

# 2015\_NSCTF\_Reverse1\_writeup

转载

[weixin\\_30569153](#) 于 2015-10-12 11:35:00 发布 59 收藏

原文链接: <http://www.cnblogs.com/Viwill/p/4871164.html>

版权

链接: <http://pan.baidu.com/s/1x6Ywm> 密码: j4we

OD加载程序 alt+M, 对rsrc区段F2下断

执行到窗口提示输入: 输入如图字符串nsF0cuS!x01

```
C:\Documents and Settings\Administrator\桌面\Reverse01.exe
please input ns-ctf password: nsF0cuS!x01
flag:<NSCTF_md5065ca>01??ab7e0f4>>a701c>cd17340>
```

但这是错误的==。

如何找到正确flag?

1、OD加载程序, F9运行起来, 此时弹出窗口提示输入密码。OD右键“查看”——>“所有参考文本字符串”:

地址	反汇编	文本字符串
004010A6	push Reverse0.00402150	ASCII "please input ns-ctf password: "
004010BA	push Reverse0.00402170	ASCII "%s"
004010CF	push Reverse0.00402110	ASCII "nsF0cuS!x01"
004010E1	push Reverse0.00402174	ASCII "try again!\n"
004010FB	push Reverse0.00402150	ASCII "please input ns-ctf password: "
00401109	push Reverse0.00402170	ASCII "%s"
0040111A	push Reverse0.00402110	ASCII "nsF0cuS!x01"
00401168	push Reverse0.0040211C	ASCII "flag:<NSCTF_md5065ca>01??ab7e0f4>>a701c>cd17340>
00401180	push 0x10000	UNICODE "=:::\0"

如果首次运行没有看到字符串, 跳到“%s”的地方, 重新分析下代码就好。

2、下面是进行第一判定的函数, 如果输入字符串“nsF0cuS!x01”则第一轮验证通过, 为了方便调试, 我们输入字符串之前在004010DF处设断点, 接着输入程序期望的字符串“nsF0cuS!x01”, 回车后程序断在004010DF1。

```
00401086 |. 68 FF000000 push 0xFF ;/n = FF (255.)
0040108B |. 8D85 FDFFFFFF lea eax,dword ptr ss:[ebp-0x103] ;|
00401091 |. C685 FCFFFFFF >mov byte ptr ss:[ebp-0x104],0x0 ;|
00401098 |. 6A 00 push 0x0 ;|c = 00
0040109A |. 50 push eax ;|s
0040109B |. E8 14090000 call Reverse0.004019B4 ;|memset
004010A0 |. 8B35 94204000 mov esi,dword ptr ds:[0x402094] ;|msvcr120.printf
004010A6 |. 68 50214000 push Reverse0.00402150 ;/format = "please input ns-ctf password: "
004010AB |. FFD6 call esi ;|printf
004010AD |. 8B1D 90204000 mov ebx,dword ptr ds:[0x402090] ;|msvcr120 scanf_s
004010B3 |. 8D85 FCFFFFFF lea eax,[local.65]
004010B9 |. 50 push eax
004010BA |. 68 70214000 push Reverse0.00402170 ; ASCII "%s"
004010BF |. FFD3 call ebx
004010C1 |. 6A 0B push 0xB ;/maxlen = B (11.)
004010C3 |. 8D85 FCFFFFFF lea eax,[local.65] ;|
004010C9 |. BF 01000000 mov edi,0x1 ;|初始化edi=1
004010CE |. 50 push eax ;|s2
004010CF |. 68 10214000 push Reverse0.00402110 ;|s1 = "nsF0cuS!x01"
004010D4 |. FF15 9C204000 call dword ptr ds:[0x40209C] ;|strcmp
004010DA |. 83C4 24 add esp,0x24
004010DD |. 85C0 test eax,eax
004010DF |. 74 4B je XReverse0.0040112C ; 判断部分, 验证通过跳转, 跳转到
004010E1 |. 68 74214000 /push Reverse0.00402174 ; ASCII "try again!\n", 下面这一段为输错之后要重新输入
执行的代码
004010E6 |. FFD6 |call esi
004010E8 |. 68 00010000 |push 0x100 ;/n = 100 (256.)
004010ED |. 8D85 FCFFFFFF |lea eax,[local.65] ;|
004010F3 |. 6A 00 |push 0x0 ;|c = 00
004010F5 |. 50 |push eax ;|s
004010F6 |. E8 B9080000 |call Reverse0.004019B4 ;|memset
004010FB |. 68 50214000 |push Reverse0.00402150 ; ASCII "please input ns-ctf password: "
00401100 |. FFD6 |call esi
```

```

00401102 |. 8D85 FCFEFFFF |lea eax,[local.65]
00401108 |. 50          |push eax
00401109 |. 68 70214000 |push Reverse0.00402170          ; ASCII "%s"
0040110E |. FFD3       |call ebx
00401110 |. 6A 0B      |push 0xB                          ;/maxlen = B (11.)
00401112 |. 8D85 FCFEFFFF |lea eax,[local.65]              ;|
00401118 |. 47         |inc edi                            ;| 出错一次edi+1
00401119 |. 50         |push eax                          ;|s2
0040111A |. 68 10214000 |push Reverse0.00402110          ;|s1 = "nsF0cuS!x01"
0040111F |. FF15 9C204000 |call dword ptr ds:[0x40209C]    ;|strncmp 比较函数
00401125 |. 83C4 28    |add esp,0x28
00401128 |. 85C0       |test eax,eax
0040112A |.^ 75 B5     |jnz XReverse0.004010E1          ; 验证失败跳转到“try again!\n”

```

上面的函数很清晰明了了，我们看看验证通过之后程序还会做些什么。

3、F8单步运行，程序跳转到这里：

```

0040112C |> \8D8D FCFEFFFF lea ecx,[local.65]          ; F8单步运行，这句之后寄存器窗口查看ECX=0012FE7C,保
存了“nsF0cuS!x01”
00401132 |. C705 68334000>mov dword ptr ds:[0x403368],0x1
0040113C |. 8D51 01     |lea edx,dword ptr ds:[ecx+0x1]    ; edx=“sF0cuS!x01”
0040113F |. 90         |nop
00401140 |> 8A01       |mov al,byte ptr ds:[ecx]
00401142 |. 41         |jnc ecx
00401143 |. 84C0       |test al,al
00401145 |.^ 75 F9     |jnz XReverse0.00401140
00401147 |. 2BCA      |sub ecx,edx                      ; 计算“sF0cuS!x01”长度
00401149 |. 74 27     |je XReverse0.00401172
0040114B |. 83FF 03    |cmp edi,0x3                      ; edi表示判断的次数（上面一段程序中有定义），这里将edi和3
比较
0040114E |. 7E 18     |jle XReverse0.00401168          ; edi=<=3跳转
00401150 |. E8 ABFEFFFF |call Reverse0.00401000

```

4.上面的0040114B一行是关键的一行，这里将edi和3进行比较，当小于等于3的时候跳到下面：

```

00401168 |> \68 1C214000 |push Reverse0.0040211C ; ASCII "flag:{NSCTF_md5065ca>01??ab7e0f4>>a701c>cd17340}"
0040116D |. FFD6       |call esi

```

到这里，屏幕打印出"flag:{NSCTF\_md5065ca>01??ab7e0f4>>a701c>cd17340}"，高高兴兴地拿去提交吧  
--。可是这还不是正确答案，为什么呢？这串字符看起来像是加密过的，于是联想到0040114B这行代码，如果edi>3会怎么样呢？edi又是什么？我们往回找edi，在004010C9行将edi初始化为1，00401118一行，edi在比较函数里面出现了，出错一次edi+1,即edi代表验证的次数，照此推理，只需要验证次数达到4次，0040114E这一行就不会跳转。我们来验证一下。

5.重新运行程序，在00401118、0040114B处下断点，输入3次错误密码，程序运行后edi=4，与3比较后进入函数Reverse0.00401000：

```

C:\Documents and Settings\Administrator\桌
please input ns-ctf password: 123
try again!
please input ns-ctf password: 456
try again!
please input ns-ctf password: 789
try again!
please input ns-ctf password: nsF0cuS!x01
_

```

00401108	-. 50	push eax		EBX 10083347 m
00401109	-. 68 70214000	push Reverse0.0040217	ASCII "%s"	ESP 0012FE4C
0040110E	-. FFD3	call ebx		EBP 0012FF80
00401110	-. 6A 0B	push 0xB	/maxlen = B (11.)	ESI 10082FD9 m
00401112	-. 8D85 FCFEFFFF	lea eax,[local.65]		EDI 00000004
00401118	-. 47	inc edi		EIP 00401119 R
00401119	-. 50	push eax	s2 = "nsF0cuS!x01"	C 0 ES 0023 3
0040111A	-. 68 10214000	push Reverse0.0040211	s1 = "nsF0cuS!x01"	P 0 CS 001B 3
0040111F	-. FF15 9C204000	call dword ptr ds:[0x	strncmp	

6.在函数 Reverse0.00401000 里找到了解密函数，鼠标选中00401043，数据窗口跟随内存地址，可以看到内存中的数据变化。

```

00401021 |. 6A 30      |push 0x30

```

```

00401023 |. 68 1C214000 push Reverse0.0040211C ; ASCII "flag:{NSCTF_md5065ca>01??
ab7e0f4>>a701c>cd17340}"
00401028 |. 8D45 C8 lea eax,[local.14]
0040102B |. 6A 31 push 0x31
0040102D |. 50 push eax
0040102E |. FF15 8C204000 call dword ptr ds:[0x40208C] ; msvcrt120.strncpy_s
00401034 |. 83C4 1C add esp,0x1C
00401037 |. 8D45 D7 lea eax,dword ptr ss:[ebp-0x29]
0040103A |. 807D D7 7D cmp byte ptr ss:[ebp-0x29],0x7D
0040103E |. 74 0B je XReverse0.0040104B
00401040 |> 8030 07 /xor byte ptr ds:[eax],0x7
00401043 |. 8D40 01 |lea eax,dword ptr ds:[eax+0x1]
00401046 |. 8038 7D |cmp byte ptr ds:[eax],0x7D
00401049 |.^ 75 F5 |jnz XReverse0.00401040 ; 解密算法，每个字节与7抑或，遇到}结束。

```

地址	HEX 数据	ASCII
0012FE3B	37 31 32 64 66 39 37 36 38 38 66 65 30 62 37 61	712df97688fe0b7a
0012FE4B	33 39 39 66 30 37 36 64 39 64 63 36 30 34 33 37	399f076d9dc60437
0012FE5B	7D 00 FE 12 00 2E E0 A6 8F 80 FF 12 00 55 11 40	}?...喉沿ij.U@a
0012FE6B	00 00 00 00 00 01 00 00 00 00 00 00 00 58 2D 93	.....x-7

到这里答案就出来啦，将原字符串“NSCTF\_md5”后的字符串替换成解密后的字符就对啦~~  
flag:{NSCTF\_md5 712df97688fe0b7a399f076d9dc60437}

ps:初学逆向，欢迎指正！感谢Lnju的指点（<http://www.plaype.cc/>）~~~  
by.Vi  
转载于:<https://www.cnblogs.com/Viwilla/p/4871164.html>