# **符号执行**

本题

<https://www.52pojie.cn/thread-1187493-1-1.html>

<https://blog.csdn.net/Hotspurs/article/details/106039643>

Angr官网

<http://angr.io/api-doc/search.html?q=posix&check_keywords=yes&area=default#>

Angr学习笔记

https://blog.csdn.net/qq\_35713009/article/details/89766154?utm\_medium=distribute.pc\_relevant\_download.none-task-blog-2~default~BlogCommendFromBaidu~default-1.nonecase&depth\_1-utm\_source=distribute.pc\_relevant\_download.none-task-blog-2~default~BlogCommendFromBaidu~default-1.nonecas

<https://www.secpulse.com/archives/83197.html>

https://blog.csdn.net/doudoudouzoule/article/details/79394447?utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control

**Ida反编译和c逆向技巧**

# **注意点1：这里应该是int而不是char，一开始因为直接从正向函数复制过来下意识地认为中间变量应该是char，其实不是，仔细看发现在正向比较的时候是和整型比较，这也是为什么key数组里存在超过255的数字的原因，使用char复制key中的元素时会丢失精度所以会出现打印出部分flag的情况**

# **即正向函数中的v4（char），反向函数中的temp（int）**

# 

# 

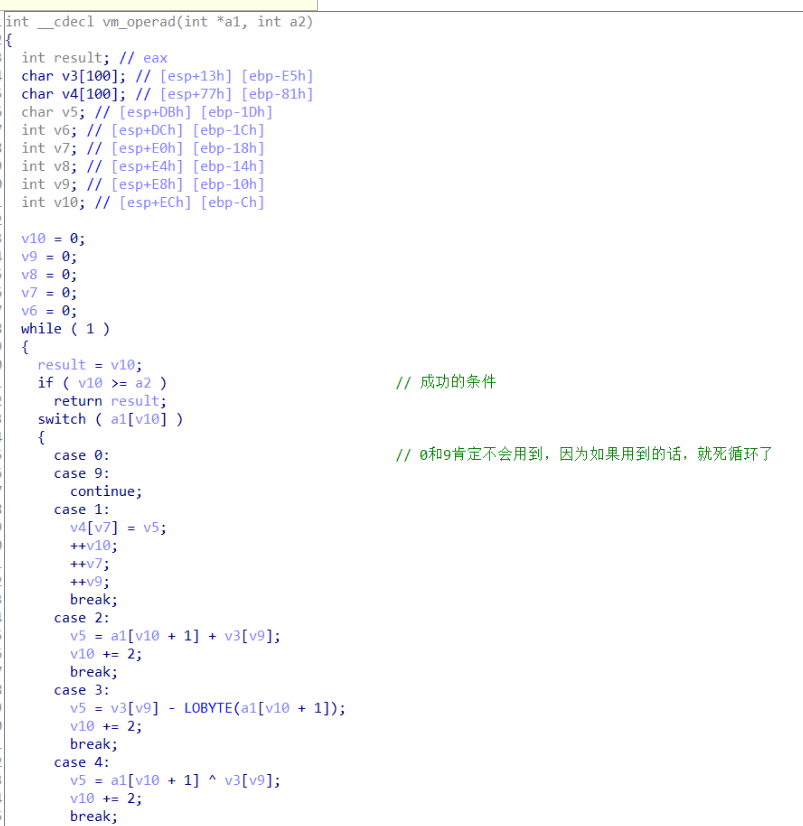
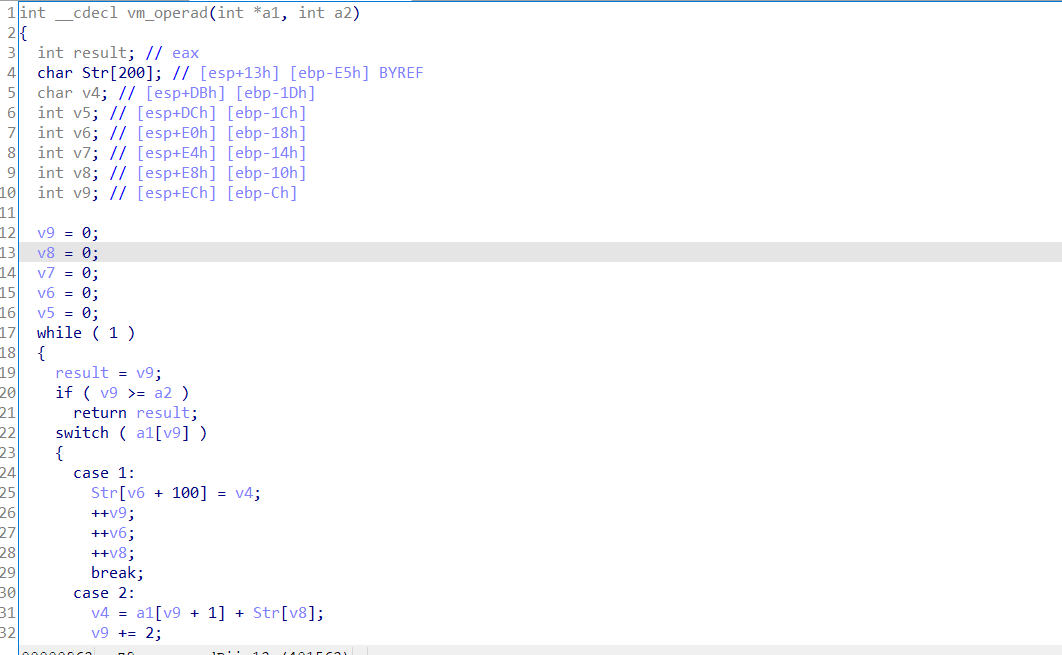
# **注意点2：原函数里用了lOBYTE一开始我根据经验认为即a1[v9+1]，但其实不是**

# **这样的，因为原数字是整形，取byte，即4字节取1字节而非LOWORD四字节去二字节，所以要用char（+1）而非（+0）**





# **注意点3：在看wp的时候非常不解，为什么这里这个原函数中的Str要换成key，而其他位置的Str要换成flag，后来经过对比发现了关键，我使用ida进行反编译的时候反编译出了Str[+100]这种东西，而在wp中的图里同位置就是Str[+0]，一开始我以为ida编译错了就直接忽视了这个100，却没仔细想过上面正向函数同样用Str[+100]得到了正确的结果，其实我的ida将wp中的两个大小为100数组合成了一个大小为200的数组，然后用Str[+100]表示第二个数组，这里是一个ida反编译需要注意的点**



# **注意点4：这点是经过我上面的第一个逆向函数到的的结论，那就是order数组是有必要的，一开始看wp我其实不明白为什么要求order数组后来经过第一个逆向函数的调试，我发现两个函数区别从在order=84的时候开始，1函数会经过case7而直接到82，2函数会根据数组到83原因是路径不唯一，即可以是83-->+1-->84和82-->+2-->84这样一来同样的结果逆向后会出现不同的初始路径，所以要求出来order**

char order[200] = { 1, 3, 4, 6, 7, 9, 10, 12, 13, 15, 16, 17, 18, 19, 20, 22, 23, 25, 26, 28, 29, 30, 31, 32, 33, 35, 36, 38, 39, 41, 42, 44, 45, 46, 47, 48, 49, 51, 52, 54, 55, 57, 58, 60, 61, 63, 64, 66, 67, 69, 70, 72, 73, 75, 76, 78, 79, 81, 82, 83, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114 };