Key"
004B4813	MOV ECX,canasta.004B48B8	ASCII "User Name"
004B4823	PUSH canasta.004B48CC	ASCII "CodePageEx"
004B4832	MOV ECX,canasta.004B48E0	ASCII "CodePage"
004B4880	ASCII "Settings\DateFor"	
004B4890	ASCII "mat",0	
004B489C	ASCII "Registration Key"	
004B48AC	ASCII 0	
004B48B8	ASCII "User Name",0	
004B48CC	ASCII "CodePageEx",0	
004B48E0	ASCII "CodePage",0	
004B493E	MOV EDX,canasta.004B49C4	ASCII "Settings\DateFormat"
004B495C	PUSH canasta.004B49E0	(Initial CPU selection)
004B496B	MOV ECX,canasta.004B49FC	ASCII "User Name"
004B497B	PUSH canasta.004B4A10	ASCII "CodePageEx"
004B498A	MOV ECX,canasta.004B4A24	ASCII "CodePage"
004B49C4	ASCII "Settings\DateFor"	
004B49D4	ASCII "mat",0	
004B49E0	ASCII "Registration Key"	
004B49F0	ASCII 0	
004B49FC	ASCII "User Name",0	
004B4A10	ASCII "CodePageEx",0	
004B4A24	ASCII "CodePage",0	
004B4A66	MOV EAX,canasta.004B4B24	ASCII "LICENSE INVALID"
004B4B24	ASCII "LICENSE INVALID",0	
004B4BB8	MOV EDX,canasta.004B4D70	ASCII "http://www.canasta.net/keychecker/"
004B4BE0	MOV EAX,canasta.004B4D9C	ASCII "000"
004B4C0A	MOV EAX,canasta.004B4DA8	ASCII "yyyy/mm/dd"
004B4C2D	MOV EAX,canasta.004B4DA8	ASCII "yyyy/mm/dd"
004B4C49	MOV EAX,canasta.004B4DBC	ASCII "app=%s&name=%s&postfix=%s&key=%s&instal
004B4D70	ASCII "ta.net/keychecke"	
004B4D80	ASCII "ta.net/keychecke"	

由于 OK 按钮不可用,所以我们不能通过单击 OK 按钮来获取错误提示是什么,嘿嘿。

**Enter License Information...** ? X

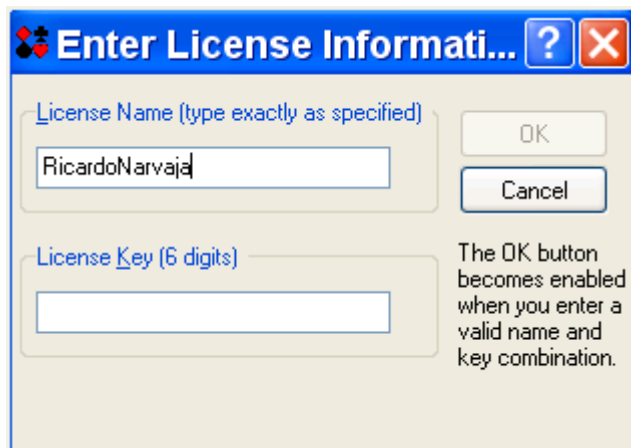
License Name (type exactly as specified)

License Key (6 digits)

The OK button becomes enabled when you enter a valid name and key combination.

OK Cancel

我们运行起来,然后打开注册窗口,我们运用一些方法来攻击这种保护。



**Enter License Information...**

License Name (type exactly as specified)

RicardoNarvaja

License Key (6 digits)

The OK button becomes enabled when you enter a valid name and key combination.

OK

Cancel

我们输入用户名,假设该程序对用户名没有限制,接在我们输入 6 个字母的序列号。例如:WMYXSZ。首先单击 W。



**Enter License Information...**

License Name (type exactly as specified)

RicardoNarvaja

License Key (6 digits)

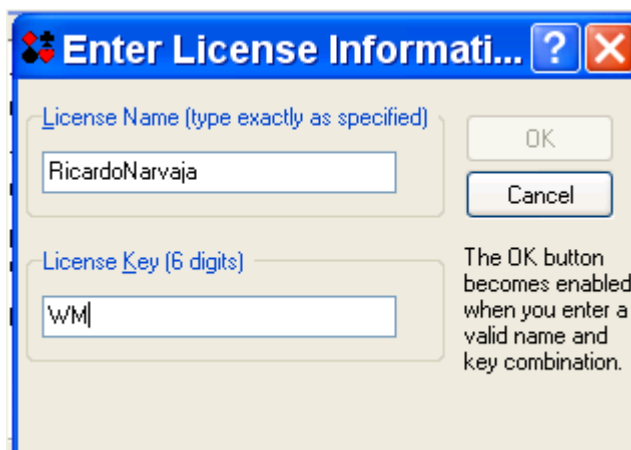
W

The OK button becomes enabled when you enter a valid name and key combination.

OK

Cancel

接着按 M 键。



**Enter License Information...**

License Name (type exactly as specified)

RicardoNarvaja

License Key (6 digits)

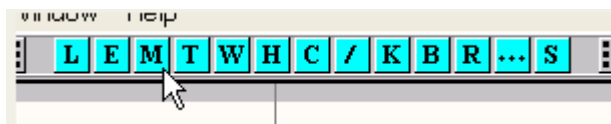
WM

The OK button becomes enabled when you enter a valid name and key combination.

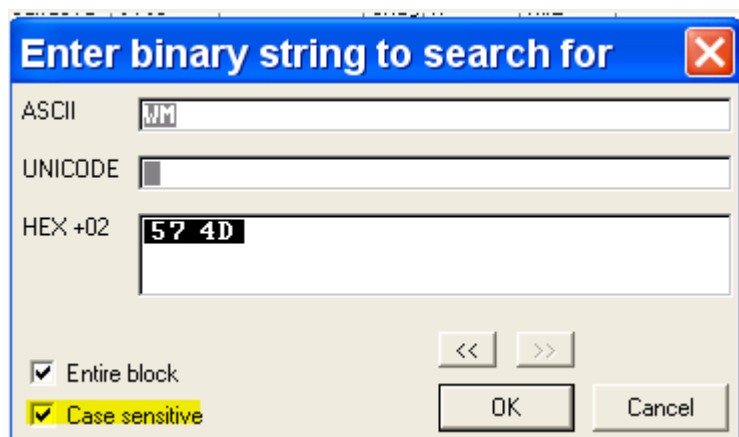
OK

Cancel

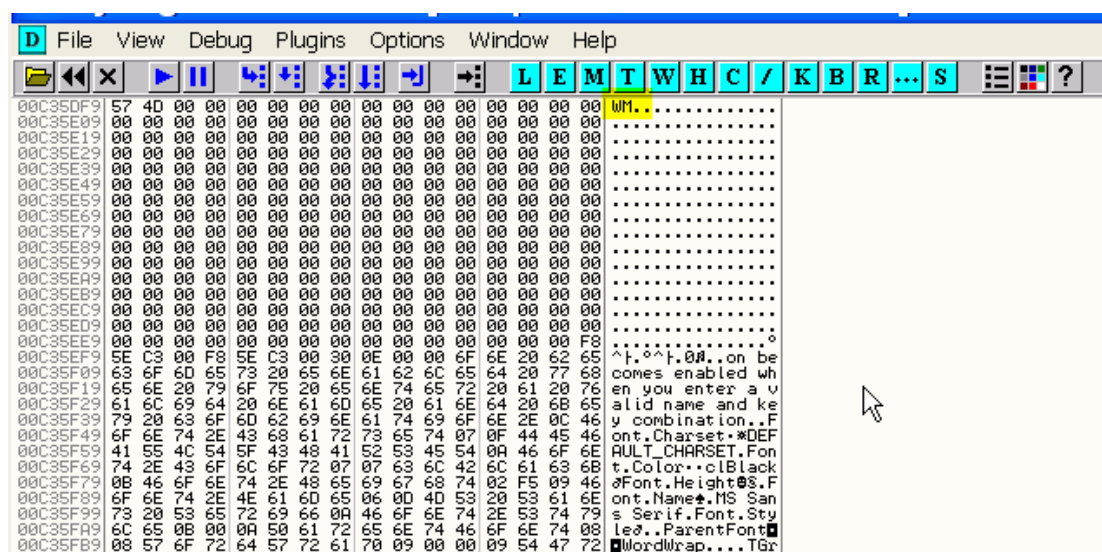
现在我们在内存中搜索刚刚输入的两个字母。



我们单击工具栏中的 M 按钮,打开内存窗口,查找 WM 字符串。

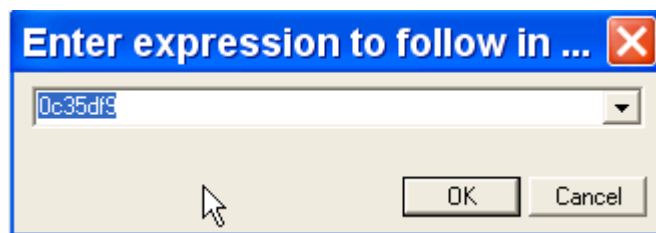


这里,我们勾选上区分大小写。



找到了一个,我们继续按 CTRL+L 查找看看还有没有其他地方有该字符串。提示未找到其他条目,说明这就是我们刚刚输入的序列号。

我们在数据窗口中定位到该字符串。



我们输入刚刚找到的这个地址,由于该地址是字母 c 开头,所以我们要在前面加上 0,不然 OD 不认识。

Address	Hex dump	ASCII
00C35DF9	57 4D 00 00 00 00 00 00	WM.....
00C35E01	00 00 00 00 00 00 00 00	.....
00C35E09	00 00 00 00 00 00 00 00	.....
00C35E11	00 00 00 00 00 00 00 00	.....
00C35E19	00 00 00 00 00 00 00 00	.....
00C35E21	00 00 00 00 00 00 00 00	.....
00C35E29	00 00 00 00 00 00 00 00	.....
00C35E31	00 00 00 00 00 00 00 00	.....
00C35E39	00 00 00 00 00 00 00 00	.....

可以用户输入的序列号就保存在这里,我们输入 “WMYXSZ”接下来一个字母 Y。





0040299E	. 78 11	US SHUKI canasta.004029B1	
004029A0	. FD	STD	
004029A1	. F3:A5	REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]	
004029A3	. 89C1	MOV ECX,EAX	
004029A5	. 83E1 03	AND ECX,3	
004029A8	. 83C6 03	ADD ESI,3	
004029AB	. 83C7 03	ADD EDI,3	
004029AE	. F3:A4	REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]	
004029B0	. FC	CLD	
004029B1	. 5F	POP EDI	
004029B2	. 5E	POP ESI	
004029B3	. C3	RETN	
004029B4	. 53	PUSH EBX	
004029B5	. 56	PUSH ESI	
004029B6	. 57	PUSH EDI	
004029B7	. 55	PUSH EBP	

现在将保存该字母了。

Registers (FPU)	
EAX	00000004
ECX	00000001
EDX	00C35DF9 ASCII "WMY"
EBX	00000004
ESP	0012E9A4
EBP	0012E908
ESI	00C13544 ASCII "WMYX"
EDI	00C35DF9 ASCII "WMY"
EIP	004029A1 canasta.004029A1
C 0	ES 0023 32bit 0(FFFFFFFF)
P 0	CS 001B 32bit 0(FFFFFFFF)

这里如果按 F8 键,ESI 将保存到 EDI 中。

Address	Hex dump	ASCII
00C35DF9	57 4D 59 58 00 00 00 00	WMYX...
00C35E01	00 00 00 00 00 00 00 00	.....
00C35E09	00 00 00 00 00 00 00 00	.....
00C35E11	00 00 00 00 00 00 00 00	.....

004B3E97	. 33C9	XOR ECX,ECX
004B3E99	. 8A0E	MOV CL,BYTE PTR DS:[ESI]
004B3E9B	. 41	INC ECX
004B3E9C	. F3:A4	REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
004B3E9E	. 8BF0	MOV ESI,EAX
004B3EA0	. 8DB0 FFFFFFFF	LEA EDI,DWORD PTR SS:[EBP-101]
004B3EA6	. 33C9	XOR ECX,ECX
004B3EA8	. 8A0E	MOV CL,BYTE PTR DS:[ESI]
004B3EAA	. 41	INC ECX

这里拷贝我们前面输入的 4 个字节。

Registers (FPU)	
EAX	00C35CF8 ASCII 0E,"RicardoNarva"
ECX	00000004
EDX	00C35DF8 ASCII 04,"WMYX"
EBX	00C15AE8 ASCII "HiE"
ESP	0012E690
EBP	0012E99C
ESI	00C35DF9 ASCII "WMYX"
EDI	0012E79C
EIP	004B3E9C canasta.004B3E9C
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)

按 F8 键,这 4 个字节将被拷贝到 12E79C 起始的内存单元中。

Address	Hex dump	ASCII
0012E79C	57 4D 59 58 41 3B 3C 77	WMYXA;<w
0012E7A4	73 B4 D1 77 B4 E7 12 00	s10wHf+
0012E7AC	0E 00 00 00 2C 00 00 00	.....
0012E7B4	BE 94 D1 77 98 87 D2 77	800w9cEw
0012E7BC	46 A8 11 A0 A0 A0 A0 A0	F0.....

既然我们已经设置过内存访问断点,那我们就来设置硬件访问断点。

Address	Hex dump	ASCII
0012E79C	57 4D 59 58	W Y 9 X
0012E7A4	73 B4 01 77	W B 01 W
0012E7AC	0E 00 00 00	
0012E7B4	BE 94 01 77	B E 94 W
0012E7BC	46 08 11 00	F 08 11
0012E7C4	05 00 00 00	
0012E7CC	EF 08 FF FF	E F 08
0012E7D4	01 00 00 00	
0012E7DC	E8 5A C1 00	E 8 5A C1
0012E7E4	05 E9 01 77	05 E9 01 W
0012E7EC	46 08 11 00	F 08 11
0012E7F4	05 00 00 00	
0012E7FC	01 00 00 00	
0012E804	24 CA 46 00	4 C A 46
0012E80C	46 08 11 00	F 08 11
0012E814	05 00 00 00	
0012E81C	5C E9 12 00	5 C E9 12
0012E824	E8 5A C1 00	E 8 5A C1
0012E82C	56 CB 45 00	5 6 C B 45
0012E834	88 E9 12 00	8 8 E9 12
0012E83C	C4 E9 12 00	C 4 E9 12
0012E844	E8 5A C1 00	E 8 5A C1
0012E84C	84 E8 12 00	8 4 E8 12
0012E854	A4 E9 12 00	A 4 E9 12
0012E85C	01 00 00 00	

004B3EA6	33C9	XOR ECX,ECX
004B3EA8	8A0E	MOV CL, BYTE PTR DS:[ESI]
004B3EAA	41	INC ECX
004B3EAB	F3:A4	REP MOVS BYTE PTR ES:[EDI], BYTE PTR DS:[ESI]
004B3EAD	80BD FFFFFFFF	CMP BYTE PTR SS:[EBP-101], 0
004B3EB4	0F84 F6000000	JE canasta.004B3FB0
004B3EBA	8D85 FFFFFFFF	LEA EAX, DWORD PTR SS:[EBP-101]

我们继续跟,就可以看到我们输入的用户名也被保存到了堆栈中。

Register	Value	Comment
EAX	00C35CF8	ASCII 0E, "RicardoNarvaja"
ECX	0000000F	
EDX	00C35DF8	ASCII 04, "WMVX"
EBX	00C15AE8	ASCII "HiE"
ESP	0012E690	
EBP	0012E99C	
ESI	00C35CF8	ASCII 0E, "RicardoNarvaja"
EDI	0012E898	
EIP	004B3EAB	canasta.004B3EAB

Address	Hex dump	ASCII
0012E898	0E 52 69 63 61 72 64 6F	Ricardo
0012E8A3	4E 61 72 76 61 6A 61 00	Narvaja.
0012E8AB	00 FF FF FF FF 3A EC 12	
0012E8B3	00 24 EA 12 00 F4 E9 12	
0012E8BB	00 0C 0D 01 77 B7 1A 01	

我们输入的用户名长度为 0E,与 0 做比较。

004B3EAB	F3:A4	REP MOVS BYTE PTR ES:[EDI], BYTE PTR DS
004B3EAD	80BD FFFFFFFF	CMP BYTE PTR SS:[EBP-101], 0
004B3EB4	0F84 F6000000	JE canasta.004B3FB0
004B3EBA	8D85 FFFFFFFF	LEA EAX, DWORD PTR SS:[EBP-101]

如果用户名长度不为零,程序将继续执行。

004B3EAB	F3A4	REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]	
004B3EAD	80BD FFFFFFFF	CMP BYTE PTR SS:[EBP-101],0	
004B3EB4	0F84 F0000000	JE canasta.004B3FB0	
004B3EB8	8D85 FFFFFFFF	LEA EAX,DWORD PTR SS:[EBP-101]	
004B3EC0	BA 5C404000	MOV EDI,canasta.004B405C	ASCII 03,"TNO"
004B3EC5	33C9	XOR ECX,ECX	
004B3EC7	8A08	MOV CL,BYTE PTR DS:[EAX]	
004B3EC9	41	INC ECX	
004B3ECA	E8 E9F2F4FF	CALL canasta.004031B8	
004B3ECF	0F84 D0000000	JE canasta.004B3FB0	
004B3ED5	8D85 FFFFFFFF	LEA EAX,DWORD PTR SS:[EBP-101]	
004B3ED8	BA 5C404000	MOV EDI,canasta.004B405C	ASCII 05,"afdad"
004B3EE0	33C9	XOR ECX,ECX	
004B3EE2	8A08	MOV CL,BYTE PTR DS:[EAX]	
004B3EE4	41	INC ECX	
004B3EE5	E8 CEF2F4FF	CALL canasta.004031B8	
004B3EEA	0F84 C0000000	JE canasta.004B3FB0	
004B3EF0	8D85 FFFFFFFF	LEA EAX,DWORD PTR SS:[EBP-101]	
004B3EF6	BA 5C404000	MOV EDI,canasta.004B405C	ASCII 05,"Gauss"
004B3EF8	33C9	XOR ECX,ECX	
004B3EFD	8A08	MOV CL,BYTE PTR DS:[EAX]	
004B3EFF	41	INC ECX	
004B3F00	E8 B3F2F4FF	CALL canasta.004031B8	
004B3F05	0F84 A0000000	JE canasta.004B3FB0	
004B3F08	8D85 FFFFFFFF	LEA EAX,DWORD PTR SS:[EBP-101]	
004B3F11	BA 5C404000	MOV EDI,canasta.004B4070	ASCII 16,"StArDoGg CHaMPioN PC97"
004B3F16	33C9	XOR ECX,ECX	
004B3F18	8A08	MOV CL,BYTE PTR DS:[EAX]	
004B3F1A	41	INC ECX	
004B3F1B	E8 98F2F4FF	CALL canasta.004031B8	

然后判断是不是黑名单中的名称,也就是说不能是 TNO,afdad 等,嘿嘿。

继续运行。

004031C0	> 74 26	JE SHORT canasta.004031E8
004031C2	> 8B08	MOV ECX,DWORD PTR DS:[EAX]
004031C4	> 8B1A	MOV EBX,DWORD PTR DS:[ECX]
004031C6	> 39D9	CMP ECX,EBX
004031C8	> 75 45	JNZ SHORT canasta.0040320F
004031CA	> 4E	DEC ESI

断在了比较指令处,我们可以看到是我们输入的序列号在与另一个字符串进行比较。

Registers (FPU)	
EAX	0012E79B ASCII 04,"WMVX"
ECX	594D5704
EDX	0012E69B ASCII 06,"354910"
EBX	00C15AE8 ASCII "HiE"
ESP	0012E680
EBP	0012E99C
ESI	00000001
EDI	0012E8AA
EIP	004031C4 canasta.004031C4
C 0	ES 0023 32bit 0(FFFFFFFF)
P 0	CS 001B 32bit 0(FFFFFFFF)
D 1	DS 0023 32bit 0(FFFFFFFF)

我们可以看到"354910"就是正确的序列号,我们删除所有断点。

Enter License Informati... ? X

License Name (type exactly as specified)

RicardoNarvaja

License Key (6 digits)

354910

OK

Cancel

The OK button becomes enabled when you enter a valid name and key combination.

输入正确的序列号以后,OK 按钮变为可用状态了,嘿嘿。两种解决方法这里就不再赘述了,也不是很复杂,就利用我们每按下一个键就发产生 WM\_KEYUP 消息,显得比较麻烦,但也很直接。

好了,我们休息一下,下一章,我们来看看别的例子。