

软件逆向笔记

1、软件汇编指令比较

```
1  cmp a,b  //比较a与b
2  mov a,b  //把b值送给a, 使a=b
3  ret      //返回主程序
4  nop      //无作用
5  call     //调用子程序, 子程序以ret结尾
6  je或jz   //相等则跳 (机器码是74或84)
7  jne或jnz //不相等则跳 (机器码是75或85)
8  jmp      //无条件跳
9  jb       //若小于则跳
10 ja      //若大于则跳
11 jg       //若大于则跳
12 jge     //若大于等于则跳
13 jl      //若小于则跳
14 pop xxx  //xxx出栈
15 push xxx //xxx压栈
```

2、字符串

```
1  Unregister  ---  未注册
2  Trial  -----  试用
3  success  -----  成功
4  unreg  -----  未校准
```

3、按钮事件查找

```
1  C++的按钮事件采用查找SUB EAX,0A
2  DELPHI的按钮事件查找二进制字符串740E8BD38B83????????FF93????????,每一个都需下断点
3  易语言按钮事件特征码: FF 55 FC 5F 5E
4  VC8的按钮事件采用查找SUB EAX,0A
5  VB的按钮事件查找二进制字符串816C2404 ?? 000000
6  (注: 识别VB和P-code编译时, 只要找不到按钮事件就是P-code编译)
```

4、网站文章

- * [x64dbg使用技巧与实用插件合集](#)
- * [x64官网](#)
- * [Baymax Patch Tools v2.9.5.4/x64 v2.5.4 \(06月08号更新\)](#)
 - [【五一礼物】StudyPE+ v1.11 出炉](#)

5、x64dbg快捷键

快捷键	功能
F2	在光标处设置或取消断点
Ctrl + F2	重新启动 被调试程序
F4	运行到所选指令处 ,如果选择的指令不被执行, 则被调试程序一直运行下去
F7	单步 步进 , 遇到 call 等子程序时进入其中
Shift + F7	单步步进 , 但当被调试程序发生异常时, 调试器尝试步入被调试程序指定的异常处理
Ctrl + F7	自动步进 , 在所有的函数调用中一条一条地执行命令
F8	单步 步过 , 遇到 call 等子程序时不进入其中
Shift + F8	单步步过 , 但当被调试程序发生异常时, 调试器尝试步入被调试程序指定的异常处理
Ctrl + F8	自动步过 , 在所有的函数调用中一条一条地执行命令
F9	运行, 直到遇到下一个断点
Shift + F9	运行到断点, 但当被调试程序发生异常时, 调试器尝试步入被调试程序指定的异常处理
Ctrl + F9	执行到返回 , 在执行到一个 ret 指令时暂停, 常用于从当前函数快速返回到上一个函数
Alt + F9	执行到用户代码 , 可用于从系统部分快速返回到被调试程序部分
F12	停止执行 , 暂停所有线程, 可用于暂停自动执行
;	添加注释
:	添加标签
-	回退上一条指令的位置
+	执行下一条指令

6、寄存器

通用寄存器

AX(AH、AL)：累加器。有些指令约定以AX(或AL)为源或目的寄存器。输入/输出指令必须通过AX或AL实现，例如：端口地址为43H的内容读入CPU的指令为INAL，43H或INAX，43H。目的操作数只能是AL/AX，而不能是其他的寄存器。

BX(BH、BL)：基址寄存器。BX可用作间接寻址的**地址寄存器**和基地址寄存器，BH、BL可用作8位通用**数据寄存器**。

CX(CH、CL)：计数寄存器。CX在循环和串操作中充当计数器，指令执行后CX内容自动修改，因此称为计数寄存器。

DX(DH、DL)：数据寄存器。除用作通用寄存器外，在I/O指令中可用作端口地址寄存器，乘除指令中用作辅助累加器。

指针和变址寄存器

BP(Base Pointer regilter)：基址指针寄存器。

SP(Stack Pointer Register)：**堆栈指针**寄存器。

SI(Source Index register)：源变址寄存器。

DI(Destination Index Register)：目的变址寄存器。

这组寄存器存放的内容是某一段内地址偏移量，用来形成操作数地址，主要在堆栈操作和变址运算中使用。BP和SP寄存器称为指针寄存器，与SS联用，为访问现行堆栈段提供方便。通常BP寄存器在间接寻址中使用，操作数在堆栈段中，由SS段寄存器与BP组合形成操作数地址即BP中存放现行堆栈段中一个数据区的“基址”的偏移量，所以称BP寄存器为基址指针。

SP寄存器在堆栈操作中使用，PUSH和POP指令是从SP寄存器得到现行堆栈段的段内地址偏移量，所以称SP寄存器为堆栈指针，SP始终指向栈顶。

寄存器SI和DI称为变址寄存器，通常与DS一起使用，为访问现行数据段提供段内地址偏移量。在串指令中，其中源操作数的偏移量存放在SI中，目的操作数的偏移量存放在DI中，SI和DI的作用不能互换，否则传送地址相反。在串指令中，SI、DI均为隐含寻址，此时，SI和DS联用，DI和ES联用。

段寄存器

8086/8088CPU可直接寻址1MB的存储器空间，直接寻址需要20位地址码，而所有内部寄存器都是16位的，只能直接寻址64KB，因此采用分段技术来解决。将1MB的存储空间分成若干逻辑段，每段最长64KB，这些逻辑段在整个存储空间中可浮动。

8086/8088CPU内部设置了4个16位段寄存器，它们分别是代码段寄存器CS、数据段寄存器DS、堆栈段寄存器SS、附加段寄存器ES、由它们给出相应逻辑段的首地址，称为“段基址”。段基址与段内偏移地址组合形成20位物理地址，段内偏移地址可以存放在寄存器中，也可以存放在存储器中。

例如：代码段寄存器CS存放当前代码段基地址，IP指令指针寄存器存放了下一条要执行指令的段内偏移地址，其中CS=2000H，IP=001AH。通过组合，形成20位存储单元的寻址地址为2001AH。

代码段内存放可执行的指令代码，数据段和附加段内存放操作的数据，通常操作数在现行数据段中，而在串指令中，目的操作数指明必须在现行附加段中。堆栈段开辟为程序执行中所要用的堆栈区，采用先进后出的方式访问它。各个段寄存器指明了一个规定的现行段，各段寄存器不可互换使用。程序较小时，代码段、数据段、堆栈段可放在一个段内，即包含在64KB之内，而当程序或数据量较大时，超过了64KB，那么可以定义多个代码段或数据段、堆栈段、附加段。现行段由段寄存器指明段地址，使用中可以修改段寄存器内容，指向其他段。有时为了明确起见，可在指令前加上段超越的前缀，以指定操作数所在段。

指令指针寄存器IP

8086/8088CPU中设置了一个16位指令指针寄存器IP，用来存放将要执行的下一条指令在现行代码段中的偏移地址。程序运行中，它由BIU自动修改，使IP始终指向下一条将要执行的指令的地址，因此它是用来控制指令序列的执行流程的，是一个重要的寄存器。8086程序不能直接访问IP，但可以通过某些指令修改IP的内容。例如，当遇到中断指令或调用 **子程序** 指令时，8086自动调整IP的内容，将IP中下一条将要执行的指令地址偏移量入栈保护，待中断程序执行完毕或子程序返回时，可将保护的内容从堆栈中弹出到IP，使 **主程序** 继续运行。在跳转指令时，则将新的跳转目标地址送入IP，改变它的内容，实现了程序的转移。

标志寄存器FR

标志寄存器FR也称程序状态字寄存器。

FR是16位寄存器，其中有9位有效位用来存放状态标志和控制标志。状态标志共6位，CF、PF、AF、ZF、SF和OF，用于寄存程序运行的状态信息，这些标志往往用作后续指令判断的依据。控制标志有3位，IF、DF和TF，用于控制CPU的操作，是人为设置的。

寄存器的工作原理

在计算机及其他计算系统中，寄存器是一种非常重要的、必不可少的数字电路器件，它通常由 **触发器**（**D触发器**）组成，主要作用是用来暂时存放数码或指令。一个触发器可以存放一位二进制代码，若要存放N位二进制数码，则需用N个触发器。

寄存器应具有接收数据、存放数据和输出数据的功能，它由触发器和门电路组成。只有得到“存入脉冲”（又称“存入指令”、“写入指令”）时，寄存器才能接收数据；在得到“读出”指令时，寄存器才将数据输出。

寄存器存放数码的方式有并行和串行两种。并行方式是数码从各对应位输入端同时输入到寄存器中；串行方式是数码从一个输入端逐位输入到寄存器中。

寄存器读出数码的方式也有并行和串行两种。在并行方式中，被读出的数码同时出现在各位的输出端上；在串行方式中，被读出的数码在一个输出端逐位出现。

7、常用的5种编程语言入口（OEP）特征

1、Borland Delphi

地址	HEX 数据	反汇编	注释
0044EDF4	\$ 55	push ebp	
0044EDF5	. 8BEC	mov ebp,esp	
0044EDF7	. 83C4 F0	add esp,-0x10	
0044EDFA	. B8 14EC4400	mov eax, 吾爱破解.0044EC14	UNICODE ";
0044EDFF	. E8 C46DFBFF	call 吾爱破解.00405BC8	
0044EE04	. A1 D0FF4400	mov eax,dword ptr ds:[0x44FFD0]	
0044EE09	. 8B00	mov eax,dword ptr ds:[eax]	
0044EE0B	. E8 04E6FFFF	call 吾爱破解.0044D414	
0044EE10	. A1 D0FF4400	mov eax,dword ptr ds:[0x44FFD0]	
0044EE15	. 8B00	mov eax,dword ptr ds:[eax]	
0044EE17	. 83C0 50	add eax,0x50	
0044EE1A	. BA 68EE4400	mov edx, 吾爱破解.0044EE68	ASCII "吾爱破解论坛学习脱壳实例"
0044EE1F	. E8 4050FBFF	call 吾爱破解.00403E64	
0044EE24	. A1 D0FF4400	mov eax,dword ptr ds:[0x44FFD0]	
0044EE29	. 8B00	mov eax,dword ptr ds:[eax]	
0044EE2B	. BA 68EE4400	mov edx, 吾爱破解.0044EE68	ASCII "吾爱破解论坛学习脱壳实例"
0044EE30	. E8 EFE1FFFF	call 吾爱破解.0044D024	
0044EE35	. 8B00 AC00450	mov ecx,dword ptr ds:[0x4500AC]	吾爱破解.00451BD0
0044EE3B	. A1 D0FF4400	mov eax,dword ptr ds:[0x44FFD0]	
0044EE40	. 8B00	mov eax,dword ptr ds:[eax]	
0044EE42	. 8B15 84E9440	mov edx,dword ptr ds:[0x44E9B4]	吾爱破解.0044EA00
0044EE48	. E8 DFE5FFFF	call 吾爱破解.0044D42C	
0044EE4D	. A1 D0FF4400	mov eax,dword ptr ds:[0x44FFD0]	
0044EE52	. 8B00	mov eax,dword ptr ds:[eax]	
0044EE54	. E8 53E6FFFF	call 吾爱破解.0044D4AC	
0044EE59	. E8 C24EFBFF	call 吾爱破解.00403D20	

2、Microsoft Visual C++ 6.0

地址	HEX 数据	反汇编	注释
00401700	\$ 55	push ebp	
00401701	. 8BEC	mov ebp,esp	
00401703	. 6A FF	push -0x1	
00401705	. 68 00254000	push 吾爱破解.00402500	
0040170A	. 68 86184000	push <jmp.&MSUCRT._except_handler3>	SE 处理程序安装
0040170F	. 64:A1 000000	mov eax,dword ptr fs:[0]	
00401715	. 50	push eax	
00401716	. 64:8925 0000	mov dword ptr fs:[0],esp	
0040171D	. 83EC 68	sub esp,0x68	
00401720	. 53	push ebx	
00401721	. 56	push esi	
00401722	. 57	push edi	
00401723	. 8965 E8	mov [local.6],esp	
00401726	. 33DB	xor ebx,ebx	
00401728	. 895D FC	mov [local.1],ebx	
0040172B	. 6A 02	push 0x2	
0040172D	. FF15 9021400	call dword ptr ds:[<&MSUCRT.__set_app_type>	msvcrt.__set_app_type
00401733	. 59	pop ecx	kernel32.7C817077
00401734	. 830D 4C31400	or dword ptr ds:[0x40314C],-0x1	
0040173B	. 830D 5031400	or dword ptr ds:[0x403150],-0x1	
00401742	. FF15 8C21400	call dword ptr ds:[<&MSUCRT.__p__fmode>	msvcrt.__p__fmode
00401748	. 8B0D 4031400	mov ecx,dword ptr ds:[0x403140]	
0040174E	. 8908	mov dword ptr ds:[eax],ecx	
00401750	. FF15 8821400	call dword ptr ds:[<&MSUCRT.__p__commode>	msvcrt.__p__commode
00401756	. 8B0D 3C31400	mov ecx,dword ptr ds:[0x40313C]	
0040175C	. 8908	mov dword ptr ds:[eax],ecx	

3、Microsoft Visual Basic 5.0 / 6.0

地址	HEX 数据	反汇编	注释
0040104C	68 3C1F4000	push 吾爱破解.00401F3C	ASCII "U85?##ub6chs.dll"
00401051	E8 EFFFFFFF	call <jmp.&MSVBU60.#ThunRTMain_100>	
00401056	0000	add byte ptr ds:[eax],al	
00401058	0000	add byte ptr ds:[eax],al	
0040105A	0000	add byte ptr ds:[eax],al	
0040105C	3000	xor byte ptr ds:[eax],al	
0040105E	0000	add byte ptr ds:[eax],al	
00401060	3800	cmp byte ptr ds:[eax],al	
00401062	0000	add byte ptr ds:[eax],al	
00401064	0000	add byte ptr ds:[eax],al	
00401066	0000	add byte ptr ds:[eax],al	
00401068	96	xchg eax,esi	
00401069	47	inc edi	
0040106A	0E	push cs	
0040106B	9F	lahf	
0040106C	321E	xor bl,byte ptr ds:[esi]	
0040106E	9E	sahf	
0040106F	47	inc edi	
00401070	84 7A	mov ah,0x7A	
00401072	73 D4	jnb short 吾爱破解.00401048	
00401074	F7D1	not ecx	
00401076	54	push esp	
00401077	F8	clic	
00401078	0000	add byte ptr ds:[eax],al	
0040107A	0000	add byte ptr ds:[eax],al	
0040107C	0000	add byte ptr ds:[eax],al	

4、MASM32 / TASM32

地址	HEX 数据	反汇编	注释
00401088	6A 00	push 0x0	
0040108D	E8 4A000000	call <jmp.&kernel32.GetModuleHandleA>	
00401092	A3 04304000	mov dword ptr ds:[0x403004],eax	
00401097	6A 00	push 0x0	
00401099	68 00104000	push 吾爱破解.00401000	
0040109E	6A 00	push 0x0	
004010A0	68 E8030000	push 0x3E8	
004010A5	FF35 04304000	push dword ptr ds:[0x403004]	
004010AB	E8 0E000000	call <jmp.&user32.DialogBoxParamA>	
004010B0	6A 00	push 0x0	
004010B2	E8 1F000000	call <jmp.&kernel32.ExitProcess>	
004010B7	CC	int3	
004010B8	- FF25 0C204000	jmp dword ptr ds:[<&shell32.ShellExecuteA>	shell32.ShellExecuteA
004010BE	- FF25 18204000	jmp dword ptr ds:[<&user32.DialogBoxParamA>	user32.DialogBoxParamA
004010C4	- FF25 14204000	jmp dword ptr ds:[<&user32.EndDialog>	user32.EndDialog
004010CA	- FF25 20204000	jmp dword ptr ds:[<&user32.LoadIconA>	user32.LoadIconA
004010D0	- FF25 1C204000	jmp dword ptr ds:[<&user32.SendMessageA>	user32.SendMessageA
004010D6	- FF25 00204000	jmp dword ptr ds:[<&kernel32.ExitProcess>	kernel32.ExitProcess
004010DC	- FF25 04204000	jmp dword ptr ds:[<&kernel32.GetModuleHandleA>	kernel32.GetModuleHandleA
004010E2	0000	add byte ptr ds:[eax],al	
004010E4	0000	add byte ptr ds:[eax],al	
004010E6	0000	add byte ptr ds:[eax],al	
004010E8	0000	add byte ptr ds:[eax],al	
004010EA	0000	add byte ptr ds:[eax],al	
004010EC	0000	add byte ptr ds:[eax],al	
004010EE	0000	add byte ptr ds:[eax],al	

5、易语言独立编译

地址	HEX 数据	反汇编	注释
0046C607	55	push ebp	
0046C608	8BEC	mov ebp,esp	
0046C60A	6A FF	push -0x1	
0046C60C	68 006A4900	push 吾爱破解.00496A00	
0046C611	68 BC124700	push 吾爱破解.004712BC	
0046C616	64:A1 00000000	mov eax,dword ptr fs:[0]	
0046C61C	50	push eax	
0046C61D	64:8925 000000	mov dword ptr fs:[0],esp	
0046C624	83EC 58	sub esp,0x58	
0046C627	53	push ebx	
0046C628	56	push esi	
0046C629	57	push edi	
0046C62A	8965 E8	mov dword ptr ss:[ebp-0x18],esp	
0046C62D	FF15 38C34800	call dword ptr ds:[<&KERNEL32.GetVersion	kernel32.GetVersion
0046C633	33D2	xor edx,edx	ntdll.KiFastSystemCallRet
0046C635	8AD4	mov dl,ah	
0046C637	8915 48134C00	mov dword ptr ds:[0x4C1348],edx	ntdll.KiFastSystemCallRet
0046C63D	8BC8	mov ecx,eax	
0046C63F	81E1 FF000000	and ecx,0xFF	
0046C645	890D 44134C00	mov dword ptr ds:[0x4C1344],ecx	
0046C648	C1E1 08	shl ecx,0x8	
0046C64E	03CA	add ecx,edx	ntdll.KiFastSystemCallRet
0046C650	890D 40134C00	mov dword ptr ds:[0x4C1340],ecx	
0046C656	C1E8 10	shr eax,0x10	
0046C659	A3 3C134C00	mov dword ptr ds:[0x4C133C],eax	
0046C65E	6A 01	push 0x1	

6、Borland C++ 1999 (BC++6)

地址	HEX 数据	反汇编	注释
004012FC	EB 10	jmp short 吾爱破解.0040130E	
004012FE	66:623A	bound di,dword ptr ds:[edx]	
00401301	43	inc ebx	
00401302	2B2B	sub ebp,dword ptr ds:[ebx]	
00401304	48	dec eax	
00401305	4F	dec edi	
00401306	4F	dec edi	
00401307	4B	dec ebx	
00401308	90	nop	
00401309	E9 98304600	jmp 008643A6	
0040130E	A1 8B304600	mov eax,dword ptr ds:[0x46308B]	
00401313	C1E0 02	shl eax,0x2	
00401316	A3 8F304600	mov dword ptr ds:[0x46308F],eax	
0040131B	52	push edx	ntdll.KiFastSystemCallRet
0040131C	6A 00	push 0x0	
0040131E	E8 670A0600	call <jmp.&KERNEL32.GetModuleHandleA>	
00401323	8BD0	mov edx,eax	
00401325	E8 62730500	call 吾爱破解.0045868C	kernel32.7C817077
0040132A	5A	pop edx	
0040132B	E8 C0720500	call 吾爱破解.004585F0	
00401330	E8 97730500	call 吾爱破解.004586CC	
00401335	6A 00	push 0x0	
00401337	E8 9C870500	call 吾爱破解.00459AD8	kernel32.7C817077
0040133C	59	pop ecx	
0040133D	68 34304600	push 吾爱破解.00463034	
00401342	6A 00	push 0x0	

7、Borland C++ 1999 (BC++2010) 感觉跟BC++6一样

地址	HEX 数据	反汇编	注释
004014EC	EB 10	jmp short 吾爱破解.004014FE	
004014EE	66:623A	bound di,dword ptr ds:[edx]	
004014F1	43	inc ebx	
004014F2	2B2B	sub ebp,dword ptr ds:[ebx]	
004014F4	48	dec eax	
004014F5	4F	dec edi	
004014F6	4F	dec edi	
004014F7	4B	dec ebx	
004014F8	90	nop	
004014F9	E9 ACB04C00	jmp 008CC5AA	
004014FE	A1 9FB04C00	mov eax,dword ptr ds:[0x4CB09F]	
00401503	C1E0 02	shl eax,0x2	
00401506	A3 A3B04C00	mov dword ptr ds:[0x4CB0A3],eax	
00401508	52	push edx	ntdll.KiFastSystemCallRet
0040150C	6A 00	push 0x0	
0040150E	E8 578F0C00	call <jmp.&KERNEL32.GetModuleHandleA>	
00401513	8BD0	mov edx,eax	
00401515	E8 1ECC0B00	call 吾爱破解.004BE138	
0040151A	5A	pop edx	kerne132.7C817077
0040151B	E8 18C80B00	call 吾爱破解.004BD038	
00401520	E8 2FD10B00	call 吾爱破解.004BE654	
00401525	6A 00	push 0x0	
00401527	E8 20E40B00	call 吾爱破解.004BF94C	
0040152C	59	pop ecx	kerne132.7C817077
0040152D	68 48B04C00	push 吾爱破解.004CB048	
00401532	6A 00	push 0x0	

8、Microsoft Visual C++ 8 * (VS2008)

地址	HEX 数据	反汇编	注释
0041DDAC	E8 EF4E0000	call 吾爱破解.00422CA0	
0041DDB1	E9 79FEFFFF	jmp 吾爱破解.0041DC2F	
0041DDB6	3B0D B0074400	cmp ecx,dword ptr ds:[0x4407B0]	
0041DDBC	75 02	jnz short 吾爱破解.0041DDC0	
0041DDBE	F3:c3	rep ret	
0041DDC0	E9 714F0000	jmp 吾爱破解.00422D36	
0041DDC5	8BF8	mov edi,edi	
0041DDC7	55	push ebp	
0041DDC8	8BEC	mov ebp,esp	
0041DDCA	51	push ecx	
0041DDCB	53	push ebx	
0041DDCC	56	push esi	
0041DDCD	57	push edi	
0041DDCE	FF35 A8574400	push dword ptr ds:[0x4457A8]	
0041DDD4	E8 75490000	call 吾爱破解.0042274E	
0041DDD9	FF35 A4574400	push dword ptr ds:[0x4457A4]	
0041DDDF	8BF8	mov edi,eax	
0041DDE1	897D FC	mov dword ptr ss:[ebp-0x4],edi	
0041DDE4	E8 65490000	call 吾爱破解.0042274E	
0041DDE9	8BF8	mov esi,eax	
0041DDEB	59	pop ecx	kerne132.7C817077
0041DDEC	59	pop ecx	kerne132.7C817077
0041DDED	3BF7	cmp esi,edi	
0041DDEF	0F82 83000000	jb 吾爱破解.0041DE78	
0041DDF5	8BDE	mov ebx,esi	
0041DDF7	2BDF	sub ebx,edi	

9、Microsoft Visual C++ 8 * (VS2013)

地址	HEX 数据	反汇编	注释
00423359	E8 A9520000	call 吾爱破解.00428607	
0042335E	E9 7FFFFFFFFF	jmp 吾爱破解.004231E2	
00423363	3B0D F0A64400	cmp ecx,dword ptr ds:[0x44A6F0]	
00423369	75 02	jnz short 吾爱破解.0042336D	
0042336B	F3:C3	rep ret	
0042336D	E9 F8340000	jmp 吾爱破解.0042686A	
00423372	55	push ebp	
00423373	8BEC	mov ebp,esp	
00423375	837D 08 00	cmp dword ptr ss:[ebp+0x8],0x0	
00423379	74 2D	je short 吾爱破解.004233A8	
0042337B	FF75 08	push dword ptr ss:[ebp+0x8]	吾爱破解.<ModuleEntryPoint>
0042337E	6A 00	push 0x0	
00423380	FF35 70F04400	push dword ptr ds:[0x44F070]	
00423386	FF15 A4824300	call dword ptr ds:[&KERNEL32.HeapFree]	ntdll.RtlFreeHeap
0042338C	85C0	test eax,eax	
0042338E	75 18	jnz short 吾爱破解.004233A8	
00423390	56	push esi	
00423391	E8 35100000	call 吾爱破解.004243CB	
00423396	8BF0	mov esi,eax	
00423398	FF15 90824300	call dword ptr ds:[&KERNEL32.GetLastError]	ntdll.RtlGetLastWin32Error
0042339E	50	push eax	
0042339F	E8 3A100000	call 吾爱破解.004243DE	
004233A4	59	pop ecx	kernel32.7C817077
004233A5	8906	mov dword ptr ds:[esi],eax	
004233A7	5E	pop esi	kernel32.7C817077
004233A8	5D	pop ebp	kernel32.7C817077

10、易语言非独立编译

地址	HEX 数据	反汇编	注释
00401000	E8 89000000	call 吾爱破解.0040108E	
00401005	50	push eax	
00401006	E8 B5010000	call <jmp.&KERNEL32.ExitProcess>	
0040100B	47	inc edi	
0040100C	65:74 4e	je short 0040105d	
0040100F	65:77 53	ja short 00401065	
00401012	6F	outs dx,dword ptr ds:[esi]	
00401013	636B 00	arpl word ptr ds:[ebx],bp	
00401016	45	inc ebp	
00401017	72 72	jb short 吾爱破解.0040108B	
00401019	6F	outs dx,dword ptr ds:[esi]	
0040101A	72 00	jb short 吾爱破解.0040101C	
0040101C	6B72 6E 6C	imul esi,dword ptr ds:[edx+0x6E],0x6C	
00401020	6e	outs dx,byte ptr ds:[esi]	
00401021	2e:66:6e	outs dx,byte ptr cs:[esi]	
00401024	65:004E 6F	add byte ptr gs:[esi+0x6F],cl	
00401028	74 20	je short 吾爱破解.0040104A	
0040102A	66:6F	outs dx,word ptr ds:[esi]	
0040102C	75 6E	jnz short 吾爱破解.0040109C	
0040102E	64:207468 65	and byte ptr fs:[eax+ebp*2+0x65],dh	
00401033	206B 65	and byte ptr ds:[ebx+0x65],ch	
00401036	72 6E	jb short 吾爱破解.004010A6	
00401038	65:6c	ins byte ptr es:[edi],dx	
0040103A	206C69 62	and byte ptr ds:[ecx+ebp*2+0x62],ch	
0040103E	72 61	jb short 吾爱破解.004010A1	
00401040	72 79	jb short 吾爱破解.004010BB	

11、QT5.4

地址	HEX 数据	反汇编	注释
004019B0	53	push ebx	
004019B1	83EC 18	sub esp,0x18	
004019B4	83D0 30404000	cmp dword ptr ds:[0x404030],0x2	
004019BB	8B4424 24	mov eax,dword ptr ss:[esp+0x24]	52pojie.004019B0
004019BF	74 0A	je short 52pojie.004019CB	
004019C1	C705 30404000	mov dword ptr ds:[0x404030],0x2	
004019CB	83F8 02	cmp eax,0x2	
004019CE	74 11	je short 52pojie.004019E1	
004019D0	83F8 01	cmp eax,0x1	
004019D3	74 3B	je short 52pojie.00401A10	
004019D5	83C4 18	add esp,0x18	
004019D8	B8 01000000	mov eax,0x1	
004019DD	5B	pop ebx	ntdll.7C92118A
004019DE	C2 0C00	ret 0xC	
004019E1	B8 30A04000	mov ebx,52pojie.0040A030	
004019E6	81FB 30A04000	cmp ebx,52pojie.0040A030	
004019EC	74 E7	je short 52pojie.004019D5	
004019EE	66:90	nop	
004019F0	8B03	mov eax,dword ptr ds:[ebx]	
004019F2	85C0	test eax,eax	52pojie.004019B0
004019F4	74 02	je short 52pojie.004019F8	
004019F6	FFD0	call eax	52pojie.004019B0
004019F8	83C3 04	add ebx,0x4	
004019FB	81FB 30A04000	cmp ebx,52pojie.0040A030	
00401A01	75 ED	jnz short 52pojie.004019F0	
00401A03	83C4 18	add esp,0x18	

12、PB

地址	HEX 数据	反汇编	注释
10001B30	55	push ebp	
10001B31	8BEC	mov ebp,esp	
10001B33	6A FF	push -0x1	
10001B35	68 A0500010	push 吾爱破解.100050A0	
10001B3A	68 28300010	push 吾爱破解.10003028	
10001B3F	64:A1 00000000	mov eax,dword ptr fs:[0]	
10001B45	50	push eax	
10001B46	64:8925 000000	mov dword ptr fs:[0],esp	
10001B4D	83C4 A8	add esp,-0x58	
10001B50	53	push ebx	
10001B51	56	push esi	
10001B52	57	push edi	
10001B53	8965 E8	mov dword ptr ss:[ebp-0x18],esp	
10001B56	FF15 24500010	call dword ptr ds:[<&KERNEL32.GetVersion	kernel32.GetVersion
10001B5C	33D2	xor edx,edx	ntdll.KiFastSystemCallRet
10001B5E	8AD4	mov dl,ah	
10001B60	8915 24870010	mov dword ptr ds:[0x10008724],edx	ntdll.KiFastSystemCallRet
10001B66	8BC8	mov ecx,eax	
10001B68	81E1 FF000000	and ecx,0xFF	
10001B6E	890D 20870010	mov dword ptr ds:[0x10008720],ecx	
10001B74	C1E1 08	shl ecx,0x8	
10001B77	03CA	add ecx,edx	ntdll.KiFastSystemCallRet
10001B79	890D 1C870010	mov dword ptr ds:[0x1000871C],ecx	
10001B7F	C1E8 10	shr eax,0x10	
10001B82	A3 18870010	mov dword ptr ds:[0x10008718],eax	
10001B87	E8 64130000	call 吾爱破解.10002EF0	

12、Autolt_v3

地址	HEX 数据	反汇编	注释
00425F74	E8 6ACE0000	call 吾爱破解.00432DE3	
00425F79	E9 7FEEFFFF	jmp 吾爱破解.00425DFD	
00425F7E	CC	int3	
00425F7F	CC	int3	
00425F80	57	push edi	
00425F81	56	push esi	
00425F82	8B7424 10	mov esi,dword ptr ss:[esp+0x10]	
00425F86	8B4C24 14	mov ecx,dword ptr ss:[esp+0x14]	
00425F8A	8B7C24 0C	mov edi,dword ptr ss:[esp+0xC]	
00425F8E	8BC1	mov eax,ecx	
00425F90	8BD1	mov edx,ecx	
00425F92	03C6	add eax,esi	
00425F94	3BFE	cmp edi,esi	
00425F96	76 08	jbe short 吾爱破解.00425FA0	
00425F98	3BF8	cmp edi,eax	
00425F9A	0F82 68030000	jb 吾爱破解.00426308	
00425FA0	0FBA25 58014C0	bt dword ptr ds:[0x4C0158],0x1	
00425FA8	73 07	jnb short 吾爱破解.00425FB1	
00425FAA	F3:A4	rep movs byte ptr es:[edi],byte ptr ds:	
00425FAC	E9 17030000	jmp 吾爱破解.004262C8	
00425FB1	81F9 80000000	cmp ecx,0x80	
00425FB7	0F82 CE010000	jb 吾爱破解.0042618B	
00425FBD	8BC7	mov eax,edi	
00425FBF	33C6	xor eax,esi	
00425FC1	A9 0F000000	test eax,0xF	
00425FC6	75 0E	jnz short 吾爱破解.00425FD6	