hackme inndy reverse rc87cipher writeup

原创

charlie_heng 于 2018-02-01 11:14:10 发布 416 收藏

分类专栏： 二进制-逆向工程

本文链接：https://blog.csdn.net/charlie_heng/article/details/79225734

二进制-逆向工程 专栏收录该内容

34 篇文章 3 订阅

订阅专栏

话说这题真的挺难折腾出来的

首先这个程序是有upx壳的

但是这个壳被魔改了，关键的信息都被删除了，在google搜了半天linux upx upack 都找不到什么有用的信息

有两条路，一条是修复关键信息，一条路是像windows那些程序那样手动脱壳脱下来

两条路其实都试了一下，不过最终行得通的是第二条路

这里说下第一条路，如果你们有兴趣的话，可以去试试

首先作者把0xB4~0xB7处的UPX!填充为0xff了，然后把代码段有一个copyright的声明用nop给填掉了（话说查的时候看到upx的github上有人po了rctf的一道题，然后被upx的人吐槽copyright被改了，违反了规则hhhhh

然后去看了下github上upx的源码，源码里有一个函数 canUnpack，这里貌似是会检查文件尾的32个字节，具体的可以去看源码，然后如果想修的话，找一个正常的upx程序，把文件尾的32个字节复制上去，然后根据报错信息来慢慢改

然后来说下第二条路
首先upx壳在解压的过程中是不会加载so的，只会一直用syscall，这里用到一个神器，radare2（用得好爽

然后upx壳的规律是，最后一个syscall肯定是munmap，所以只要试一下就知道oep在哪里了

我这里写个简单的radare2脚本来找oep
对于静态链接的程序

```
9dcs
ds
```

对于动态链接的程序

```
15dsc
ds
```

这样就能找到oep，或者可以参考一下别人找oep的脚本
https://asciinema.org/a/35005

找到oep之后，也是按照上面链接里面的方式dump下来
dump下来之后用ida解析一波，有可能是我不会dump吧。。。很多函数都没识别出来，但是无所谓，大概都能猜出来

这里的oep是_start，所以最后那个call是__libc_start_main，所以第一个参数，也就是rdi，就是真正的main的地址

这里大概说下程序做了什么
这个程序只实现了加密的功能

加密的过程是，首先读取8个byte随机字符
然后根据这8个byte的字符来生成sbox

之后把这8个byte写到加密文件的开头

然后每次从被加密的文件读一个字符，就用password中的一个字符来对sbox操作一波

这里有rc87和rc87.enc，于是就可以暴力dfs破出password，这里的password其实就是flag

最后给出解题的脚本

```
f = open('./rc87', 'rb')
rc87_data = f.read()
f.close()

f = open('./rc87.enc', 'rb')
rc87enc_data = f.read()
f.close()


rc87enc_random_seed = rc87enc_data[:8]
rc87enc_data = rc87enc_data[8:]


def sbox_loop(v1, v2, sbox):
    for q in range(36):
        v2 = (13 * (~v2)) & 0xff
        v1 = (17 * (~v1)) & 0xff
        t1 = sbox[v1]
        t2 = sbox[v2]
        sbox[v1] = t2
        sbox[v2] = t1


def generate_sbox(seed):
    sbox = []
    for i in range(256):
        sbox.append(i)

    for i in range(8):
        sbox_loop(seed[i], i, sbox)
    return sbox


rc87enc_sbox = generate_sbox(rc87enc_random_seed)

password = ''
password_length = 40

def get_xor_value(tsbox):
    v9 = 0xdeadbeef
    for w in range(256):
        v11 = tsbox[w]
```

```
        v9 = 821891 * v11 ^ 23159 * v9
        v9 &= 0xffffffff
    return v9


def find_password(pas, tsbox):
    if len(pas) >= password_length:
        if(pas[39]=='}'):
           return [pas]
        else:
           return []
    possible_pas = []
    for i in range(32,127):
        ttsbox = [i for i in tsbox]
        sbox_loop(i, len(pas), ttsbox)
        v9 = get_xor_value(ttsbox)
        v15 = (17 * rc87_data[len(pas)]) ^ v9
        v15 &= 0xff
        if v15 == rc87enc_data[len(pas)]:
            possible_pas += find_password(pas + chr(i), ttsbox)
    return possible_pas


print(find_password('', rc87enc_sbox))
```