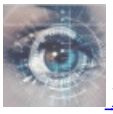
XCTF reverse maze

原创

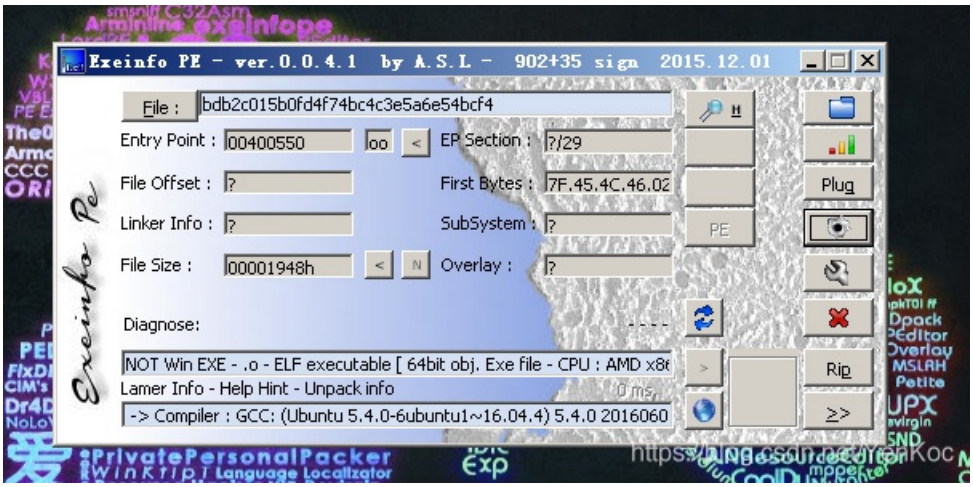
YenKoc 于 2020-02-05 15:43:11 发布 264 收藏 1

分类专栏： XCTF

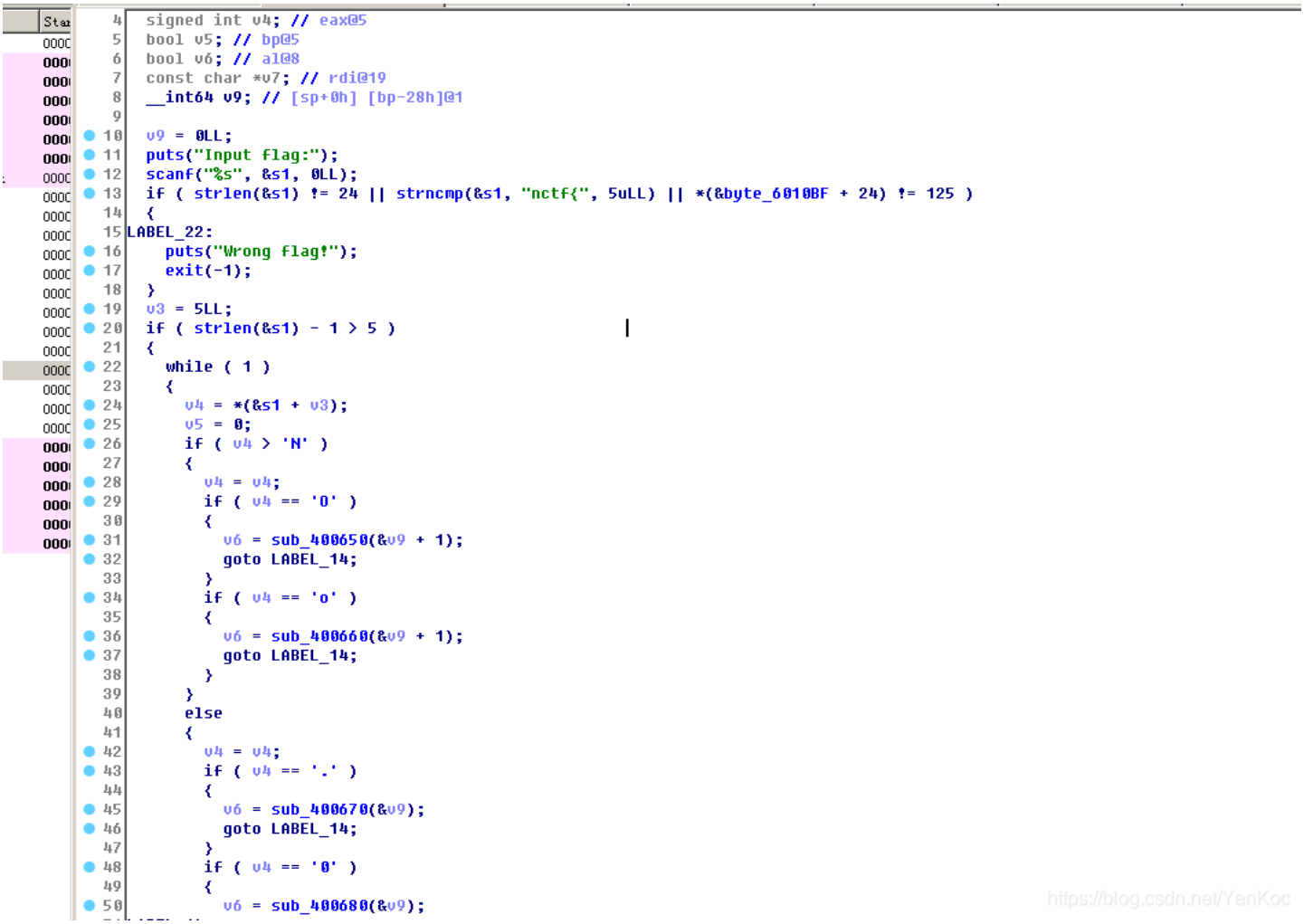
本文链接：https://blog.csdn.net/YenKoc/article/details/104183346



一.查壳



二.拖入ida64，静态调试，找到主函数F5反编译



```
  signed int v4; // eax@5
  bool v5; // bp@5
  bool v6; // al@8
  const char *v7; // rdi@19
  __int64 v9; // [sp+0h] [bp-28h]@1

  v9 = 0LL;
  puts("Input flag:");
  scanf("%s", &s1, 0LL);
  if ( strlen(&s1) != 24 || strncmp(&s1, "nctf{", 5uLL) || *(&byte_6010BF + 24) != 125 )
  {
LABEL_22:
    puts("Wrong flag!");
    exit(-1);
  }
  v3 = 5LL;
  if ( strlen(&s1) - 1 > 5 )
  {
    while ( 1 )
    {
      v4 = *(&s1 + v3);
      v5 = 0;
      if ( v4 > 'N' )
      {
        v4 = v4;
        if ( v4 == 'O' )
        {
          v6 = sub_400650(&v9 + 1);
          goto LABEL_14;
        }
        if ( v4 == 'o' )
        {
          v6 = sub_400660(&v9 + 1);
          goto LABEL_14;
        }
      }
      else
      {
        v4 = v4;
        if ( v4 == '.' )
        {
          v6 = sub_400670(&v9);
          goto LABEL_14;
        }
        if ( v4 == '0' )
        {
          v6 = sub_400680(&v9);
```


二.1 思路分析（逆向是真的费时间，每个函数都要分析过去）：

1.发现每个if最终都会进入LABEL-15

```
000
      }
LABEL_15:
      if ( !sub_400690(asc_601060, SHIDWORD(v9), v9) )
        goto LABEL_22;
      if ( ++v3 >= strlen(&s1) - 1 )
      {
        if ( v5 )
          break;
```

点进去，看看这个函数是干啥的。

```
__int64 __fastcall sub_400690(__int64 a1, int a2, int a3)
{
  __int64 result; // rax@1

  result = *(a1 + a2 + 8LL * a3);
  LOBYTE(result) = result == ' ' || result == '#';
  return result;
}
```

这里基本可以推理出a3是行，a2是列了。
那么我们的迷宫一定是一个8列的矩阵。
得知这点，退出去，再分析。

```
EL_15:
    if ( !sub_400690(asc_601060, SHIDWORD(v9), v9) )
```

v9是行，SHIDWORD函数是指的下一个字节，得知这点，从上面分析可知：
就将四个函数点进去就可以判断方向了。
举个例子：

```
bool __fastcall sub_400650(_DWORD *a1)
{
  int v1; // eax@1

  v1 = (*a1)--;
  return v1 > 0;
}
```

相对应，其他方向，O:下,o:上，0:左，.:右。
2.将迷宫打印出来，python脚本如下。

```
maze="  *******   *  **** * ****  * ***  *#  *** *** ***     *********"
res=""
count=0
for str in maze:
    res+=str
    count+=1
    if count%8==0:
        print(res)
        res=""
```

3.由于我们是从左上角开始的，开始我们的路线
个人感觉答案不唯一的。。。。没办法。》-《。。。
根据迷宫最后得到的**flag: nctf{o0oo00O000oooo…OO}**