

看雪ctf2017第一题详解

原创

[zsd747289639](#) 于 2017-06-10 17:05:42 发布 1491 收藏 1

分类专栏: [PE逆向笔记](#) 文章标签: [看雪ctf2017](#)

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/zsd747289639/article/details/72993552>

版权



[PE逆向笔记](#) 专栏收录该内容

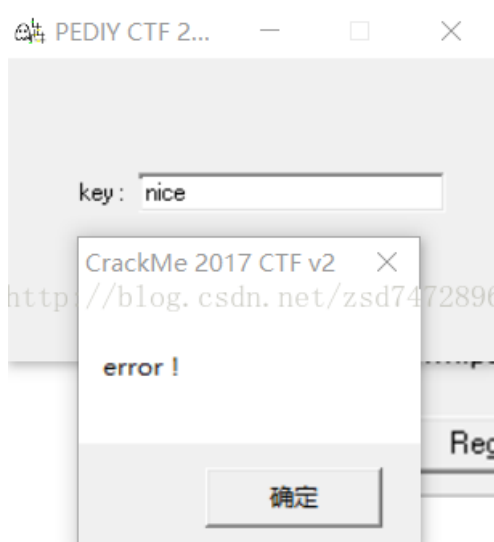
2 篇文章 0 订阅

订阅专栏

第一步、先瞅瞅



随便输入点啥, 发现如下报错



第二步、Od加载程序, 下断点

一般这种程序, 第一感觉就是在下api断点, 右键->查找->当前模块中的名称

找到getDlgItemTextA

0040708C	rdata	输入	KERNEL32.FreeEnvironmentStringsA	
00407090	rdata	输入	KERNEL32.FreeEnvironmentStringsW	
00407028	rdata	输入	KERNEL32.GetACP	
0040706C	rdata	输入	KERNEL32.GetCommandLineA	
0040702C	rdata	输入	KERNEL32.GetCPIInfo	
00407080	rdata	输入	KERNEL32.GetCurrentProcess	
004070E4	rdata	输入	USER32.GetDlgItem	
004070A8	rdata	输入	USER32.GetDlgItemTextA	http://www.csdn.net/zsd747289639
00407060	rdata	输入	KERNEL32.GetEnvironmentStrings	
0040706C	rdata	输入	KERNEL32.GetEnvironmentStringsW	
0040704C	rdata	输入	KERNEL32.GetEnvironmentVariableA	
00407050	rdata	输入	KERNEL32.GetFileType	
004070D4	rdata	输入	USER32.GetMessageA	

右键->在输入函数上切换断点 设置断点

F9运行程序，程序断在了此处

748AD5C0	8BFF	mov edi,edi	
748AD5C2	55	push ebp	
748AD5C3	8BEC	mov ebp,esp	
748AD5C5	FF75 0C	push dword ptr ss:[ebp+0xC]	
748AD5C8	FF75 08	push dword ptr ss:[ebp+0x8]	WannaLOL.00401041
748AD5CB	E8 E0E5F7FF	call user32.GetDlgItem	
748AD5D0	85C0	test eax,eax	
748AD5D2	74 0E	je short user32.748AD5E2	
748AD5D4	FF75 14	push dword ptr ss:[ebp+0x14]	
748AD5D7	FF75 10	push dword ptr ss:[ebp+0x10]	user32.7483D2B3
748AD5DA	50	push eax	
748AD5DB	E8 90E1F7FF	call user32.GetWindowTextA	
748AD5E0	EB 0E	jmp short user32.748AD5F0	
748AD5E2	837D 14 00	cmp dword ptr ss:[ebp+0x14],0x0	
748AD5E6	74 06	je short user32.748AD5EE	http://www.csdn.net/zsd747289639
748AD5E8	8B45 10	mov eax,dword ptr ss:[ebp+0x10]	user32.7483D2B3
748AD5EB	C600 00	mov byte ptr ds:[eax],0x0	
748AD5EE	33C0	xor eax,eax	
748AD5F0	5D	pop ebp	WannaLOL.00401211
748AD5F1	C2 1000	retn 0x10	
748AD5F4	CC	int3	
748AD5F5	CC	int3	
748AD5F6	CC	int3	
748AD5F7	CC	int3	
748AD5F8	CC	int3	
748AD5F9	CC	int3	
748AD5FA	CC	int3	
748AD5FB	CC	int3	

第三步，算法分析

一路f8执行到retn返回，来到

00401211	68 F4010000	push 0xF4	Timeout = 500. ms
00401216	FF15 00704000	call dword ptr ds:[<&KERNEL32.Sleep>]	Sleep
0040121C	8D45 E4	lea eax,dword ptr ss:[ebp-0x1C]	
0040121F	50	push eax	
00401220	E8 DB000000	call WannaLOL.00401300	判断输入的序列号是否为4位
00401225	83F8 04	cmp eax,0x4	
00401228	59	pop ecx	win32u.73BD26FC
00401229	0F85 A0000000	jnz WannaLOL.004012CF	
0040122F	6A 30	push 0x30	
00401231	59	pop ecx	win32u.73BD26FC
00401232	384D E4	cmp byte ptr ss:[ebp-0x1C],cl	
00401235	0F84 94000000	je WannaLOL.004012CF	
00401238	384D E5	cmp byte ptr ss:[ebp-0x18],cl	判断这四个数是否为0
0040123E	0F84 8B000000	je WannaLOL.004012CF	http://www.csdn.net/zsd747289639
00401244	384D E6	cmp byte ptr ss:[ebp-0x14],cl	
00401247	0F84 82000000	je WannaLOL.004012CF	
0040124D	384D E7	cmp byte ptr ss:[ebp-0x19],cl	
00401250	74 7D	je short WannaLOL.004012CF	
00401252	807D E4 31	cmp byte ptr ss:[ebp-0x1C],0x31	判断第一个数是否为1，1的ascii码为31
00401256	75 77	jnz short WannaLOL.004012CF	
00401258	807D E5 35	cmp byte ptr ss:[ebp-0x18],0x35	判断第二个数是否为5
0040125C	75 71	jnz short WannaLOL.004012CF	
0040125E	74 03	je short WannaLOL.00401263	
00401260	75 01	jnz short WannaLOL.00401263	
00401262	E8	db E8	
00401263	66:B8 0800	mov ax,0x8	
00401267	66:35 0700	xor ax,0x7	
0040126D	8B45 E4	mov ecx,byte ptr ss:[ebp-0x1C]	

这里我们看到它调用了wannaLOL.00401300这么个函数，

执行返回后比较eax和4是否相等，这里我们不难想到，这里应该是判断输入的序列号的位数。

这里我们设[ebp-0x1c]=f(1),[ebp-0x1b]=f(2),ebp-0x1a=(3),ebp-0x19=f(4)

紧接着，下面判断这四个数是否为0,然后判断f(1)是否为1，f(2)是否为5。图上已经标注得很清楚了。

这里我们为了进一步调试，需要在判断跳转时候将z标志位置1。在判断f(1)和f(2)时。

```

C 0 ES 002B 32位 0(FFFFFFFF)
P 0 CS 0023 32位 0(FFFFFFFF)
A 0 SS 002B 32位 0(FFFFFFFF)
Z 1 ← 002B 32位 0(FFFFFFFF)
S http://blog.csdn.net/zsd747289639 这里为了下次调试方便，我们将z置为1
T 0 GS 002B 32位 0(FFFFFFFF)
D 0
O 0 LastErr ERROR_SUCCESS (00000000)

```

接着我们便来到了这里，需要进行一些简单的浮点计算,最后和[0x407118]处数值比较

如图，我已经分析得很清楚了

00401280	- 0FBE45 E5	movsx eax,byte ptr ss:[ebp-0x1B]	
00401284	- DB45 FC	Fild dword ptr ss:[ebp-0x4]	将[ebp-0x4]中的整数转为浮点数
00401287	- 2BC1	sub eax,ecx	ecx为30
00401289	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	
0040128C	- DA75 FC	Fdiv dword ptr ss:[ebp-0x4]	将浮点栈道中st(0)/[ebp-0x4],即f(1)/f(2)
0040128F	- 0FBE45 E7	movsx eax,byte ptr ss:[ebp-0x19]	
00401293	- 2BC1	sub eax,ecx	
00401295	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	浮点栈中st(1)-st(0)
00401298	- DEE9	Fsubp st(1),st	
0040129A	- DA4D FC	Fimul dword ptr ss:[ebp-0x4]	st(0)*[ebp-0x4],这里[ebp-0x4]是f(3)
0040129D	- D80D 1C714000	Fmul dword ptr ds:[0x40711C]	st(0)*[0x40711C]根据od的提示，可知为16
004012A3	- D95D FC	Fstp dword ptr ss:[ebp-0x4]	将st(0)存入[ebp-0x4]中
004012A6	- 74 03	je short WannaLOL.004012A8	
004012A8	- 75 01	jnz short WannaLOL.004012A8	

004012B3	- D945 FC	Fld dword ptr ss:[ebp-0x4]	fld是将st(0)中的数据复制到[ebp-0x4]中
004012B6	- D81D 18714000	Fcomp dword ptr ds:[0x407118]	将st(0)处数据和[0x407118]处数据进行比较
004012BC	- 6A 00	push 0x0	
004012BE	- 68 78804000	push WannaLOL.00408078	ASCII "CrackMe 2017 CTF"
004012C3	- DFE0	fstsw ax	
004012C5	- 9E	sahf	http://blog.csdn.net/zsd747289639
004012C6	- 75 0E	jnz short WannaLOL.004012D6	
004012C8	- 68 5C804000	push WannaLOL.0040805C	ASCII "Registration successful !"
004012CD	- EB 0C	jmp short WannaLOL.004012DB	

我们可以整理得这个浮点计算的公式为 (f(3)-0.2)*f(4)*[0x40711c]=[0x407118]

其中根据od的提示，我们可以知道[0x407118]=384，[0x40711c]=16

004012B6	- D81D 18714000	Fcomp dword ptr ds:[0x407118]	
004012BC	- 6A 00	push 0x0	
004012BE	- 68 78804000	push WannaLOL.00408078	ASCII "CrackMe 2017 CTF"
004012C3	- DFE0	fstsw ax	
004012C5	- 9E	sahf	
004012C6	- 75 0E	jnz short WannaLOL.004012D6	
004012C8	- 68 5C804000	push WannaLOL.0040805C	ASCII "Registration successful !"
004012CD	- EB 0C	jmp short WannaLOL.004012DB	
004012CF	- 6A 00	push 0x0	
004012D1	- 68 78804000	push WannaLOL.00408078	ASCII "CrackMe 2017 CTF"
st=51.0000000000000000		这个数为384	
ds:[00407118]=384.0000			

```

0040129D - D80D 1C71400 fmul dword ptr ds:[0x40711C]
004012A3 - D95D FC fstp dword ptr ss:[ebp-0x4]
004012A6 - 74 03 je short WannaLOL.004012A8
004012A8 - 75 01 jnz short WannaLOL.004012AB
004012AA - E8 db E8
004012AB - 66:B8 0800 mov ax,0x8
004012AF - 66:35 0700 xor ax,0x7
004012B3 - D945 FC fld dword ptr ss:[ebp-0x4]
004012B6 - D81D 1871400 fcomp dword ptr ds:[0x407118]
004012BC - 6A 00 push 0x0
004012BE - 68 78804000 push WannaLOL.00408078
004012C3 - DFE0 fstsw ax
004012C5 - 9E sahf
004012C6 - 75 0E jnz short WannaLOL.004012D6
004012C8 - 68 5C804000 push WannaLOL.0040805C
004012CB - 75 0E jnz short WannaLOL.004012D6
st=51.0000000000000000
ds:[0040711C]=16.00000

```

这个数为16

所以公式整理得 $(f(3)-0.2)*f(4)=24$ ，易估得， $f(3)=5, f(4)=5$

最后得序列号为：1555

输入验证下：



破解成功！



[创作打卡挑战赛](#)
赢取流量/现金/CSDN周边激励大奖