2021巅峰极客逆向baby_maze题wp

原创

sln_1550 于 2021-08-02 01:02:26 发布 126 收藏

分类专栏： 逆向 文章标签： 安全

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/sln_1550/article/details/119280274

版权

逆向 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

IDA分析发现题目所谓的迷宫没有一个数据结构来表示，而是每一个函数就是一关：

```
1  void __noreturn sub_40180E()
2  {
3    char v0; // [rsp+Fh] [rbp-1h]
4
5    sub_55D500("This is the beginning. You can only go south.");
6    while ( 1 )
7    {
8      sub_597460(0LL, 0LL, &qword_6E4380);
9      v0 = sub_55F1F0();
10     sub_597460(0LL, 0LL, &qword_6E4340);
11     if ( v0 == 'Q' )
12       goto LABEL_5;
13     if ( v0 == 'S' )
14     {
15       sub_40187C();
16 LABEL_5:
17       sub_55BD20(0LL);
18     }
19   }
20 }
```

https://blog.csdn.net/sln_1550

输入S后进入下一关：

```
1  void sub_40187C()
2  {
3    char v0; // [rsp+Fh] [rbp-1h]
4
5    sub_55D500("Just do it");
6    while ( 1 )
7    {
8      sub_597460(0LL, 0LL, &qword_6E4380);
9      v0 = sub_55F1F0();
10     sub_597460(0LL, 0LL, &qword_6E4340);
11     switch ( v0 )
12     {
13       case 'A':
14       case 'D':
15         goto LABEL_5;
16       case 'Q':
17         sub_55BD20(0LL);
18         return;
19       case 'S':
20         sub_4066F2();
21 LABEL_5:
22         sub_55D500("Oh!!Monster");
23         break;
24       case 'W':
25         sub_40180E();
26       default:
27         continue;
28     }
29   }
30 }
```

https://blog.csdn.net/sln_1550

然后一直过关到这个函数：

```
void __noreturn sub_54DE35()
{
  sub_55D500("Good Job. \nAnd the flag is flag md5(your input)\nIf not, you may need to go faster!");
  sub_55BD20(0LL);
}
```

开始的时候我没注意看有多少个函数，以为就几十个，手工用crossref从结果逆推，搞了很久发现有问题，然后已统计，发现有7000多个函数（关），看来只能写脚本跑。
用IDA导出C代码，然后写python脚本处理，先解析每个关卡的函数，得到一个字典：

funcdicts - Dictionary (7439 elements)

Key	Type	Size	Value
sub_4A0A76	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A468E', 'LABEL_6' ...
sub_4A0B2D	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A47FC', 'LABEL_6' ...
sub_4A0BE4	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A48B3', 'LABEL_6' ...
sub_4A0C9B	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A496A', 'LABEL_6' ...
sub_4A0D52	dict	6	{'D':'IO_puts', 'A':'IO_puts', 'Q':'exit', 'S':'sub_4A4A21', 'W':'sub_ ...
sub_4A0E09	dict	7	{'A':'IO_puts', 'D':'sub_4A0EC0', 'Q':'exit', 'LABEL_5':'sub_4A4AD8', ...
sub_4A0EC0	dict	7	{'A':'sub_4A0E09', 'D':'sub_4A0F77', 'LABEL_7':'exit', 'Q':'exit', 'LA ...
sub_4A0F77	dict	7	{'A':'sub_4A0EC0', 'D':'sub_4A102E', 'LABEL_7':'exit', 'Q':'exit', 'LA ...
sub_4A1A30	dict	7	{'A':'sub_4A1979', 'D':'sub_4A1AE7', 'LABEL_7':'exit', 'Q':'exit', 'LA ...
sub_4A1AE7	dict	6	{'A':'sub_4A1A30', 'D':'IO_puts', 'LABEL_7':'exit', 'Q':'exit', 'S':'I ...
sub_4A1BA2	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A4DB4', 'LABEL_6' ...
sub_4A1C59	dict	7	{'A':'IO_puts', 'D':'sub_4A1D10', 'Q':'exit', 'LABEL_5':'sub_4A4E6B', ...
sub_4A1D10	dict	7	{'A':'sub_4A1C59', 'D':'sub_4A1DC7', 'LABEL_7':'exit', 'Q':'exit', 'LA ...
sub_4A1DC7	dict	7	{'A':'sub_4A1D10', 'D':'IO_puts', 'LABEL_7':'exit', 'Q':'exit', 'S':'I ...
sub_4A01E2	dict	7	{'A':'IO_puts', 'D':'IO_puts', 'Q':'exit', 'S':'sub_4A4A5B', 'LABEL_6' ...

https://blog.csdn.net/sln_1550

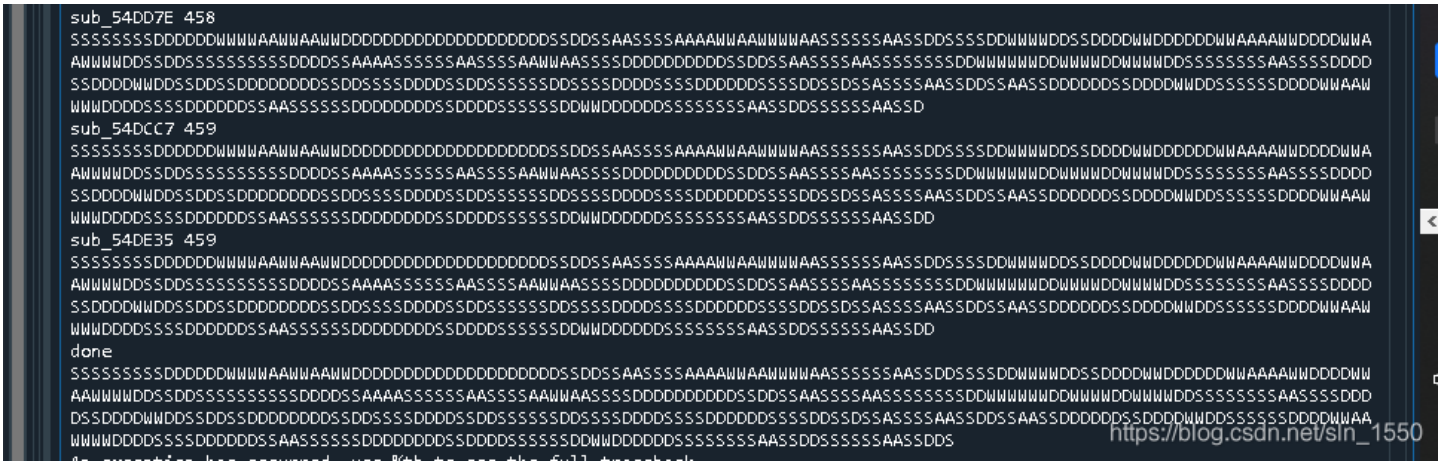
再用穷举路径来爆破，可以得到正确的输入：

```
import sys
with open(r'F:\share\20210731\baby_maze\maze.c','r') as f:
    srclines=f.readlines()

totals=len(srclines)
line=0
infunc=0
funcdicts={}
while(line<391893):
    if infunc==0 and srclines[line][0:9]=="void sub_":
        func_name=srclines[line][5:].split("(")[0].strip()
        funcdict={}
        label=''
        infunc=1
    elif srclines[line]=="}\n":
        infunc=0
        funcdicts[func_name]=funcdict
    elif infunc:
        if srclines[line][0:11]="      case ":
            key=srclines[line][12:13]
```

```
            if srclines[line+1][0:11]=="      case ":
                if srclines[line+2][0:11]==""      case ":
                    if srclines[line+3][0:5]=="LABEL":
                        label=srclines[line+3].split(":")[0]
                        labeldo=srclines[line+4].split("(")[0].strip()
                        funcdict[label]=labeldo
                        funcdict[key]=labeldo
                        funcdict[srclines[line+2][12:13]]=labeldo
                        funcdict[srclines[line+1][12:13]]=labeldo
                        line+=1
                    elif srclines[line+3][0:13]==""        goto ":
                        funcdict[srclines[line+1][12:13]]=srclines[line+3][13:].split(";")[0]
                        funcdict[srclines[line+2][12:13]]=srclines[line+3][13:].split(";")[0]
                        funcdict[key]=srclines[line+3][13:].split(";")[0]
                    else:
                        funcdict[srclines[line+1][12:13]]=srclines[line+3].split("(")[0].strip()
                        funcdict[srclines[line+2][12:13]]=srclines[line+3].split("(")[0].strip()
                        funcdict[key]=srclines[line+3].split("(")[0].strip()
                    line+=1
                elif srclines[line+2][0:5]=="LABEL":
                    label=srclines[line+2].split(":")[0]
                    labeldo=srclines[line+3].split("(")[0].strip()
                    funcdict[label]=labeldo
                    funcdict[key]=labeldo
                    funcdict[srclines[line+1][12:13]]=labeldo
                    line+=1
                elif srclines[line+2][0:13]==""        goto ":
                    funcdict[srclines[line+1][12:13]]=srclines[line+2][13:].split(";")[0]
                    funcdict[key]=srclines[line+2][13:].split(";")[0]
                else:
                    funcdict[srclines[line+1][12:13]]=srclines[line+2].split("(")[0].strip()
                    funcdict[key]=srclines[line+2].split("(")[0].strip()
                line+=1
            elif srclines[line+1][0:13]==""        goto ":
                funcdict[key]=srclines[line+1][13:20]
            elif srclines[line+1][0:5]=="LABEL":
                label=srclines[line+1].split(":")[0]
                labeldo=srclines[line+2].split("(")[0].strip()
                funcdict[label]=labeldo
                funcdict[key]=labeldo
                line+=1
            else:
                funcdict[key]=srclines[line+1].split("(")[0].strip()
            line+=1
        elif srclines[line][0:5]=="LABEL":
            label=srclines[line].split(":")[0]
            labeldo=srclines[line+1].split("(")[0].strip()
            funcdict[label]=labeldo
            line+=1
    line+=1

keys=['A','D','W','S']
for f in funcdicts:
    for k in keys:
            if funcdicts[f][k].count('LABEL'):
                l=funcdicts[f][k]
                funcdicts[f][k]=funcdicts[f][1]

start='sub_40187C'
```

```
end='sub_54DE35'
def goforward(start,flag,path):
    if start=='sub_54DE35':
        print("done",'S'+flag)
        sys.exit(1)
    if start not in path:
        path=list(path)
        path.append(start)
        for k in keys:
            if start in funcdicts:
                if k in funcdicts[start]:
                    if funcdicts[start][k][0:3]=='sub':
                        #print(funcdicts[start][k],len(path),flag)
                        goforward(funcdicts[start][k],flag+k,path)

goforward(start,'',[])
```

不到1分钟可以跑出结果：



输入程序验证成功：

这题我开始手撸浪费了很多时间，不过感觉也没这么简单，为啥有61个队伍都做出来了，我个人觉得这题其实比medical_app要难一些，那题才38个队伍做出来。
可能这题有简便方法，麻烦有知道的大佬赐教下。