OllyDBG 入门系列(二)--字串参考





OllyDbg专栏收录该内容

9篇文章3订阅 订阅专栏

上一篇是使用入门,现在我们开始正式进入破解。今天的目标程序是看雪兄《加密与解密》第一版附带光盘中 的 crackmes.cjb.net 镜像打包中的 CFF Crackme #3,采用用户名/序列号保护方式。原版加了个 UPX 的壳。刚 开始学破解先不涉及壳的问题,我们主要是熟悉用 OllyDBG 来破解的一般方法。我这里把壳脱掉来分析,附件 是脱壳后的文件,直接就可以拿来用。先说一下一般软件破解的流程:拿到一个软件先别接着马上用 OllyDBG 调试,先运行一下,有帮助文档的最好先看一下帮助,熟悉一下软件的使用方法,再看看注册的方式。如果是 序列号方式可以先输个假的来试一下,看看有什么反应,也给我们破解留下一些有用的线索。如果没有输入注 册码的地方,要考虑一下是不是读取注册表或 Key 文件(一般称 keyfile,就是程序读取一个文件中的内容来判 断是否注册),这些可以用其它工具来辅助分析。如果这些都不是,原程序只是一个功能不全的试用版,那要 注册为正式版本就要自己来写代码完善了。有点跑题了,呵呵。获得程序的一些基本信息后,还要用查壳的工 具来查一下程序是否加了壳,若没壳的话看看程序是什么编译器编的,如 VC、Delphi、VB 等。这样的查壳工 具有 PEiD 和 FI。有壳的话我们要尽量脱了壳后再来用 OllyDBG 调试,特殊情况下也可带壳调试。下面进入正 题:

我们先来运行一下这个 crackme (用 PEiD 检测显示是 Delphi 编的),界面如图:

Crackers For Freedom CrackMe v3.0					
<u>Official CFF CrackMe v3.0</u>					
Unregistered	Coder Acid Bytes [CFF]				
754-GFX-IER-954	Rel. Date 05/08/2000				
Register now !	This is the official CFF CrackMe If you can manage to crack it, mail Name/Serial to:				
Quit the CrackMe	acidbytes@gmx.net				

这个 crackme 已经把用户名和注册码都输好了,省得我们动手^_^。我们在那个"Register now !"按钮上点击一下,将会跳出一个对话框:



好了,今天我们就从这个错误对话框中显示的"Wrong Serial, try again!"来入手。启动 OllyDBG,选择菜单 文件->打开 载入 CrackMe3.exe 文件,我们会停在这里:

地址	HEX	数据	反汇编	<u> 洋释</u>
00441270	\$	55	PUSH EBP	载入后停在这里
00441271		8BEC	MOV EBP, ESP	A
00441273		83C4 F4	ADD ESP, -OC	.
00441276		B8 60114400	MOV_EAX, CrackMe3.00441160	左上而伴码后的注释栏由双手带

0044127B	E8 E848FCFF	CALL CrackMe3.00405B68	1부모님에 1 신문이다 비야기보기록기로 가가지 다 많이
00441280	A1 442C4400	MOV EAX, DWORD PTR DS: [442C44]	按分号() 英文输入方式)可以输
00441285	8B00	MOV EAX, DWORD PTR DS: [EAX]	
00441287	E8 ECBBFFFF	CALL CrackMe3.0043CE78	人如图所小的注释
0044128C	A1 442C4400	MOV EAX, DWORD PTR DS: [442C44]	
00441291	8B00	MOV EAX, DWORD PTR DS: [EAX]	
00441293	BA D0124400	MOV_EDX, CrackMe3.004412D0	ASCII "Crackers For Freedom CrackMe v3.0"
00441208	RS 17BSERFF	CALL CRACHMAR DOMRCARA	

我们在反汇编窗口中右击,出来一个菜单,我们在查找->所有参考文本字串上左键点击:

查找 (<u>S</u>) 查找参考 (R)	▶ 当前模块中的名称(标签) Ctrl+ ▶ 所有模块中的名称	N
查看 复制到可执行文件 分析		·F ·S
Asm2Clipboard	▶ 二进制字串 (B) Ctrl+	·B
书签 去除花指令 超级拷贝	 ▶ 所有模块间的调用 所有命令 ● ●	
创建标签 也可以用这个插 运行脚本 件来查字串	 所有常量 所有分支 	
用OllyDump脱壳调试进程	所有参考文本字串	
011yFlow 图表 用FEdumper脱壳调试进程 超级字串参考+ W)	▶ 用户定义的标签 用户定义的标签	()

当然如果用上面那个超级字串参考+插件会更方便。但我们的目标是熟悉 OllyDBG 的一些操作,我就尽量使用 OllyDBG 自带的功能,少用插件。好了,现在出来另一个对话框,我们在这个对话框里右击,选择"查找文本"菜单项,输入"Wrong Serial, try again!"的开头单词"Wrong"(注意这里查找内容要区分大小写)来查找,找到一处:

图 文本	字串参考位于 CrackIe3:CODE		×
地址	反汇编	文本字串	~
00440F0D 00440F2F 00440F4C 00440F5A 00440F5F 00440F74	MOV EDX, CrackMe3.004 右键选择这个菜单 MOV EDX, CrackMe3.004 右键选择这个菜单 MOV E. MOV E. MOV E. MOV EDX, CrackMe3.004.这一条上按回车键 MOV EDX, CrackMe3.00441080	II "Enter a Serial!" II "Registered User" II "GFX-754-IER-954" II "CrackMe cracked successfully" IC "Congrats! You cracked this CrackMe!" ASCII "Beggar off!"	
00440F79 00440F8E 00440F93 00440FC8	MOV EDX.CrackMe3.0044108C MOV ECX.CrackMe3.00441080 MOV EDX.CrackMe3.0044108C ASCII "No Name entered",0	ASCII "Wing Serial try again!" ASCII 反汇编窗口中跟随 (2) Enter ASCII 本形立大	

在我们找到的字串上右击,再在出来的菜单上点击"反汇编窗口中跟随",我们来到这里:

地址	HEX 💈	数据	反汇编			注释
00440F79	. B	3A 8C104400	MOV EDX. CrackMe3. 00441	08C	_	ASCII "Wrong Serial, try again!"
00440F7E	. A	1 4420440	备份			
00440F83	. 8	3BOO	有制			
00440F85	. E	8 DECOFFI	夏利		·	/ 发现了网门
00440F8A	. ~ E	CB 18	二进制		۱	
00440F8C	> 5	A UU	に使い	S		10077
00440F8E	. В р	39 8010440	(二3冊(五)	Space		ASUII Beggar off!
00440193	. <u>B</u>	A 8010440	标签	:		ADULI Wrong Derial, try again!
00440790	l	1 4420440 2800	注释			
00440F5D	. U	8 C4COFFF		•		
00440FA4	3	300	断点化)		·	
00440FA6	l. š	5A	HIT 跟踪		F L	这里两个我们都能直接看
00440FA7	5	59	RIN 8月9章			到. 如果想找找看还有没有
00440FA8	. 5	59 _	ком рахря			别的参考的话,我们可以
00440FA9	. 6	34:8910	HV /N ++++F PTP	CtultCurry *		左来接触左行占去嫌 选择
00440FAC	; 6	58 C10F440	PPXENNA FIL	Ctri+Gray *		
00440FB1	> 8	3D45 FC	转到			来里登找参考=>立即数
00440FB4	. E	18 E727FCE	数据窗口中跟随		ьL	1
00440FB9	•. U	3 0 MOORCE -			-1	
00440F5A		AIZZPUI	杏状 (5)		ьL	
00440FDF	. 5	SB I				
00440FC1	. 5	59	登扱参考(風)			変定地址 CJ Ctrl+R
00440FC3	. 5	50	查看	ht	1	· 府即教史dp. perfewyp0000mm

见上图,为了看看是否还有其他的参考,可以通过选择右键菜单查找参考->立即数,会出来一个对话框:

■ 参考	位于 CrackHe3:CODE 到常量 441080	: 🔲 🗖	\mathbf{X}
地址	反汇编	注释	~
00440F79	MOV EDX, CrackMe3. 0044108C 🛛 🚬 🚚	▶ (初始 CPU 选择)	
00440F93	MOV EDX, CrackMe3.0044108C	ASCII "Wrong Serial, try again!"	
	这里对应反汇编窗口中光 标所在的位置,就是我们 刚才看到的第一个参考	这里对应的是其他的参考,在 这行上救击可以跳到反汇编 窗口中的对应位置	~

分别双击上面标出的两个地址,我们会来到对应的位置:

00440F79 |. BA 8C104400 MOV EDX,CrackMe3.0044108C ; ASCII "Wrong Serial, try again!" 00440F7E |. A1 442C4400 MOV EAX, DWORD PTR DS: [442C44] 00440F83 |. 8B00 MOV EAX, DWORD PTR DS:[EAX] 00440F85 |. E8 DEC0FFFF CALL CrackMe3.0043D068 JMP SHORT CrackMe3.00440FA4 00440F8A |. EB 18 PUSH 0 00440F8C |> 6A 00 00440F8E |. B9 80104400 MOV ECX,CrackMe3.00441080 ; ASCII "Beggar off!" ; ASCII "Wrong Serial, try again!" 00440F93 |. BA 8C104400 MOV EDX, CrackMe3.0044108C 00440F98 |. A1 442C4400 MOV EAX, DWORD PTR DS: [442C44] 00440F9D |. 8B00 MOV EAX, DWORD PTR DS:[EAX] 00440F9F |. E8 C4C0FFFF CALL CrackMe3.0043D068 我们在反汇编窗口中向上滚动一下再看看: 00440F2C |. 8B45 FC MOV EAX, DWORD PTR SS: [EBP-4] 00440F2F |. BA 14104400 MOV EDX,CrackMe3.00441014 ; ASCII "Registered User" 00440F34 |. E8 F32BFCFF CALL CrackMe3.00403B2C ;关键,要用F7跟进去 ;这里跳走就完蛋 00440F39 |. 75 51 JNZ SHORT CrackMe3.00440F8C 00440F3B |. 8D55 FC LEA EDX, DWORD PTR SS: [EBP-4] 00440F3E |. 8B83 C8020000 MOV EAX, DWORD PTR DS: [EBX+2C8] 00440F44 |. E8 D7FEFDFF CALL CrackMe3.00420E20 00440F49 |. 8B45 FC MOV EAX, DWORD PTR SS:[EBP-4] 00440F4C |. BA 2C104400 MOV EDX,CrackMe3.0044102C : ASCII "GFX-754-IER-954" 00440F51 |. E8 D62BFCFF CALL CrackMe3.00403B2C ;关键,要用F7跟进去 JNZ SHORT CrackMe3.00440F72 00440F56 |. 75 1A ;这里跳走就完蛋 00440F58 |. 6A 00 PUSH 0 00440F5A |. B9 3C104400 MOV ECX, CrackMe3.0044103C ; ASCII "CrackMe cracked successfully" 00440F5F |. BA 5C104400 MOV EDX, CrackMe3.0044105C ; ASCII "Congrats! You cracked this CrackMe!" 00440F64 |. A1 442C4400 MOV EAX, DWORD PTR DS: [442C44] 00440F69 |. 8B00 MOV EAX, DWORD PTR DS: [EAX] 00440F6B |. E8 F8C0FFFF CALL CrackMe3.0043D068 JMP SHORT CrackMe3.00440FA4 00440F70 |. EB 32 00440F72 |> 6A 00 PUSH 0 00440F74 |. B9 80104400 MOV ECX,CrackMe3.00441080 ; ASCII "Beggar off!" 00440F79 |. BA 8C104400 MOV EDX, CrackMe3.0044108C ; ASCII "Wrong Serial, try again!" 00440F7E |. A1 442C4400 MOV EAX, DWORD PTR DS:[442C44] 00440F831.8B00 MOV EAX, DWORD PTR DS: [EAX]

00440F85 |. E8 DEC0FFFF CALL CrackMe3.0043D068 00440F8A |. EB 18 JMP SHORT CrackMe3.00440FA4 00440F8C |> 6A 00 PUSH 0 00440F8E |. B9 80104400 MOV ECX,CrackMe3.00441080 00440F93 |. BA 8C104400 MOV EDX,CrackMe3.0044108C 00440F98 |. A1 442C4400 MOV EAX,DWORD PTR DS:[442C44] 00440F9D |. 8B00 MOV EAX,DWORD PTR DS:[EAX] 00440F9F |. E8 C4C0FFFF CALL CrackMe3.0043D068

; ASCII "Beggar off!" ; ASCII "Wrong Serial,try again!"

大家注意看一下上面的注释,我在上面标了两个关键点。有人可能要问,你怎么知道那两个地方是关键点?其 实很简单,我是根据查看是哪条指令跳到"wrong serial,try again"这条字串对应的指令来决定的。如果你在 调试 选项->CPU 标签中把"显示跳转路径"及其下面的两个"如跳转未实现则显示灰色路径"、"显示跳转到选定命令的 路径"都选上的话,就会看到是从什么地方跳到出错字串处的:

地址	HEX 数据	反汇编	注释	~			
00440F2C 00440F2F	8845 10	地27 FAX PWOPP PTR SS:[EBP-4] 按F2设个断点 1e3.00441014	ASCII "Registered User"				
00440F34 00440F39 00440F38	. 18 rozbrurr .v.75 51	JNZ SHORT CrackMe3.00403820 JNZ SHORT CrackMe3.00440F8C LEA EDX DWORD PTR SS [FRP-4]	大陸・安用に限定さ 这里跳走就完蛋				
00440F3E 00440F44	8B83 C802000 E8 D71 + 1 - 5	MOV EAX, DWORD PTR DS: [EBX+2C8]					
00440F49 00440F4C	. 8B45 1 ¹¹¹⁰¹⁴⁴ . BA 2C:个方向	响下的箭头表示这里是 <mark>短</mark>	ASCII "GFX-754-IER-954"				
00440F51 00440F56	· E8 D6:往下副	转。如果是向上的箭 WIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	│ 关键,要用ष्टि跟进去 │ 这里跳走就完蛋				
00440F58 00440F5A 00440F5F	. 6A 00 大 り . B9 3C104400 BA 5C104400	いないれた日本リエムのなれて IMUY LUA, UTACKINGS, UU441U3C MOV KDY CrackWa3 0044105C	ASCII "CrackMe cracked successfully" ASCII "Congrets! You procked this GreatMe!"				
00440F64 00440F69	. A1 442C4400 . 8B00	MOV EAX, DWORD PTR DS: [442C44] MOV EAX, DWORD PTR DS: [EAX]	Abori congrats: fou cracked this crackine:				
00440F6B 00440F70	E8 F8 注意机 EB 32 注意机	器码 6A 00 前面那 0FA4					
00440F72 00440F74	> 6A 00 个~>~? B9 80 标,在	符号,这里表示跳转自 这条指令上单击一下 <mark>80</mark>	ASCII "Beggar off!"				
00440F7E 00440F7E 00440F83	A1 44:就可以 8B00	看到跳转路径 42C44]	ASCII mrong Serial, try again!				
00440F85 00440F8A	E8 DECOFFFF	CALL Crackie3.0043D068 JMP SHORT Crackie3.00440FA4					
00440F8C	> 96A 00	PUSH 0					
00440F8E	. B9 80104400	MOV ECX, CrackMe3.00441080	ASCII "Beggar off!"				
00440193	. BA 80104400	MUY EDA, CrackMed. UU441060	ADULL Wrong Derial, try again!				
00440190	. AI 44204400 8800	MOV RAY DWORD PTR DS. [442044]					
00440F9F	E8 C4COFFFF	CALL CrackMe3 00430068					
00440FA4	> 3300	XOR EAX, EAX		~			
跳转来自	新教来自 00440F39 信自窓口中地約金印目ティールの						
		处跳转到当时尤标所在位星的	https://blog.csdn.net/wxh0000	mm			

我们在上图中地址 00440F2C 处按 F2 键设个断点,现在我们按 F9 键,程序已运行起来了。我在上面那个编辑 框中随便输入一下,如 CCDebuger,下面那个编辑框我还保留为原来的"754-GFX-IER-954",我们点一下那 个"Register now !"按钮,呵,OllyDBG 跳了出来,暂停在我们下的断点处。我们看一下信息窗口,你应该发现 了你刚才输入的内容了吧?我这里显示是这样:

堆栈 SS:[0012F9AC]=00D44DB4, (ASCII "CCDebuger") EAX=00000009

上面的内存地址 00D44DB4 中就是我们刚才输入的内容,我这里是 CCDebuger。你可以在 堆栈 SS: [0012F9AC]=00D44DB4, (ASCII "CCDebuger") 这条内容上左击选择一下,再点右键,在弹出菜单中选择"数据 窗口中跟随数值",你就会在下面的数据窗口中看到你刚才输入的内容。而 EAX=00000009 指的是你输入内容的 长度。如我输入的 CCDebuger 是9个字符。如下图所示:

🔆 – [сри	- 主线程	,模块 - Crac	kle3]			
🖸 文件 🕑	查看(Y) 调	間试@) 插件@)	选项(11) 窗口() 帮助(H)		_ 8 ×
		i = 1 = 1 =		MTWHC7K	B R S 🗮 📰	?
地址 HEX	数据	反汇编		注释	▲ 寄存器	(FPU)

00440F2C	. 8B45 FC	MOV EAX, DWORD F	TR SS:[EBP-4]				EA	X 00000009	
00440F2F	. BA 14104400	MOV EDX, CrackMe	3.00441014		ASCII	"Registered Use	EC	X 77D1882A	USER32.77D1882A
00440F34	. E8 F32BFCFF	CALL CrackMe3.0	0403B2C		关键,	要用F7跟进去	ED	X 00140608	
00440F39	. 🗸 75 51	JNZ SHORT Crack	Me3.00440F8C		这里跳	走就完蛋	EB	X 00D44830	
00440F3B	. 8D55 FC	LEA EDX, <mark>DWORD F</mark>	TR SS:[EBP-4]				ES	P 0012F99C	
00440F3E	. 8B83 C802000	MOV EAX, DWORD F	TR DS:[EBX+2C8]				EB	P 0012F9B0	
00440F44	. E8 D7FEFDFF	CALL CrackMe3.0	0420E20				ES	I 00D461B0	
00440F49	. 8B45 FC	MOV EAX, <mark>dword f</mark>	TR SS:[EBP-4]				ED	I 00000013	
00440F4C	. BA 2C104400	MOV_EDX, CrackMe	3.0044102C		ASCII	"GFX-754-IER-95	RT	P 0044082C	Creatille3_00440820
00440F51	. E8 D62BFCFF	CALL CrackMe3.0	10403B2C		关键,	要用F7跟进去	1.27	1 00440120	CI ACAMED, 00440F20
00440F56	75 1A	JNZ SHORT Crack	Me3.00440F72		这里跳	走就完蛋	C	0 ES 0023	32(<u>文</u> 0 (FFFFFFFF)
00440F58	. 6A 00	PUSH 0					P	0 CS 001B	32(文 O (FFFFFFFF))
00440F5A	. B9 3C104400	MOV_ECX, <mark>CrackM</mark> e	3. 0044103C		ASCII	"CrackMe cracke	A	0 SS 0023	32(文 0 (FFFFFFFF))
00440F5F	. BA 5C104400	MOV_EDX, <mark>CrackMe</mark>	3.0044105C		ASCII	"Congrats! You	Z	0 DS 0023	32位 0(FFFFFFF)
00440F64	. A1 442C4400	MOV EAX, DWORD F	TR DS:[442C44]				S	0 FS 003B	32位 7FFDF000 (FFF
00440F69	. 8B00	MOV EAX, DWORD F	TR DS:[EAX]				T	0 GS 0000	NULL
00440F6B	. E8 F8C0FFFF	CALL CrackMe3.0	043D068				D	0	
00440F70	. ∨ EB 32	JMP SHORT Crack	Me3.00440FA4				0	0 LastErr	ERROR_SUCCESS (00
00440F72	> 6A 00	PUSH 0					FF	1 00000202	OND ME ME & MS PO
00440F74	. B9 80104400	MOV ECX, <mark>CrackM</mark> e	3.00441080		ASCII	"Beggar off!"	1.11	1 00000202	(10, 10, 11, 1, 10, 10
00440F79	. BA 8C104400	MOV_EDX, <mark>CrackMe</mark>	3.0044108C		ASCII	"Wrong Serial, t	ST	0 empty -1	53020323596546355
00440F7E	. A1 442C4400	MOV EAX, DWORD F	TR DS:[442C44]				= ST	1 empty – <mark>U</mark>	NORM A654 0000000
00440F83	. 8B00	MOV EAX, DWORD F	TR DS:[EAX]				ST ST	2 empty - 5	89484696512789504
00440F85	. ES DECOFFFF	CALL CrackMe3.0	0430068				ST ST	3 empty 3.	370944181138696192
· 堆板 SS: L	UU12F9ACJ=UUD44D	B4, ÇASCII "CCDe	ebuger")				ST	4 empty U.I	120993585905696696
EAX=00000	1009 拷贝窗桐	各到剪贴板	1	広ば	三山的洲	堀	ST	5 empty -1.	.12920271423374976
	A发,只有米行力	£		P 01	1.I.H1∞		SI	b empty 1.	000000000000000000000000000000000000000
10.11		16 ·					21	í empty 1.1	
地址	<u>HEX 数</u> : 数据窗口	口中跟随地址		ASCI			J12FB	이 말 이 말 이 이	T SER 记录的指 🔨
OOD44DB4	43 43 (数据密)	コロ盟随数店	00 74 34 44 00	CCDebuger	t4D.	UU12F9AU U	J44UF.	BA HE HE POR	「一山的粉捉
OOD44DC4	30D44DC4 74 34 2004 20 + PRAVENTE 2D 47 46 58 2D + 4D. xt. 754-GFX- 001279A4 001279A4 001279A4								
00D44DD4 49 45 9 BACK TO VO 14 34 44 00 IER-954. t4D. t4D. 1 00121970.0 00046DD									
OOD44DE4	00D44DE4 5C 10 (2FUBLE-20 00 00 00 00 00 \+.??								
(00D44DF4 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00									
命令:	命令: 🗾								
断点位于	CrackMe3.0044	OF2C				https:/	//hlo	a csán n	et/暫停0000mm
						111 P.11	*******	CONTRACTOR OF THE	

现在我们来按 F8 键一步步分析一下:

00440F2C |. 8B45 FC MOV EAX,DWORD PTR SS:[EBP-4] ;把我们输入的内容送到EAX,我这里 是"CCDebuger" 00440F2F |. BA 14104400 MOV EDX,CrackMe3.00441014 ;ASCII "Registered User" 00440F34 |. E8 F32BFCFF CALL CrackMe3.00403B2C ;关键,要用F7跟进去 00440F39 |. 75 51 JNZ SHORT CrackMe3.00440F8C ;这里跳走就完蛋

当我们按 F8 键走到 00440F34 |. E8 F32BFCFF CALL CrackMe3.00403B2C 这一句时,我们按一下 F7 键, 进入这个 CALL,进去后光标停在这一句:

地址	HEX数据	反汇编	注释
00403B2C 00403B2D 00403B2E 00403B2F	 53 58 光标停在这里 57 89Ct 色时表示我(PICH TRY 里。地址底色显示为黑 门现在执行到这条指令	

我们所看到的那些 PUSH EBX、 PUSH ESI 等都是调用子程序保存堆栈时用的指令,不用管它,按 F8 键一步步过来,我们只关心关键部分:

00403B2C /\$ 53	PUSH EBX	
00403B2D . 56	PUSH ESI	
00403B2E . 57	PUSH EDI	
00403B2F . 89C6	MOV ESI,EAX	;把EAX内我们输入的用户名送到 ESI
00403B31 . 89D7	MOV EDI,EDX	;把EDX内的数据"Registered User"送到EDI
00403B33 . 39D0	CMP EAX,EDX	;用"Registered User"和我们输入的用户名作比较
00403B35 . 0F84 8F00	0000 JE CrackMe3.00403BCA	A ;相同则跳
00403B3B . 85F6 户名	TEST ESI,ESI	;看看ESI中是否有数据,主要是看看我们有没有输入用
00403B3D . 74 68	JE SHORT CrackMe3.00403	BA7 ;用户名为空则跳
00403B3F . 85FF	TEST EDI,EDI	
00403B41 . 74 6B	JE SHORT CrackMe3.00403	BAE

00403B43 |. 8B46 FC MOV EAX, DWORD PTR DS:[ESI-4] ;用户名长度送EAX 00403B46 |. 8B57 FC MOV EDX, DWORD PTR DS: [EDI-4] ; "Registered User"字串的长度送EDX ;把用户名长度和"Registered User"字串长度相减 00403B49 |. 29D0 SUB EAX, EDX 00403B4B | 77 02 JA SHORT CrackMe3.00403B4F ;用户名长度大于"Registered User"长度则跳 00403B4D |. 01C2 ;把减后值与"Registered User"长度相加,即用户名长 ADD EDX.EAX 度 00403B4F |> 52 PUSH EDX 00403B50 |. C1EA 02 SHR EDX,2 ;用户名长度值右移2位,这里相当于长度除以4 00403B53 |. 74 26 JE SHORT CrackMe3.00403B7B :上面的指令及这条指令就是判断用户名长度 最少不能低于4 00403B55 |> 8B0E MOV ECX, DWORD PTR DS: [ESI] :把我们输入的用户名送到ECX 00403B57 |. 8B1F MOV EBX, DWORD PTR DS: [EDI] ;把"Registered User"送到EBX 00403B59 |. 39D9 CMP ECX, EBX :比较 00403B5B |. 75 58 JNZ SHORT CrackMe3.00403BB5 ;不等则完蛋

根据上面的分析,我们知道用户名必须是"Registered User"。我们按 F9 键让程序运行,出现错误对话框,点确定,重新在第一个编辑框中输入"Registered User",再次点击那个"Register now !"按钮,被 OllyDBG 拦下。因为地址 00440F34 处的那个 CALL 我们已经分析清楚了,这次就不用再按 F7 键跟进去了,直接按 F8 键通过。我们一路按 F8 键,来到第二个关键代码处:

00440F49 . 8B45 FC	MOV EAX, DWORD PTR SS:[EBP-4]	;取输入的注册码
00440F4C . BA 2C10440	MOV EDX, CrackMe3.0044102C	; ASCII "GFX-754-IER-954"
00440F51 . E8 D62BFCF	F CALL CrackMe3.00403B2C	;关键,要用F7跟进去
00440F56 . 75 1A	INZ SHORT CrackMe3.00440F72	;这里跳走就完蛋

大家注意看一下,地址 00440F51 处的 CALL CrackMe3.00403B2C 和上面我们分析的地址 00440F34 处的 CALL CrackMe3.00403B2C 是不是汇编指令都一样啊? 这说明检测用户名和注册码是用的同一个子程序。而这 个子程序 CALL 我们在上面已经分析过了。我们执行到现在可以很容易得出结论,这个 CALL 也就是把我们输入的注册码与 00440F4C 地址处指令后的"GFX-754-IER-954"作比较,相等则 OK。好了,我们已经得到足够的 信息了。现在我们在菜单 查看->断点 上点击一下,打开断点窗口(也可以通过组合键 ALT+B 或点击工具栏上 那个"B"图标打开断点窗口):



为什么要做这一步,而不是把这个断点删除呢?这里主要是为了保险一点,万一分析错误,我们还要接着分析,要是把断点删除了就要做一些重复工作了。还是先禁用一下,如果经过实际验证证明我们的分析是正确的,再删不迟。现在我们把断点禁用,在 OllyDBG 中按 F9 键让程序运行。输入我们经分析得出的内容: 用户名: Registered User 注册码: GFX-754-IER-954 点击"Register now !"按钮,呵呵,终于成功了:

