

# 【jarvisoj刷题之旅】逆向题目软件密码破解-1的writeup

原创

iqiqiya 于 2018-09-10 20:26:24 发布 1000 收藏

分类专栏: [我的逆向之路](#) [我的CTF之路](#) [我的CTF进阶之路](#) 文章标签: [REVERSE](#) [【jarvisoj刷题之旅】逆向题目软件密码破解-1的wri](#) [软件密码破解-1](#) [writeup](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xiangshangbashaonian/article/details/82595483>

版权



[我的逆向之路](#) 同时被 3 个专栏收录

108 篇文章 10 订阅

订阅专栏



[我的CTF之路](#)

92 篇文章 5 订阅

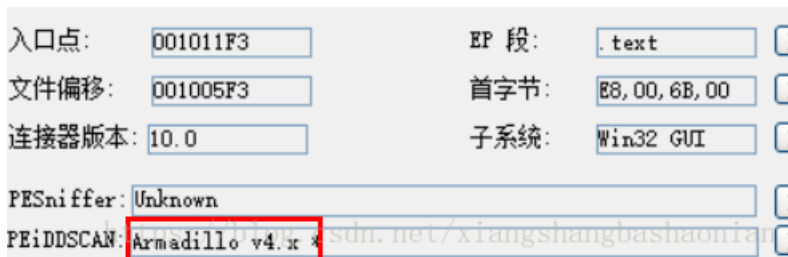
订阅专栏

[我的CTF进阶之路](#)

108 篇文章 18 订阅

订阅专栏

刚开始PEiD查到



百度了一下说是一个壳



把我吓坏了 没见过这玩意

找了几个脱壳机发现都没用

于是直接载入IDA 但好多函数 看不懂 又载入OD

中文搜索引擎发现“你赢了”

于是在段首下断

下图是正读取输入123456789

吾要破解 - CTF\_100\_0.exe - [LCG - 主线程, 模块 - CTF\_100]

002C1B83 - 81EC AC000000 sub esp,0x4C  
 002C1B89 - A1 10FE4200 mov eax,dword ptr ds:[0x2FE10]  
 002C1B8E - 33C5 xor eax,ebx  
 002C1BC0 - 8945 FC mov [local.1],eax CTF\_100\_.004379F8  
 002C1BC3 - 53 push ebx CTF\_100\_.00410790  
 002C1BC4 - 56 push esi  
 002C1BC5 - 57 push edi  
 002C1BC6 - 898D 58FFFFFF mov [local.42],ecx CTF\_100\_.004379F8  
 002C1BC7 - 50 push eax  
 002C1BCD - 64:A1 180000 mov eax,dword ptr fs:[0x18]  
 002C1BD3 - 8B48 30 mov eax,dword ptr ds:[eax+0x30]  
 002C1BD6 - 0FB40 02 movzx eax,byte ptr ds:[eax+0x2]  
 002C1BD8 - 8985 54FFFFFF mov [local.43],eax CTF\_100\_.004379F8  
 002C1BE0 - 58 pop eax 00AFF14C  
 002C1BE1 - 83BD 54FFFFFF cmp [local.43],0x0  
 002C1BE8 - 74 08 jb short CTF\_100\_.002C1BF2  
 002C1BEA - 6A 00 push 0x0  
 002C1BEC - FF15 2C22E000 call dword ptr ds:[<<KERNEL32.ExitProcess  
 002C1BF2 - 68 A0000000 push 0x0  
 002C1BF7 - 8D85 5CFFFFFF lea eax,[local.41]  
 002C1BFD - 6A 00 push 0x0  
 002C1BF8 - 50 push eax CTF\_100\_.004379F8  
 002C1C00 - E8 6B0A1000 call CTF\_100\_.003C2670  
 002C1C05 - B8 F8794300 mov eax,CTF\_100\_.004379F8 UNICODE "123456789"  
 002C1C0A - 83C4 0C add esp,0x4  
 002C1C0D - 8D50 02 lea edx,dword ptr ds:[eax+0x2]  
 002C1C10 - 66:8B08 mov cx,word ptr ds:[eax]  
 002C1C13 - 83C0 02 add eax,0x2  
 002C1C16 - 66:85C9 test cx,cx  
 002C1C19 - 75 F5 jnz short CTF\_100\_.002C1C10  
 002C1C1B - 2BC2 sub eax,edx  
 002C1C1D - D1F8 sar eax,1  
 002C1C1F - 8BF0 mov esi,eax CTF\_100\_.004379F8  
 002C1C21 - 83FE 0E cmp esi,0xE  
 002C1C24 - 0F8F 46010000 jg CTF\_100\_.002C1D70  
 002C1C2A - 8D7E 01 lea edi,dword ptr ds:[esi+0x1]

寄存器 (EIP)  
 EAX 00000000 UNICODE "123456789"  
 ECX 00000000  
 EDX 00000000  
 EBX 00000001  
 ESP 00AFF12C ASCII "L"癡"  
 EBP 00AFF1F0  
 ESI 00410790 CTF\_100\_.00410790  
 EDI 00000111  
 EIP 002C1C00 CTF\_100\_.002C1C00  
 C 0 ES 0028 32位 0(FFFFFFFF)  
 P 1 CS 0023 32位 0(FFFFFFFF)  
 A 0 SS 002B 32位 0(FFFFFFFF)  
 Z 1 DS 0028 32位 0(FFFFFFFF)  
 S 0 FS 0053 32位 0(FFFFFFFF)  
 T 0 GS 0028 32位 0(FFFFFFFF)  
 D 0  
 O 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty 0.0  
 ST1 empty 0.0  
 ST2 empty 0.0  
 ST3 empty 0.0  
 ST4 empty 1.00000000000000000000  
 ST5 empty 1.00000000000000000000  
 ST6 empty 1.00000000000000000000  
 ST7 empty 1.00000000000000000000  
 3 2 1 0 E S P U 0 2 0  
 FST 4020 Cond 1 0 0 0 Err 0 0 1 0 0 0 0  
 FCW 027F Prec NEAR,53 掩码 1 1 1 1 1

地址 HEX 数据 ASCII  
 004379F8 31 00 32 00 33 00 34 00 35 00 36 00 37 00 38 00 1.2.3.4.5.6.7.8.  
 00437A08 39 00 00 00 00 00 00 00 00 00 00 00 00 00 00 9.....  
 00437A18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
 00437A28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
 00AFF12C 00AFF14C  
 CTF\_100\_.00410790 https://blog.csdn.net/xiangshangbashaonian

002C1C05 - B8 F8794300 mov eax,CTF\_100\_.004379F8 UNICODE "123456789"  
 002C1C0A - 83C4 0C add esp,0x4  
 002C1C0D - 8D50 02 lea edx,dword ptr ds:[eax+0x2]  
 002C1C10 - 66:8B08 mov cx,word ptr ds:[eax]  
 002C1C13 - 83C0 02 add eax,0x2  
 002C1C16 - 66:85C9 test cx,cx  
 002C1C19 - 75 F5 jnz short CTF\_100\_.002C1C10  
 002C1C1B - 2BC2 sub eax,edx  
 002C1C1D - D1F8 sar eax,1  
 002C1C1F - 8BF0 mov esi,eax  
 002C1C21 - 83FE 0E cmp esi,0xE  
 002C1C24 - 0F8F 46010000 jg CTF\_100\_.002C1D70  
 002C1C2A - 8D7E 01 lea edi,dword ptr ds:[esi+0x1]

把1给cx  
 eax+2  
 按位异或  
 CTF\_100\_.00437AE3  
 上一个 是 eax-1 这个是算术右移  
 esi存放的就是len(input) = 9 输入位数应该小于E 14  
 https://blog.csdn.net/xiangshangbashaonian

下面是重要的一个地方

002C1C51 - 7E 16 jle short CTF\_100\_.002C1C69  
 002C1C53 - B9 F8774300 mov ecx,CTF\_100\_.004377F8 UNICODE "在此输入口令: "  
 002C1C58 - 8BC3 mov eax,ebx  
 002C1C5A - 2BC8 sub ecx,ebx  
 002C1C5C - 8D6424 00 lea esp,dword ptr ss:[esp]  
 002C1C60 - 8A1401 mov dl,byte ptr ds:[ecx+eax] 先将每一位取出分别与123456789进行异或  
 002C1C63 - 3010 xor byte ptr ds:[eax],dl  
 002C1C65 - 40 inc eax  
 002C1C66 - 4E dec esi  
 002C1C67 - 75 F7 jnz short CTF\_100\_.002C1C60  
 002C1C69 - 8138 1B1C1740 cmp dword ptr ds:[edx],0x46171C1B  
 002C1C6F - 0F85 E7000000 jnz CTF\_100\_.002C1D5C  
 002C1C75 - 817B 04 E4ED1D00 cmp dword ptr ds:[ebx+0x4],0x3020EDEF4  
 eax++  
 esi-- 这里相当于计数器  
 https://blog.csdn.net/xiangshangbashaonian

ds:[00437801]=53 ('S')  
 dl=E3  
 跳转来自 002C1C67

地址	HEX 数据	ASCII
004377E1	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
004377F1	00 00 00 00 00 00 00 28 57 64 6B 93 8F 65 51 E3	.....(Wdk搞EQ?)
00437801	53 E4 4E 1A FF 00 00 00 00 00 00 00 00 00 00 00	S 銷ij.....
00437811	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00437821	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

https://blog.csdn.net/xiangshangbashaonian

将与我们的input对比的数据抠出来

004377F1 28 57 64 6B 93 8F 65 51 E3 (Wdk搞eQ?)

00437801 53 E4 4E 1A FF S銷□.

整理一下得到28,57,64,6B,93,8F,65,51,E3,53,E4,4E,1A,FF

正好14个数据

下面是接下来将异或后的结果分开 分别与这些作比较

002C1C65	- 40	inc eax	eax++
002C1C66	- 4E	dec esi	esi--
002C1C67	- ^ 75 F7	inzb short CTF_100_002C1C60	这里相当于计数器
002C1C69	> 813B 1B1C1746	cmp dword ptr ds:[ebx],0x46171C1B	
002C1C6F	- ^ 0F85 E7000000	inzb CTF_100_002C1D5C	
002C1C75	- 817B 04 F4FD2	cmp dword ptr ds:[ebx+0x4],0x3020FDF4	
002C1C7C	- ^ 0F85 DA000000	inzb CTF_100_002C1D5C	
002C1C82	- 817B 08 B70C8	cmp dword ptr ds:[ebx+0x8],0x7E8E0CB7	
002C1C89	- ^ 0F85 CD000000	inzb CTF_100_002C1D5C	
002C1C8F	- 807B 0C 78	cmp byte ptr ds:[ebx+0xC],0x78	
002C1C93	- ^ 0F85 C3000000	inzb CTF_100_002C1D5C	
002C1C99	- 807B 0D DE	cmp byte ptr ds:[ebx+0xD],0xDE	
002C1C9D	- ^ 0F85 B9000000	inzb CTF_100_002C1D5C	
002C1CA3	- 8D85 5CFFFFFF	lea eax,[local.41]	
002C1CA9	- 83C0 FE	add eax,-0x2	
002C1CAC	- 8D6424 00	lea esp,dword ptr ss:[esp]	
002C1CB0	> 66:8B48 02	mov cx,word ptr ds:[eax+0x2]	
002C1CB4	- 83C0 02	add eax,0x2	
002C1CB7	- 66:85C9	test cx,cx	
002C1CBA	- ^ 75 F4	inzb short CTF_100_002C1CB0	
002C1CBC	- 8B0D 9807410	mov ecx,dword ptr ds:[0x410798]	{FLAG:
002C1CC2	- 8B15 9C07410	mov edx,dword ptr ds:[0x41079C]	LAG:

跳转已实现 <https://blog.csdn.net/xiangshangbashaonian>

将这些数据也抠出来

002C1C69 |> \813B 1B1C1746 cmp dword ptr ds:[ebx],0x46171C1B

002C1C75 |. 817B 04 F4FD2>cmp dword ptr ds:[ebx+0x4],0x3020FDF4

002C1C82 |. 817B 08 B70C8>cmp dword ptr ds:[ebx+0x8],0x7E8E0CB7

002C1C8F |. 807B 0C 78 cmp byte ptr ds:[ebx+0xC],0x78

002C1C99 |. 807B 0D DE cmp byte ptr ds:[ebx+0xD],0xDE

46,17,1C,1B 30,20,FD,F4 7E,8E,0C,B7 78,DE

但是这里顺序是相反的 我们可以选中那一行 右键数据窗口跟随—》选择就可以明白

```

002C1C67 | . ^ 75 F7 | jnz short CTF_100_002C1C68
002C1C69 | > 813B 1B1C174 | cmp dword ptr ds:[ebx], 0x46171C1B
002C1C6F | . ~ 0F85 E700000 | jnz CTF_100_002C1D5C
002C1C75 | . 817B 04 F4FD | cmp dword ptr ds:[ebx+0x4], 0x3020FDF4
002C1C7C | . ~ 0F85 DA00000 | jnz CTF_100_002C1D5C
002C1C82 | . 817B 08 B70C | cmp dword ptr ds:[ebx+0x8], 0x7E8E0CB7
002C1C89 | . ~ 0F85 CD00000 | jnz CTF_100_002C1D5C
002C1C8F | . 807B 0C 78 | cmp byte ptr ds:[ebx+0xC], 0x78
002C1C93 | . ~ 0F85 C300000 | jnz CTF_100_002C1D5C
002C1C99 | . 807B 0D DE | cmp byte ptr ds:[ebx+0xD], 0xDE
002C1C9D | . ~ 0F85 B900000 | jnz CTF_100_002C1D5C
002C1CA3 | . 8D85 5CFFFFFF | lea eax,[local.41]
002C1CA9 | . 83C0 FE | add eax,-0x2
002C1CAC | . 8D6424 00 | lea esp,dword ptr ss:[esp]
002C1CB0 | > 66:8B48 02 | mov cx,word ptr ds:[eax+0x2]
002C1CB4 | . 83C0 02 | add eax,0x2
002C1CB7 | . 66:85C9 | test cx,cx
002C1CBA | . ^ 75 F4 | jnz short CTF_100_002C1CB8
002C1CBC | . 8B0D 9807410 | mov ecx,dword ptr ds:[0x410798]
002C1CC2 | . 8B15 9C07410 | mov edx,dword ptr ds:[0x41079C]
002C1CC8 | . 8B15 9C07410 | mov dword ptr ds:[eax+7],edx
002C1CC9 | . 8B15 9C07410 | mov dword ptr ds:[eax+8],edx
ds:[02B19FC0]=5F576519
跳转来自 002C1C51

```

地址	HEX 数据	ASCII
002C1C69	81 3B 1B 1C 17 46	?????F
002C1C79	FD 20 30 0F 85 DA	?000...

所以最后应该是

1B,1C,17,46,F4,FD,20,30,B7,0C,8E,7E,78,DE

也是14位

```

软件密码破解-1.py x
a = [0x28,0x57,0x64,0x6B,0x93,0x8F,0x65,0x51,0xE3,0x53,0xE4,0x4E,0x1A,0xFF]
b = [0x1B,0x1C,0x17,0x46,0xF4,0xFD,0x20,0x30,0xB7,0x0C,0x8E,0x7E,0x78,0xDE]
flag = ''
for i in range(0,14):
    flag += chr(a[i] ^ b[i])
print(flag)

```

```

for i in range(0,14)
: 软件密码破解-1 x
C:\Users\1\PycharmProjects\test\Scripts\python.exe C:/Users/1/Pycharm
3Ks-grEaT_j0b!
Process finished with exit code 0 https://blog.csdn.net/xiangshangbashaonian

```

提交显示正确

软件密码破解-1 109 SOLVERS 100 REVERSE

请对压缩包中的程序进行分析并获取flag。flag形式为xxx-xxxxx\_xxxx。

CTF\_100\_0.rar.b5abee530fee7cdae2f5cdc33bb849e8

You have solved this Challenge! SUBMIT <https://blog.csdn.net/xiangshangbashaonian>



[创作打卡挑战赛](#) >  
[赢取流量/现金/CSDN周边激励大奖](#)