

《网络攻防实战》实验报告

第 13 次实验： lab13

姓名： 佐藤汉

学号： 215220029

21 级 计算机科学与技术系

邮箱： 2868135471@qq.com

时间： 7h（第一题卡了一下）

一、实验目的

得到 task1,task2 的 flags

二、实验内容

TASK1

需要修改 5 个地方

61 0D 66 66 00 00 00 00 00 00 00 00 00 00 00 00	61 0D 0D 0A 00 00 00 00 00 00 00 00 00 00 00 00
66 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	E3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 08 00 00 00 40 00 00 00 73 66 00 00 00 64 00	00 08 00 00 00 40 00 00 00 73 B4 00 00 00 64 00
64 01 6C 00 5A 00 65 01 65 00 6A 02 64 00 19 00	64 01 6C 00 5A 00 65 01 65 00 6A 02 64 00 19 00
64 02 83 02 8F 88 5A 03 65 03 A0 04 A1 00 5A 05	64 02 83 02 8F 88 5A 03 65 03 A0 04 A1 00 5A 05
65 06 64 03 65 07 64 04 65 05 64 05 19 00 41 00	65 06 64 03 65 07 64 04 65 05 64 05 19 00 41 00
83 01 17 00 65 07 64 06 65 05 64 07 19 00 41 00	83 01 17 00 65 07 64 06 65 05 64 07 19 00 41 00
83 01 17 00 65 07 64 08 65 05 64 09 19 00 41 00	83 01 17 00 65 07 64 08 65 05 64 09 19 00 41 00
83 01 17 00 64 0A 17 00 65 07 64 0B 65 05 64 0C	83 01 17 00 64 0A 17 00 65 07 64 0B 65 05 64 0C
19 00 41 00 83 01 17 00 65 07 64 0D 65 05 64 0E	19 00 41 00 83 01 17 00 65 07 64 0D 65 05 64 0E
19 00 41 00 83 01 17 00 65 07 64 0F 65 05 64 10	19 00 41 00 83 01 17 00 65 07 64 0F 65 05 64 10
19 00 41 00 83 01 17 00 64 11 17 00 83 01 01 00	19 00 41 00 83 01 17 00 64 11 17 00 83 01 01 00
57 00 64 01 04 00 04 00 83 03 01 00 6E 10 31 00	57 00 64 01 04 00 04 00 83 03 01 00 6E 10 31 00
73 A6 30 00 01 00 01 00 01 00 59 00 01 00 64 01	73 A6 30 00 01 00 01 00 01 00 59 00 01 00 64 01
53 00 29 12 E9 00 00 00 00 4E DA 02 72 62 7A 66	53 00 29 12 E9 00 00 00 00 4E DA 02 72 62 7A 08
54 72 69 6E 69 74 79 7B E9 7D 00 00 00 E9 02 00	54 72 69 6E 69 74 79 7B E9 7D 00 00 00 E9 02 00
00 00 E9 73 00 00 00 E9 03 00 00 00 E9 80 00 00	00 00 E9 73 00 00 00 E9 03 00 00 00 E9 80 00 00
00 E9 10 00 00 00 DA 01 5F E9 D2 00 00 00 E9 2A	00 E9 10 00 00 00 DA 01 5F E9 D2 00 00 00 E9 2A
00 00 00 E9 61 00 00 00 E9 EF 00 00 00 E9 51 00	00 00 00 E9 61 00 00 00 E9 EF 00 00 00 E9 51 00
00 00 69 3A 01 00 00 FA 01 7D 66 08 DA 03 73 79	00 00 69 3A 01 00 00 FA 01 7D 29 08 DA 03 73 79
73 DA 04 6F 70 65 6E DA 04 61 72 67 76 DA 01 66	73 DA 04 6F 70 65 6E DA 04 61 72 67 76 DA 01 66
DA 04 72 65 61 64 DA 01 78 DA 05 70 72 69 6E 74	DA 04 72 65 61 64 DA 01 78 DA 05 70 72 69 6E 74
DA 03 63 68 72 A9 00 72 18 00 00 00 72 18 00 00	DA 03 63 68 72 A9 00 72 18 00 00 00 72 18 00 00
00 FA 08 3C 73 74 72 69 6E 67 3E DA 08 3C 6D 6F	00 FA 08 3C 73 74 72 69 6E 67 3E DA 08 3C 6D 6F
64 75 6C 65 3E 01 00 00 00 73 06 00 00 00 08 01	64 75 6C 65 3E 01 00 00 00 73 06 00 00 00 08 01
12 01 08 01	12 01 08 01

第一个是魔术后面必须是 0D0A

第二个是 16 字节后面一定是 E3

第三个是 73 后面是 B4，这个 B4 是到 73 后面第一个 29 的偏移量

第四个是 7A 后面的 08，7A 后面跟着 8 个字节的字符串

第五个是 co_name 的开始部分，也就是第一个 29 结束的地方

助教课上说只动了 6 个字节，所以改完 3，4，5 后不知道哪里错了。

想来想去才看到魔术 0D0A 和 E3 没改，改了就 OK 了 TT

flag1

```
(kali㉿kali)-[~/SHARE/rev/task1]
└─$ python broken.pyc
Trinity{pyc_fix}
```

TASK2

0	LOAD_NAME	0: bytes	
2	LOAD_NAME	1: input	
4	LOAD_CONST	0: 'Please input key:'	
6	CALL_FUNCTION	1	
8	LOAD_CONST	1: 'ascii'	
10	CALL_FUNCTION	2	
12	STORE_NAME	2: :-0	
14	LOAD_NAME	3: len	
16	LOAD_NAME	2: :-0	
18	CALL_FUNCTION	1	
20	LOAD_CONST	2: 5	
22	COMPARE_OP	2 (==)	ASCII
24	POP_JUMP_IF_FALSE	106	
26	LOAD_NAME	2: :-0	
28	LOAD_CONST	3: 0	
30	BINARY_SUBSCR		
32	LOAD_CONST	4: 100	→ D
34	COMPARE_OP	2 (==)	
36	POP_JUMP_IF_FALSE	106	
38	LOAD_NAME	2: :-0	
40	LOAD_CONST	5: 1	
42	BINARY_SUBSCR		
44	LOAD_CONST	6: 105	→ I
46	COMPARE_OP	2 (==)	
48	POP_JUMP_IF_FALSE	106	
50	LOAD_NAME	2: :-0	
52	LOAD_CONST	7: 2	
54	BINARY_SUBSCR		
56	LOAD_CONST	8: 115	→ S
58	COMPARE_OP	2 (==)	
60	POP_JUMP_IF_FALSE	106	
62	LOAD_NAME	2: :-0	
64	LOAD_CONST	9: 3	
66	BINARY_SUBSCR		
68	LOAD_CONST	10: 97	→ A
70	COMPARE_OP	2 (==)	
72	POP_JUMP_IF_FALSE	106	
74	LOAD_NAME	2: :-0	
76	LOAD_CONST	11: 4	
78	BINARY_SUBSCR		
80	LOAD_CONST	8: 115	→ S
82	COMPARE_OP	2 (==)	
84	POP_JUMP_IF_FALSE	106	
86	LOAD_NAME	4: print	
88	LOAD_CONST	12: 'Trinity{pyc_'	
90	LOAD_NAME	2: :-0	
92	LOAD_METHOD	5: decode	
94	CALL_METHOD	0	
96	BINARY_ADD		
98	LOAD_CONST	13: '}'	
100	BINARY_ADD		
102	CALL_FUNCTION	1	
104	POP_TOP		
106	LOAD_NAME	4: print	
108	LOAD_CONST	14: 'haha'	
110	CALL_FUNCTION	1	
112	POP_TOP		
114	LOAD_CONST	15: None	
116	RETURN_VALUE		
118	JUMP_ABSOLUTE	255	

查看 pyc 文件内部构造可知有多个判断
直接 ascii 转换可得 “disas”

flag2

```
(kali㉿kali)-[~/SHARE/rev/task2]  
└─$ python do_not_pycdc_me.pyc  
Please input key:disas  
Trinity{pyc_disas}  
haha
```

三、实验结果

```
(kali㉿kali)-[~/SHARE/rev/task1]
└─$ python broken.pyc
Trinity{pyc_fix}

(kali㉿kali)-[~/SHARE/rev/task2]
└─$ python do_not_pycdc_me.pyc
Please input key:disas
Trinity{pyc_disas}
haha
```

四、实验中遇到的问题及解决方案

第一题卡了几个小时，但是还是做出来了

助教课上说只动了 6 个字节，所以就没有注意魔术后面的 0D0A 和 E3。。。TT

五、实验的启示/意见和建议

附：本次实验你总共用了多长时间？包括学习相关知识时间、完成实验内容时间、完成实验报告时间。（仅做统计用，时间长短不影响本次实验的成绩。）

助教讲的很好，让我从这次实验体会到了逆向工程的冰山一角。

这节课让我学到了 CTF 的很多技巧和网络安全方面的很多知识。

本人对计算机的网络安全有很大的兴趣，将来应该也会向着这方面发展。

谢谢攻防课！