

看雪.Wifi万能钥匙 2017CTF年中赛---第一题

原创

Angelki 于 2018-09-29 10:49:30 发布 504 收藏

分类专栏: CTF 文章标签: CTF

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/qq_15727809/article/details/82893603

版权



[CTF 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

考察浮点数运算

1、OD载入, 搜索字符串, 查找到错误或正确提示信息, 双击点进去。

```
004010EE mov ebx,WannaLOL.00407108 myWindowClass
0040114F push WannaLOL.00408030 PEDIY CTF 2017
004012BE push WannaLOL.00408078 CrackMe 2017 CTF
004012C8 push WannaLOL.0040805C Registration successful !
004012D1 push WannaLOL.00408048 CrackMe 2017 CTF v2
004012D6 push WannaLOL.00408040 error !
004013A2 mov dword ptr ds:[0x4080AC],eax u4@
004013C5 mov dword ptr ds:[0x4080C0],eax u4@
```

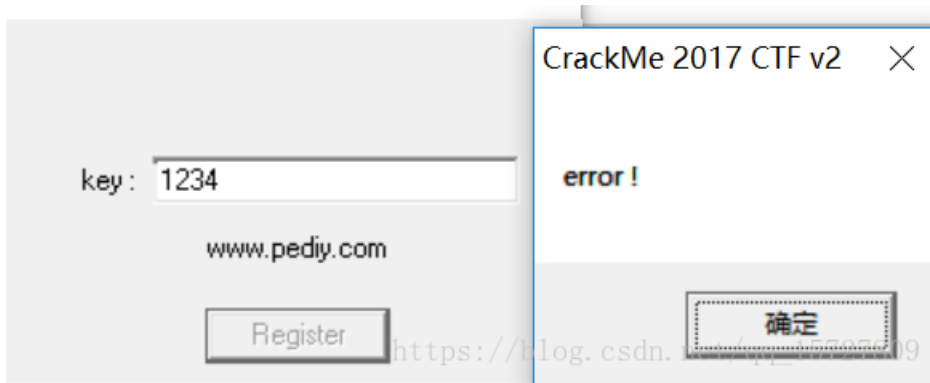
找到以为的关键跳转, 下断点, 重新载入, 输入注册码, 运行, 竟然提示错误!

```
004012C6 jnz short WannaLOL.004012D6 关键跳转
004012C8 - 68 5C804000 push WannaLOL.0040805C Registration successful !
004012CD - EB 0C jmp short WannaLOL.004012DB
004012CF > 6A 00 push 0x0
004012D1 - 68 48804000 push WannaLOL.00408048 CrackMe 2017 CTF v2
004012D6 > 68 40804000 push WannaLOL.00408040 error !
004012DB > FF35 34AA4000 push dword ptr ds:[0x40AA34] hOwner = NULL
004012E1 - FF15 AC704000 call dword ptr ds:[&USER32.MessageBoxA] MessageBoxA
004012E7 - C9 leave
004012E8 - C3 retn
```

不死心的将关键跳转NOP掉, 保存到文件, 运行,

```
004012C6 90 nop 关键跳转
004012C7 90 nop
004012C8 - 68 5C804000 push WannaLOL.0040805C Registration successful !
004012CD - EB 0C jmp short WannaLOL.004012DB
004012CF > 6A 00 push 0x0
004012D1 - 68 48804000 push WannaLOL.00408048 CrackMe 2017 CTF v2
004012D6 > 68 40804000 push WannaLOL.00408040 error !
004012DB > FF35 34AA4000 push dword ptr ds:[0x40AA34] hOwner = 00460742
004012E1 - FF15 AC704000 call dword ptr ds:[&USER32.MessageBoxA] MessageBoxA
004012E7 - C9 leave
004012E8 - C3 retn
```

结果依然报错! 果然爆破不行呀.....那就老老实实查看算法吧。



2、在该段开始位置下断点，OD重新载入，运行，输入key，断在段首，仔细分析汇编代码

0040121C	- 8D45 E4	lea eax,dword ptr ss:[ebp-0x1C]	'123456'
0040121F	- 50	push eax	'1589'
00401220	- E8 DB000000	call WannaLOL.00401300	
00401225	- 83F8 04	cmp eax,0x4	eax=6,长度?
00401228	- 59	pop ecx	将字符串写入ECX
00401229	- 0F85 A0000000	jnz WannaLOL.004012CF	
0040122F	- 6A 30	push 0x30	
00401231	- 59	pop ecx	ECX = 0x30
00401232	- 384D E4	cmp byte ptr ss:[ebp-0x1C],cl	'1'
00401235	- 0F84 94000000	je WannaLOL.004012CF	
0040123B	- 384D E5	cmp byte ptr ss:[ebp-0x1B],cl	
0040123E	- 0F84 8B000000	je WannaLOL.004012CF	
00401244	- 384D E6	cmp byte ptr ss:[ebp-0x1A],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	
00401256	- 75 77	jnz short WannaLOL.004012CF	
00401258	- 807D E5 35	cmp byte ptr ss:[ebp-0x1B],0x35	
0040125C	- 75 71	jnz short WannaLOL.004012CF	
0040125E	- 74 03	je short WannaLOL.00401263	
00401260	- 75 01	jnz short WannaLOL.00401263	

https://blog.csdn.net/qq_15727809

发现程序，首先判断输入长度是否等于4，如果不是则直接报错；接着将4个字符分别存入[ebp-0x1C]、[ebp-0x1B]、[ebp-0x1A]、[ebp-0x19]；然后将第1个字符与“1”比较，第2个字符与“5”比较，如果不等则报错。

也即key的前2个字符是15

3、继续向下分析，这次我输入的是“1589”

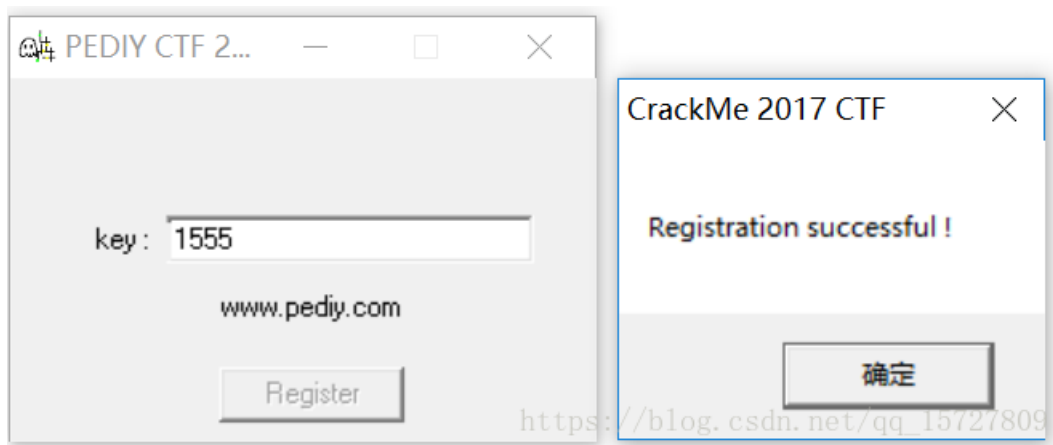
00401263	> 66:B8 0800	mov ax,0x8	
00401267	- 66:35 0700	xor ax,0x7	
0040126B	- 0FBE45 E6	movsx eax,byte ptr ss:[ebp-0x1A]	'8' = 0x38
0040126F	- 2BC1	sub eax,ecx	ecx = 0x30
00401271	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	第3个字符 '8'
00401274	- 0FBE45 E4	movsx eax,byte ptr ss:[ebp-0x1C]	'1' = 0x31 = eax
00401278	- DB45 FC	fld dword ptr ss:[ebp-0x4]	
0040127B	- 2BC1	sub eax,ecx	eax = 1-0=1
0040127D	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	
00401280	- 0FBE45 E5	movsx eax,byte ptr ss:[ebp-0x1B]	eax = 0x35 = '5'
00401284	- DB45 FC	fld dword ptr ss:[ebp-0x4]	
00401287	- 2BC1	sub eax,ecx	eax = 5-0=5
00401289	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	
0040128C	- DA75 FC	fdiv dword ptr ss:[ebp-0x4]	1.0 / 5.0 = 0.2
0040128F	- 0FBE45 E7	movsx eax,byte ptr ss:[ebp-0x19]	eax = 第4个字符 '9'
00401293	- 2BC1	sub eax,ecx	eax = 9-0=9
00401295	- 8945 FC	mov dword ptr ss:[ebp-0x4],eax	
00401298	- DEE9	fsubp st(1),st	第3个字符-0.2
0040129A	- DA4D FC	fimul dword ptr ss:[ebp-0x4]	结果*第4个字符; 7.79999*9
0040129D	- D80D 1C71400	fmul dword ptr ds:[0x40711C]	得到的结果*16.0; 70.2*16.0
004012A3	- D95D FC	fstp dword ptr ss:[ebp-0x4]	1123.2
004012A6	~ 74 03	je short WannaLOL.004012AB	
004012A8	~ 75 01	jnz short WannaLOL.004012AB	
004012AA	- E8	db E8	https://blog.csdn.net/qq_15727809

发现流程是这样的：首先把1和5转为浮点数1.0和5.0，然后相除得0.2；接着用（第3个字符-0.2）第4个字符16.0，得到结果。

4、继续向下分析，发现程序最终是将刚得到的结果与384.0比较，并将结果存入ax，并根据ax的值判断是否进行跳转

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]	1123.199
004012BC	- 6A 00	push 0x0	st=1123.199和384.0进行比较
004012BE	- 68 78804000	push WannaLOL.00408078	CrackMe 2017 CTF
004012C3	- DFE0	fstsw ax	将ah的值传入flag低8位
004012C5	- 9E	sahf	关键跳转
004012C6	~ 75 0E	jnz short WannaLOL.004012D6	Registration successful !
004012C8	- 68 5C804000	push WannaLOL.0040805C	
004012CD	~ EB 0C	jmp short WannaLOL.004012DB	
004012CF	> 6A 00	push 0x0	
004012D1	- 68 48804000	push WannaLOL.00408048	CrackMe 2017 CTF v2
004012D6	> 68 40804000	push WannaLOL.00408040	error !
004012DB	> FF35 34AA400	push dword ptr ds:[0x40AA34]	howner = 00090844 ('PEDIY CTF 2017
004012E1	- FF15 AC70400	call dword ptr ds:[<&USER32.MessageBoxA]	MessageBoxA
004012E7	- C9	leave	https://blog.csdn.net/qq_15727809

5、因此我们只要保证最后计算出的结果为384.0就可以了！很容易算出key=1555，运行验证！



6、因此注册机为：

```
# -*- coding:utf-8 -*-
while(1):
    str = raw_input("")
    #print type(str)
    if len(str) == 4:
        result = (int(str[2], 10)-0.2)*int(str[3], 10)*16.0
        if result == 384.0:
            print "Registration successful!"
        else:
            print "error!"
    else:
        print "error!"
```

https://blog.csdn.net/qq_15727809

```
1234 ..
error!
1555
Registration successful!
```