

# CUIT CTF WriteUp-鬼子进村

原创

RickGray 于 2014-05-22 16:23:57 发布 1181 收藏

分类专栏: [CTF纪实](#)

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/u013565525/article/details/26600177>

版权



[CTF纪实](#) 专栏收录该内容

10 篇文章 0 订阅

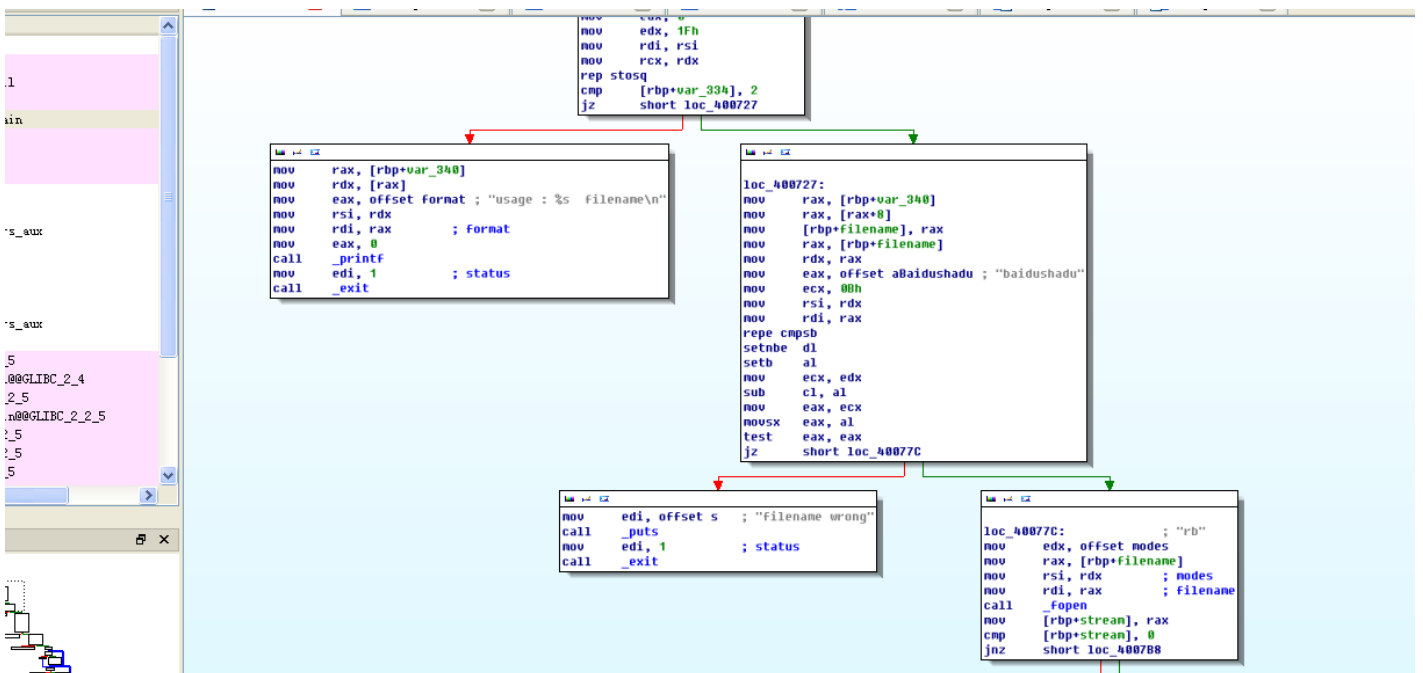
订阅专栏

该题也出的比较有意思, 64位的linux下的逆向, 对于我这种逆向菜鸟第一次利用IDA纯静态分析撸出题来, 简直是一大突破, 下面就简单分析下撸题过程。

首先运行看程序是个什么东西, 运行后发现需要输入一个文件名作为参数

```
root@Meos: ~/Desktop
root@Meos:~/Desktop# ./linux
usage : ./linux filename
root@Meos:~/Desktop# ./linux
usage : ./linux filename
root@Meos:~/Desktop# ./linux df
filename wrong
root@Meos:~/Desktop#
```

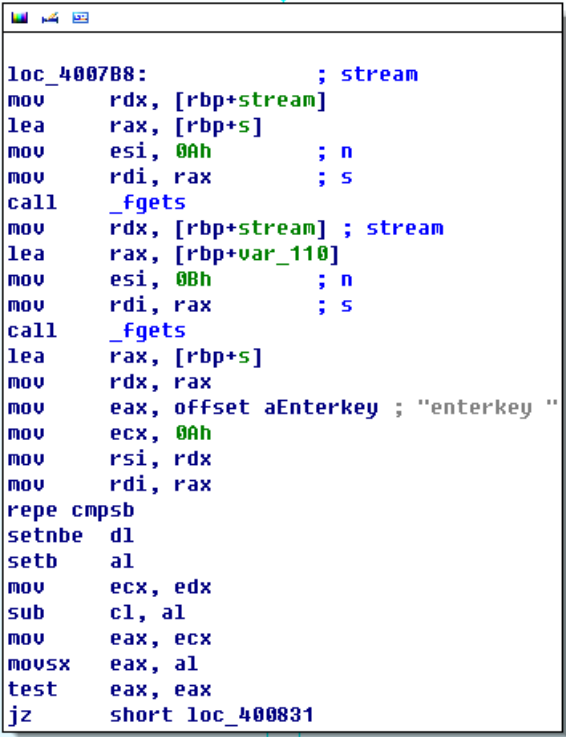
正确的文件名通过IDA定位后发现有“baidushadu” (这个名字: 好玩)



(IDA的Graph还真清晰呢) 通过IDA的Graph显示就可以很清楚地看到整个程序的流程, 后面的操作就好办多了

直接在Graph里面分析，当参数为“baidushadu”时，程序会分两次读取文件内的信息，两次读取的字节不一样

通过IDA分析可以知道，第一次读取10Bytes，第二次为12Bytes，并且通过IDA反汇编处的代码，可以判断出，程序会将第一次读取的字符串与“enterkey”进行匹配



```
loc_4007B8:                ; stream
mov     rdx, [rbp+stream]
lea    rax, [rbp+s]
mov     esi, 0Ah           ; n
mov     rdi, rax           ; s
call   _fgets
mov     rdx, [rbp+stream] ; stream
lea    rax, [rbp+var_110]
mov     esi, 0Bh           ; n
mov     rdi, rax           ; s
call   _fgets
lea    rax, [rbp+s]
mov     rdx, rax
mov     eax, offset aEnterkey ; "enterkey "
mov     ecx, 0Ah
mov     rsi, rdx
mov     rdi, rax
repe   cmpsb
setnbe dl
setb   al
mov     ecx, edx
sub     cl, al
mov     eax, ecx
movsx  eax, al
test   eax, eax
jz     short loc_400831
```

继续往下面看，发现程序会将第二次读入的12bytes进行处理然后与“pqllauzduh”进行比对，经过分析具体处理过程如下

设第二次读入的12bytes为字符串Str1，并令Str2 = “pqllauzduh”，那么处理过程如下

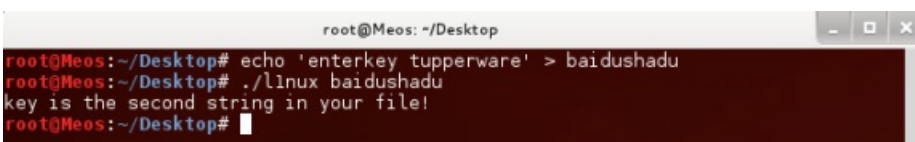
```
for i in range(0, 5):
    Str1[i] = chr(ord(Str[i]) - 4)

for i in range(5, 10):
    Str1[i] = chr(ord(Str[i]) + 3)
```

然后写个程序逆向处理可以得到Str1为：tupperware

```
s = 'pqllauzduh'
flag = ''
for i in range(0, 5):
    flag += chr(ord(s[i]) + 4)
for i in range(5, 10):
    flag += chr(ord(s[i]) - 3)
```

So, flag: *tupperware*



```
root@Meos: ~/Desktop
root@Meos:~/Desktop# echo 'enterkey tupperware' > baidushadu
root@Meos:~/Desktop# ./linux baidushadu
key is the second string in your file!
root@Meos:~/Desktop#
```