《网络攻防实战》实验报告

第<u>13</u>次实验: <u>lab13</u>

姓名: 佐藤汉

学号: <u>215220029</u>

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)

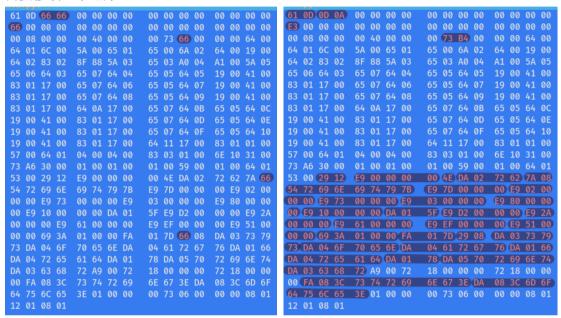
一、实验目的

得到 task1,task2 的 flags

二、实验内容

TASK1

需要修改5个地方



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

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

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!