XCTF easyre-153

原创

酸酸菜鱼 于 2020-10-10 16:18:07 发布 85 收藏
分类专栏： CTF

本文链接：https://blog.csdn.net/lhk124/article/details/108999290

该题目使用pipe 和 fock 知识（摘抄他人的文字）

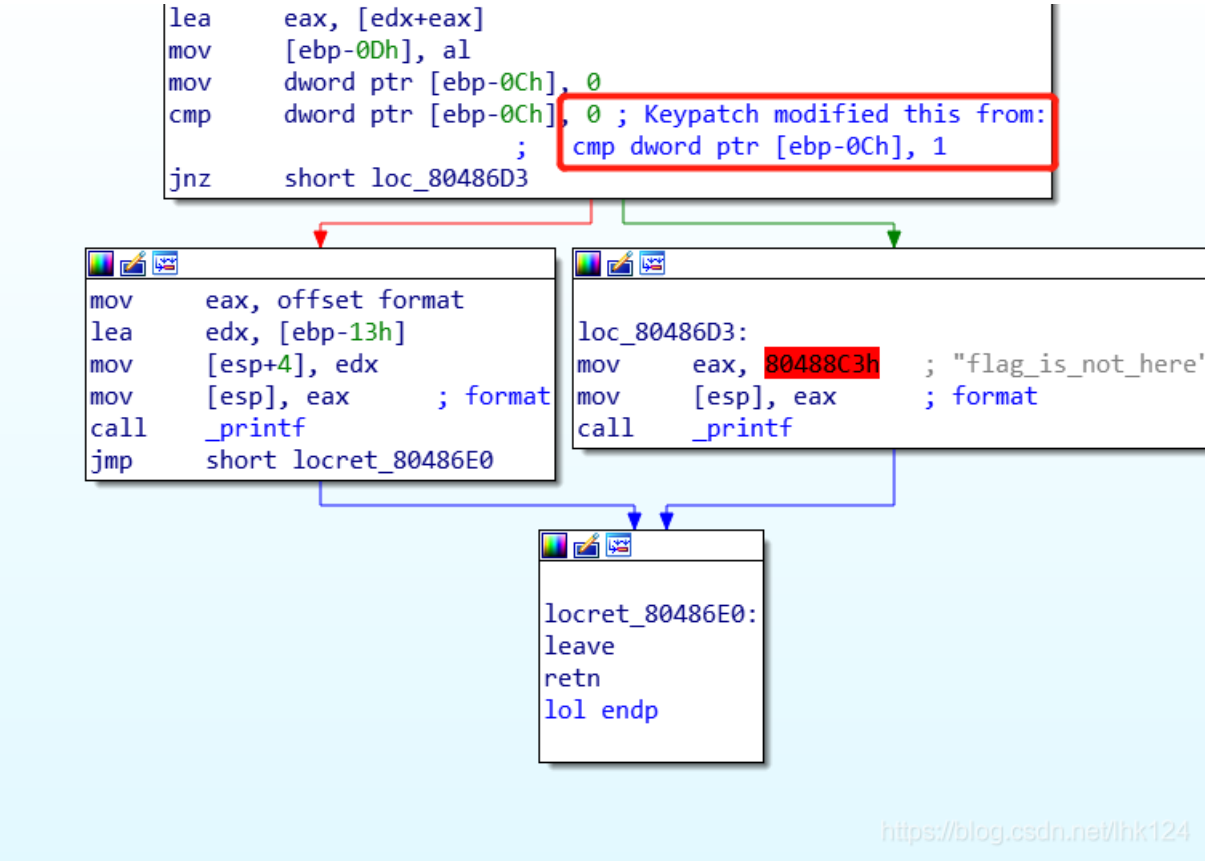
> 一个进程在由 pipe()创建管道后，一般再fork一个子进程，然后通过管道实现父子进程间的通信，管道两端可分别用描述字fd[0]以及fd[1]来描述，需要注意的是，管道的两端是固定了任务的。即一端只能用于读，由描述字fd[0]表示，称其为管道读端；另 一端则只能用于写，由描述字fd[1]来表示，称其为管道写端。如果试图从管道写端读取数据，或者向管道读端写入数据都将导致错误发生。
>
> 关于 fork 函数
>
> 1）在父进程中，fork返回新创建子进程的进程ID；
> 2）在子进程中，fork返回0；
> 3）如果出现错误，fork返回一个负值；
>
> 在子进程中写入了一个字符串，在父进程中读取

> ps -A | grep xxx(文件名)
>
> 找到程序的pid
>
> 运行程序，会输出子进程的内容；再输入pid，开始父进程的内容。
>
> 父进程主要函数是lol
>
> 看汇编代码，发现被F5误导（详情如下注意点1）
>
> patch后直接出结果

ps -A 找到fock创建的子进程。因为是经过函数exit(0)的，所以是已经退出(defunct)的那个PID。

说几个要注意的点：

1. 注意不要总是直接F5，若不直接F5,看汇编代码 会发现，直接patch一下，在上边填上对应的子进程后，运行直接出结果



2. 也可以直接写脚本解决。

```
string = "69800876143568214356928753"
v3 = 2 * ord(string[1])
v4 = ord(string[4]) + ord(string[5])
v5 = ord(string[8]) + ord(string[9])
v6 = 2 * ord(string[12])
v7 = ord(string[18]) + ord(string[17])
v8 = ord(string[10]) + ord(string[21])
v9 = ord(string[9]) + ord(string[25])
v10 = 0
s = chr(v3) + chr(v4) + chr(v5) + chr(v6) + chr(v7) + chr(v8) + chr(v9) + chr(v10)
print(s)
```

由于是给自己记录的，所以没有很详细，表示抱歉。有问题的可以直接询问，一定尽心尽力回答。