2022DASCTF X SU Re WriteUp

原创

Whitebird_ 已于 2022-04-01 20:35:09 修改 350 收藏 1

分类专栏： DASCTF 文章标签： 安全

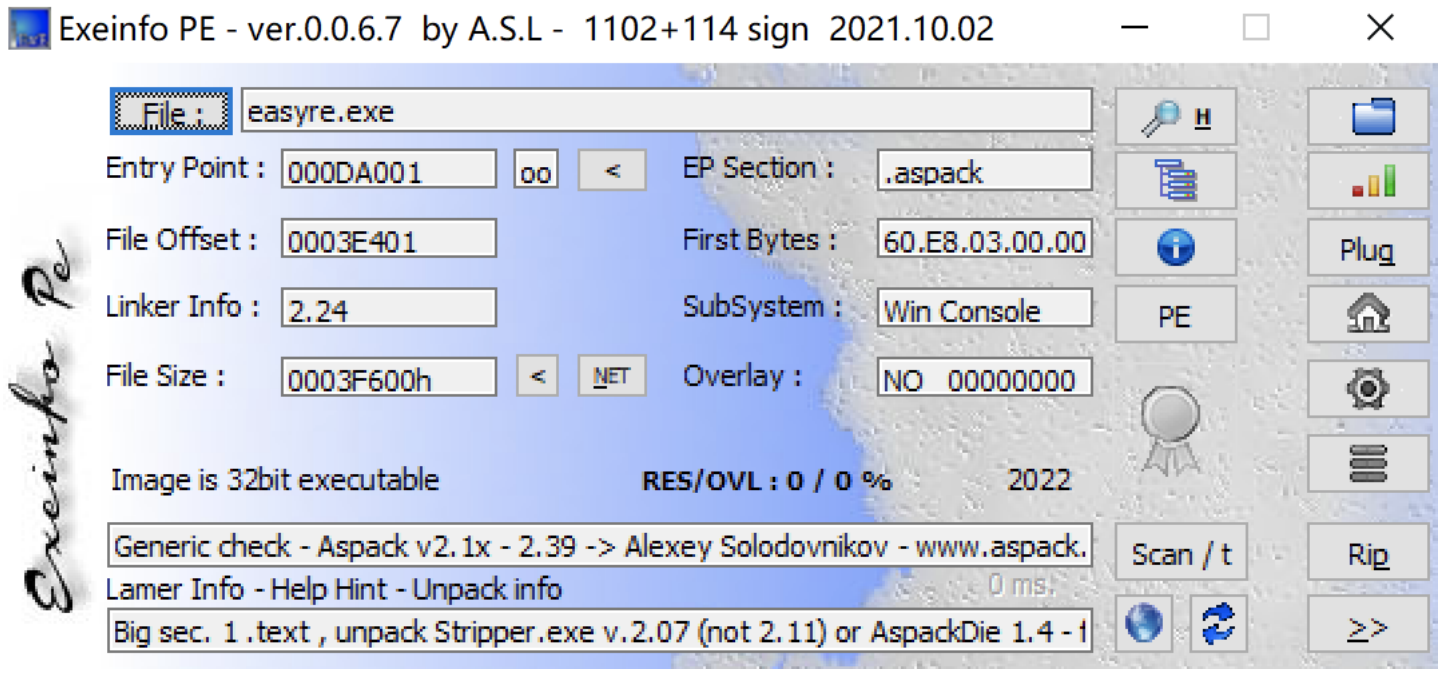
于 2022-04-01 20:29:55 首次发布

本文链接：https://blog.csdn.net/jxnu_666/article/details/123906920

2022DASCTF X SU Re WriteUp

easyre

一道签到题,老样子先查壳



asp壳比较容易脱，可以百度了解一下，我直接通过ESP定律脱的,然后把内存dump下来用IDA分析

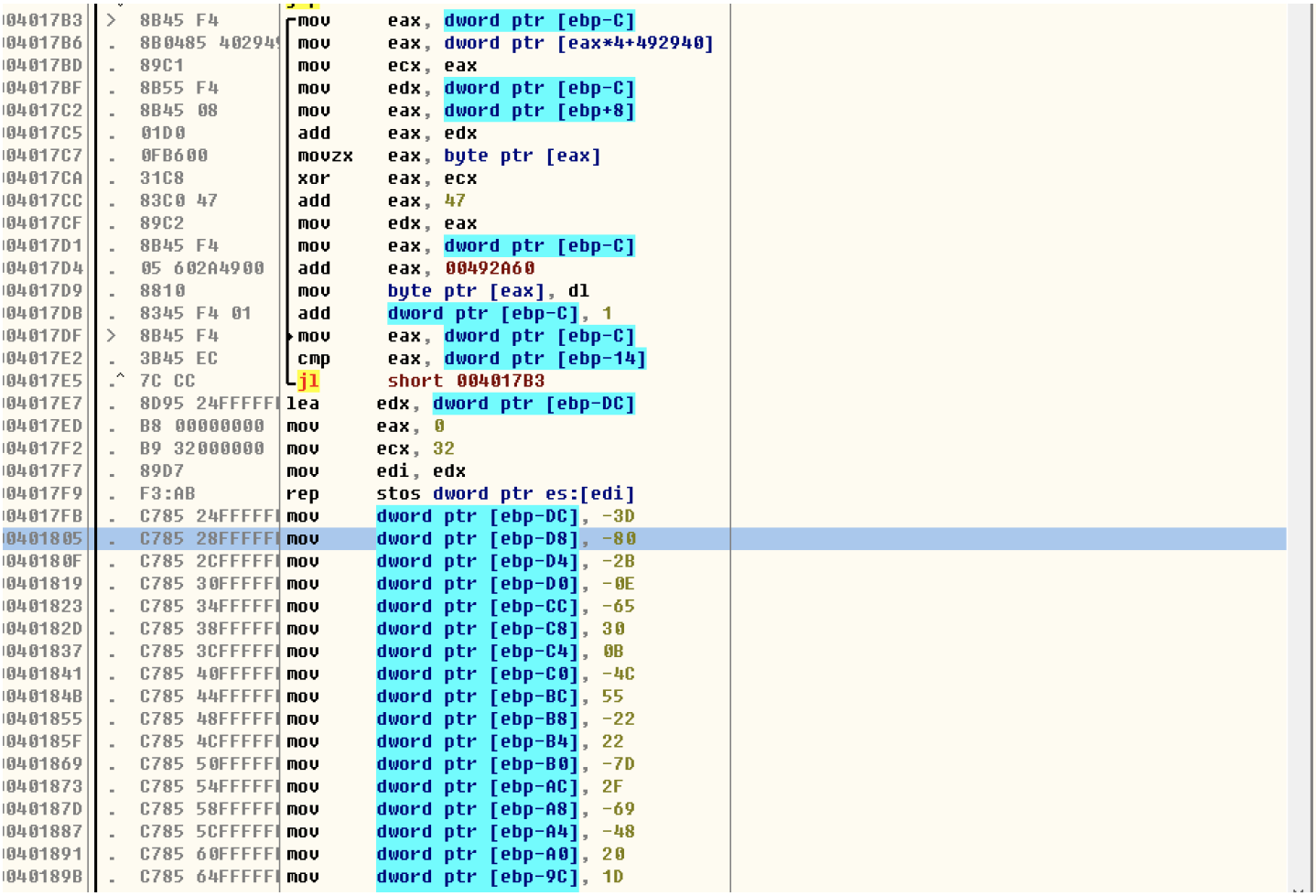
```
int __cdecl sub_401771(char *Str)
{
  int v2[50]; // [esp+1Ch] [ebp-DCh] BYREF
  int v3; // [esp+E4h] [ebp-14h]
  int j; // [esp+E8h] [ebp-10h]
  int i; // [esp+ECh] [ebp-Ch]

  v3 = strlen(Str);
  sub_401500();
  sub_40152B();
  sub_401593();
  sub_401610(Str, v3);
  for ( i = 0; i < v3; ++i )
    byte_492A60[i] = (LOBYTE(dword_492940[i]) ^ Str[i]) + 71;
  memset(v2, 0, sizeof(v2));
  v2[0] = -61;
  v2[1] = -128;
  v2[2] = -43;
  v2[3] = -14;
  v2[4] = -101;
  v2[5] = 48;
  v2[6] = 11;
  v2[7] = -76;
  v2[8] = 85;
  v2[9] = -34;
  v2[10] = 34;
  v2[11] = -125;
  v2[12] = 47;
  v2[13] = -105;
  v2[14] = -72;
  v2[15] = 32;
  v2[16] = 29;
  v2[17] = 116;
  v2[18] = -47;
  v2[19] = 1;
  v2[20] = 115;
  v2[21] = 26;
  v2[22] = -78;
  v2[23] = -56;
  v2[24] = -59;
  v2[25] = 116;
```

通过字符串定位来到加密函数，下面的v2数组是密文，str数组是我们输入的flag，分析了一下sub_401500、sub_40152B、sub_401593，是一个变种的RC4.

如果是正常RC4，我们可以用脚本或者网站解，变种虽然也可以硬逆，但是比较麻烦。我这里是直接动调出密钥流，RC4虽然内部有改变，但是每次XOR的密钥流是一样的.

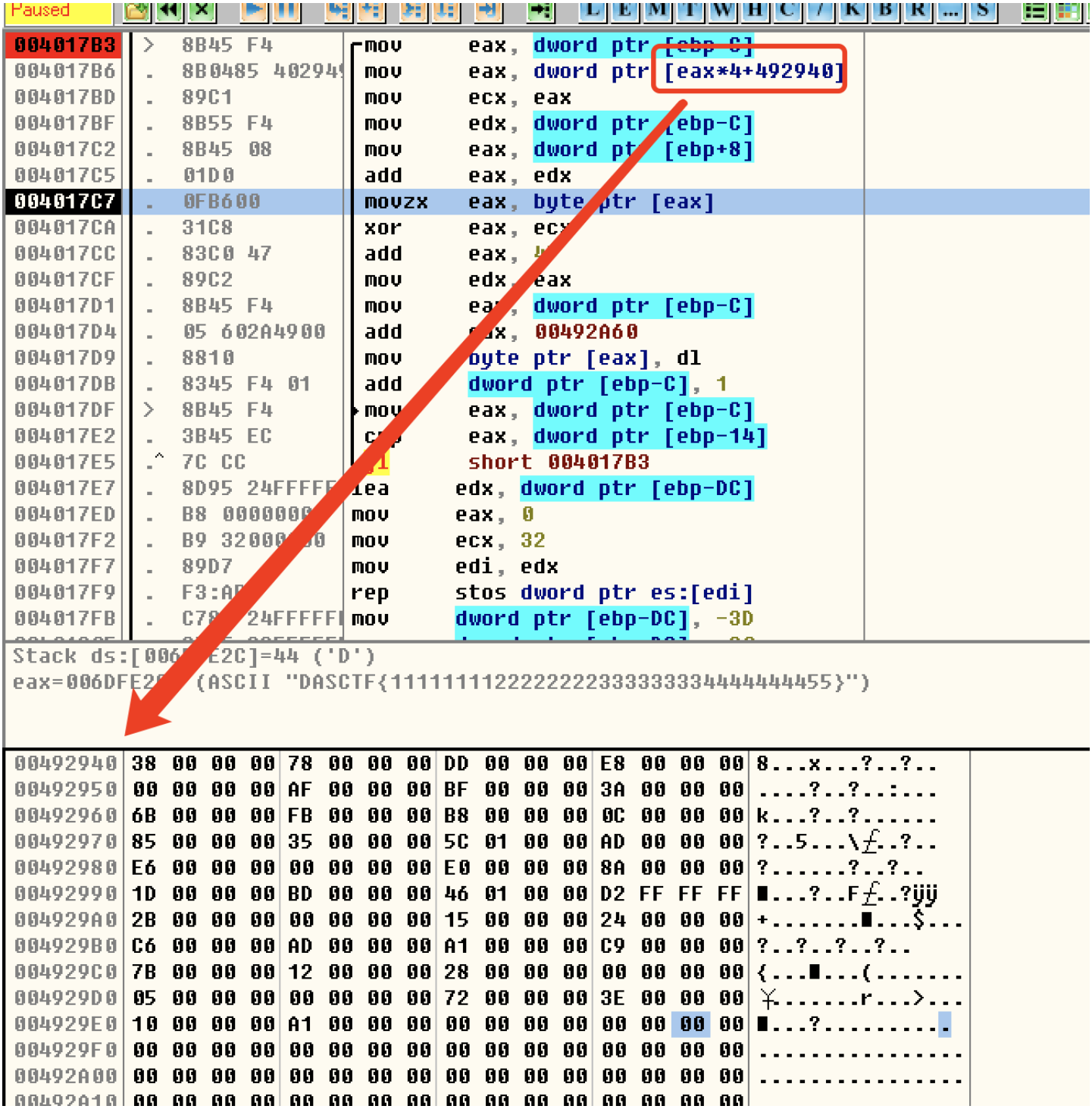
我们用OD进行动态调试，记住要先脱壳。



这一块就是对应我们IDA分析的代码区域，上面的循环是xor，下面是密文

在加密的地方下个断点，然后开始调试，前提得先知道flag的格式

格式为:DASCTF{11111111222222223333333344444444455}



找到密钥流，现在拿密文XOR密钥流就是flag，贴一下脚本

```
xor=[0x38,0x78,0xDD,0xE8,0X00,0XAF,0XBF,0X3A,0X6B,0XFB,0XB8,0XC,0X85,0X35,0X15C,0XAD,0XE6,0X00,0XE0,0X8A,0X1D,0X
BD,0X146,0XFFFFFFD2,0X2B,0X00,0X15,0X24,0XC6,0XAD,0XA1,0XC9,0X7B,0X12,0X28,0X00,0X05,0X00,0X72,0X3E,0X10,0XA1]

decode=[0xFFFFFFC3,0xFFFFFF80,0xFFFFFFD5,0xFFFFFFF2,0xFFFFFF9B,0x30,0x0b,0xFFFFFFB4,0x55,0xFFFFFFDE,0x22,0xFFFFF
F83,0x2f,0xFFFFFF97,0xFFFFFFB8,0x20,0x1d,0x74,0xFFFFFFD1,0x1,0x73,26,0xFFFFFFB2,0xFFFFFFC8,0xFFFFFFC5,0x74,0xFFF
FFFC0,0x5B,0xFFFFFFF7,0xF,0xFFFFFFD3,0x1,0x55,0xFFFFFFB2,0xFFFFFFA4,0xFFFFFFAE,0x7B,0xFFFFFFAC,0x5C,0x56,0xFFFFF
FBC,0x23]
flag=""
for i in range(len(xor)):
    flag+=chr(((decode[i]-0x47)^xor[i])&0x7f)
print(flag)
```

StarGate

这题比赛的时候没写出来，后面复现了下，还是挺有意思的。

首先这题与常规的re不同，没有给文件，给了个靶机，那就先nc一下

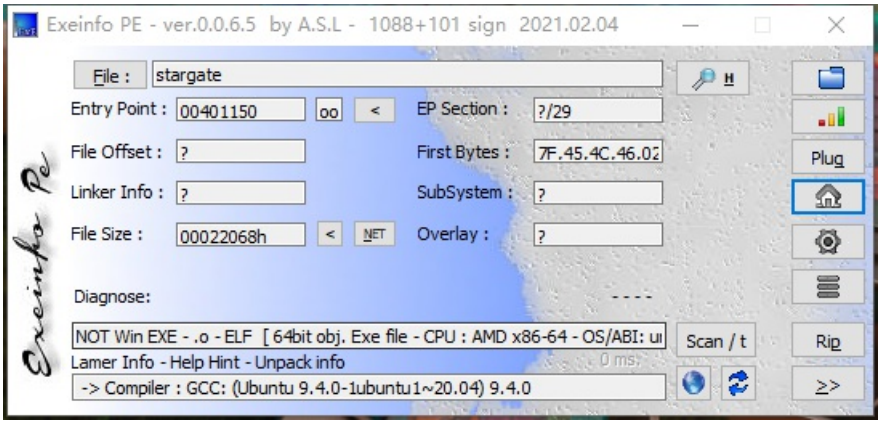
```
AAAAAAAAAOYFQAAAAAAA5gUAAAAAAAAeAAAAAAAAAAYAAAAAAAAAAgAAAAAAAAACAAAAAAAAAG4AAAD+//9vAgAAAAAAAAIBkA
AAAAAAAgGAAAAAAAAQAAAAAAAAAAHAAAAAQAAAAgAAAAAAAAAAAAAAAAAAAAB9AAAABAAAAAIAAAAAAAAASAZAAAAAAABIBgAAAA
AAAHgAAAAAAAAABgAAAAAAAAAIAAAAAAAAABgAAAAAAAAhwAAAAQAAABCAAAAAAAAAMAGQAAAAAAwAYAAAAAADYAAAAAAAAA
AYAAAAYAAAACAAAAAAAAAAYAAAAAAAAAJEAAAABAAAABgAAAAAAAAAEEAAAAAAAAAQAAAAAAAGwAAAAAAAAAAAAAAAAAAAQA
AAAAAAAAAAAAAAAAAACMAAAAAQAAAAYAAAAAAAAAIBBAAAAAAAAgEAAAAAAAAKAAAAAAAAAAAAAAAAAAAAQAAAAAAAABAAAAA
AAAAAlwAAAAEAAAAGAAAAAAAAAMAQQAAAAAAAwBAAAAAAAACQAAAAAAAAAAAAAAAAAAAAEAAAAAAAAAQAAAAAAAAKAAAAABAA
AABgAAAAAAABQEUAAAAAAAFARAAAAAAAAVXYBAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAACmAAAAQAAAAYAAAAAAAAq
IdBAAAAAACohwEAAAAAA0AAAAAAAAAAAAAAAAAAAAEAAAAAAAAAAAAAAAAAAAArAAAAEAAAACAAAAAAAAACQQQAAAAAAAJAB
AAAAADqWAAAAAAAAAAAAAAAAAAAACAAAAAAAAAAAAAAAAAAAALQAAAABAAAAgAAAAAAADs6EEAAAAAAOzoAQAAAAAAzAUAAAA
AAAAAAAAAAAAAAAQAAAAAAAAAAAAAAAAAAADCAAAAQAAAAIAAAAAAAAAuO5BAAAAAAC47gEAAAAAABgXAAAAAAAAAAAAAAAAA
AIAAAAAAAAAAAAAAAAAAAAzAAAAA4AAADAAAAAAAAABAeQgAAAAAAEA4CAAAAAAAIAAAAAAAAAAAAAAAAAAAACAAAAAAAAAIA
AAAAAAAANgAAAAPAAAAwAAAAAAAAAYHkIAAAAAABgOAgAAAAAACAAAAAAAAAAAAAAAAAAAAgAAAAAAAACAAAAAAAADkAAAA
BgAAAAMAAAAAAAAIB5CAAAAAAgDgIAAAAAANABAAAAAAABwAAAAAAAAIAAAAAAAABAAAAAAAAA7QAAAAEAAAADAAAAAAA
AAPAfQgAAAAAA8A8CAAAAAAQAAAAAAAAAAAAAAAAAACAAAAAAAAAIAAAAAAAAPIAAAABAAAAwAAAAAAAAIEIAAAAAAA
AQAgAAAAAYAAAAAAAAAAAAAAAAAAAgAAAAAAAACAAAAAAAAD7AAAAQAAAAMAAAAAAAAYCBCAAAAAABgEAIAAAAANgHA
AAAAAAAAAAAAAAAAAIAAAAAAAAAAAAAAAAAAAAQEAAAgAAAADAAAAAAAAEAoQgAAAAAOBgCAAAAAAwAAAAAAAAAAAAA
AAAAIAAAAAAAAAAAAAAAAAAAAYBAAABAAAAMAAAAAAAAAAAAAAAAAAAADgYAgAAAAAKQAAAAAAAAAAAAAAAAAAEAAAAAAA
AAQAAAAAAAAABAAAAwAAAAAAAAAAAAAAAAAAAABhGAIAAAAAA8BAAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAA
==end==
Legend has it that there are hidden treasures in the universe.
Explorer, can you find the treasure through the stargate?
Passing through each stargate requires the correct code, can you find the correct way through the s
tar gate?
The first stargate is right in front of you.
Password : Wrong password!
You hava be stuck in first star forever!!
Bye Bye!
```

上面好像是个base64编码的文件，先用网站的base64解密下看看

我这里直接用py写了个脚本交互下载文件，后面也会用到

```
from pwn import*
import base64
import sys
import subprocess
p=remote('node4.buuoj.cn',26237)
p.recvuntil("Gate")#将Wellcome to Star Gate接收
code=p.recvuntil("==end==")#接收中间的base64
with open("stargate",'wb') as f:
    f.write(base64.b64decode(code[:-8]))#code包含==end==，所以只编码到倒数第八个字符
p.interactive()
```

用IDA分析一下这个文件，ELF、无壳、64位的程序




```
void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
{
  char psw[40]; // [rsp+0h] [rbp-30h] BYREF
  unsigned __int64 v4; // [rsp+28h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  sub_401254(a1, a2, a3);
  puts("Legend has it that there are hidden treasures in the universe. ");
  puts("Explorer, can you find the treasure through the stargate? ");
  puts("Passing through each stargate requires the correct code, can you find the correct way through the star gate? ");
  puts("The first stargate is right in front of you. ");
  printf("Password : ");
  __isoc99_scanf("%20s", psw);
  if ( !strcmp("plTIOPTYFfi", psw) )
    sub_4012C3();
  if ( !strcmp("JvteSmWoH", psw) )
    sub_4013AE();
  if ( !strcmp("KltuRVDhNEpA", psw) )
    sub_401599();
  if ( !strcmp("WgDrztg", psw) )
    sub_401704();
  if ( !strcmp("DccwEiB", psw) )
    sub_4018EF();
  if ( !strcmp("afOgUcwpT", psw) )
    sub_401ADA();
  if ( !strcmp("GToWSAHtMlF", psw) )
    sub_401C45();
  if ( !strcmp("IhCBxd", psw) )
    sub_401E30();
  if ( !strcmp("tGOcyntXeBkva", psw) )
    sub_401F9B();
  if ( !strcmp("nYnGOtyrRfJO", psw) )
    sub_402186();
  if ( !strcmp("pVIMHiCDMyvcUwc", psw) )
    sub_4024F1();
  if ( !strcmp("oJoHGUE", psw) )
    sub_40265C();
  if ( !strcmp("dnoSkEMyB", psw) )
    sub_4027C7();
  if ( !strcmp("cbRNHqBlveM", psw) )
    sub_402932();
  if ( !strcmp("irtYDTaqQqhqT", psw) )
    sub_402A9D();
  if ( !strcmp("dGnOkWvBijXBX", psw) )
```

会发现让我们输入一个password，然后进入一个函数

```
IDA View-A | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums

1  __int64 sub_4012C3()
2  {
3    char s2[40]; // [rsp+0h] [rbp-30h] BYREF
4    unsigned __int64 v2; // [rsp+28h] [rbp-8h]
5
6    v2 = __readfsqword(0x28u);
7    printf("Now you in universe plTIOPTYFfi, please enter the password to enter the next universe through the stargate. \n: ");
8    __isoc99_scanf("%20s", s2);
9    if ( !strcmp("xhgrNMZizKMWwM", s2) && dword_4221C8 == 1 )
10   {
11     dword_4221C8 = 0;
12     sub_40A60A();
13     exit(0);
14   }
15   if ( !strcmp("acJOFKGoRXppqN", s2) && dword_42211C == 1 )
16   {
17     dword_42211C = 0;
18     sub_413591();
19     exit(0);
20   }
21   puts("Wrong password!\nYou're stuck in this universe forever!!");
22   return 0LL;
23 }
```

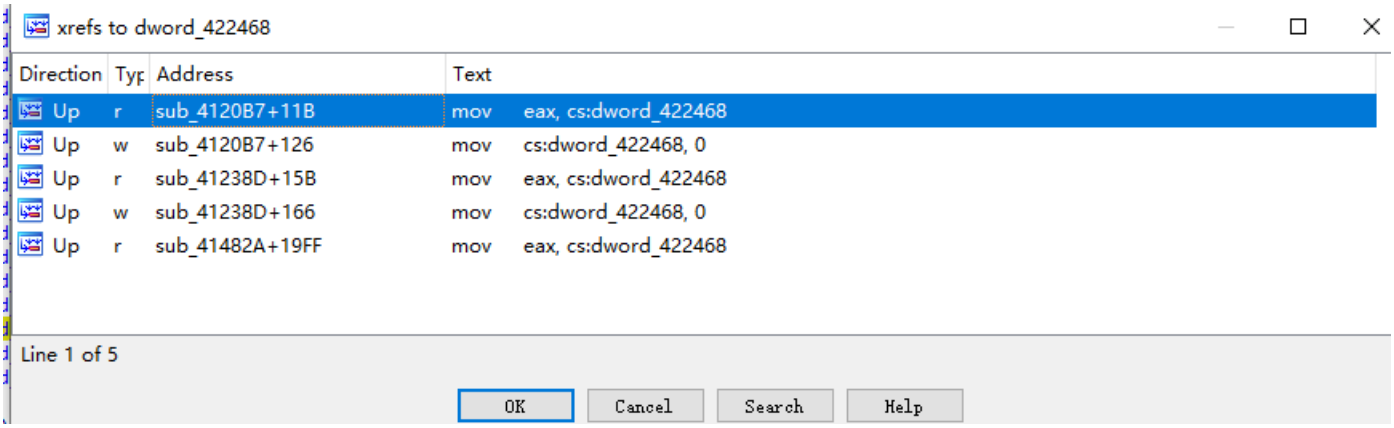
进去之后，会让你继续输入一个password，再调用其他函数，有点套娃，并且把dword_4221C8这种全局变量由1变成了0

然后我在字符串中看到了cat flag，这个命令是用来获取远程主机上的flag，莫非是个pwn？？

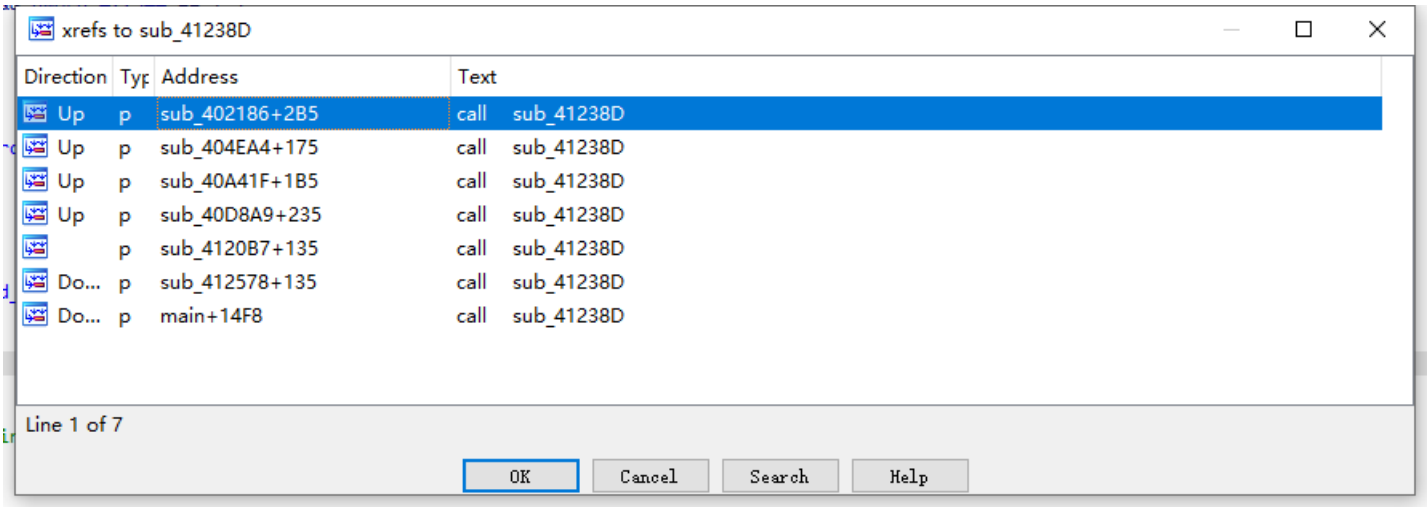
对字符串进行交叉引用，找到了调用该命令的函数

```
457    && !dword_42231C
458    && !dword_4226A8
459    && !dword_4221B8
460    && !dword_42216C
461    && !dword_422774
462    && !dword_422380
463    && !dword_422608
464    && !dword_4227D4
465    && !dword_4226F4
466    && !dword_422440
467    && !dword_4220A0
468    && !dword_422704
469    && !dword_422498
470    && !dword_422754
471    && !dword_4220D8
472    && !dword_4222DC
473    && !dword_422290
474    && !dword_42276C
475    && !dword_422178
476    && !dword_422444
477    && !dword_422138
478    && !dword_422690
479    && !dword_422180
480    && !dword_4226AC
481    && !dword_4227C8
482    && !dword_422524
483    && !dword_422468
484    && !dword_42271C
485    && !dword_422314 )
486  {
487    catflag();
488  }
489  puts("There seems to be treasure in this universe but you're not qualified to get it.");
```

上面是个if(!xxxxx)的条件判断，大概有四五百个dword_xxxxx这种变量并且初始值都是1，并且我们需要将所有的1变成0，这样就可以获得flag了



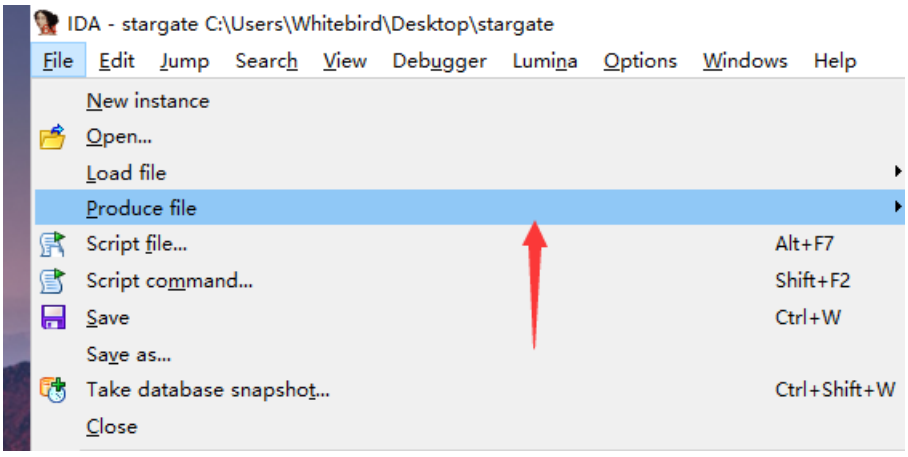
发现一个变量对应了多个函数，而一个函数又有可能会出现在多个函数里，并且也对应了多个变量



需要注意的是如果这个变量已经是0的话, 你是不能走这条路的.

通过简单分析后，就知道这个类似于一个图。变量类似于是图中的边，而函数就是图中的点

这样子题目的意思是让我们以某种顺序来输入password，并且没有重复，最后将所有的变量变为0，出题人也提示了欧拉路径，其实我感觉像一个迷宫



先生成一个asm文件，然后用py脚本把所有的password提取出来

```
import os
points = []
now, rt = -1, 0
flag = False
edges = []
with open("stargate.asm") as f :
    tmp = f.readlines()
    for i in range(len(tmp)) :
        if 'lea     rdi, aLegendHasItTha' in tmp[i] :#起点
            break
        if 'lea     rdi, aThere' in tmp[i] :#终点
            flag = True
        if 'call    _strcmp' in tmp[i] :#获取_strcmp上一行的password，也就是进入下一个星球的password
            s = tmp[i-1].find('"')
            p = tmp[i-1][s+1:-2]
            if p not in points :#如果之前没走过，就新加一个
                points.append(p)
            if now == -1 :
                continue
            print(points[now], p, i)
            edges.append([now, points.index(p)])
        elif 'lea     rdi, aNowYouIn' in tmp[i] :
            s = tmp[i].find('se ')#获取"Now you in universe xxxxx中的password
            t = tmp[i][s:].find(', ')
            p = tmp[i][s+3:s+t]
            if p not in points :
                points.append(p)
            now = points.index(p)#now为p在点中的索引
            if flag :
                rt = now#最后终点的index
                flag = False#后面还要继续遍历
print(points)
ff = open("tt.in", "w")
ff.write(str(len(points)) + " " + str(len(edges)) + " " + str(rt) + '\n')
for i in edges :
    ff.write(str(i[0]) + " " + str(i[1]) + "\n")#以两个索引之间作为边，每个password就是索引，也是点
ff.close()
os.system(".\dfs.exe")#利用dfs跑欧拉
s = input().split(' ')#将跑完的路径给作为输入带回，用来以索引生成真正的password
with open('tt.ans', "w") as f :
    for i in range(len(s)) :
        if s[i] == s[i-1] :#结束
            continue
        try :
            f.write(points[int(s[i])-1]+" ")#将index替换成password
        except :
            print(i)
print(len(edges)//2, len(s))
```

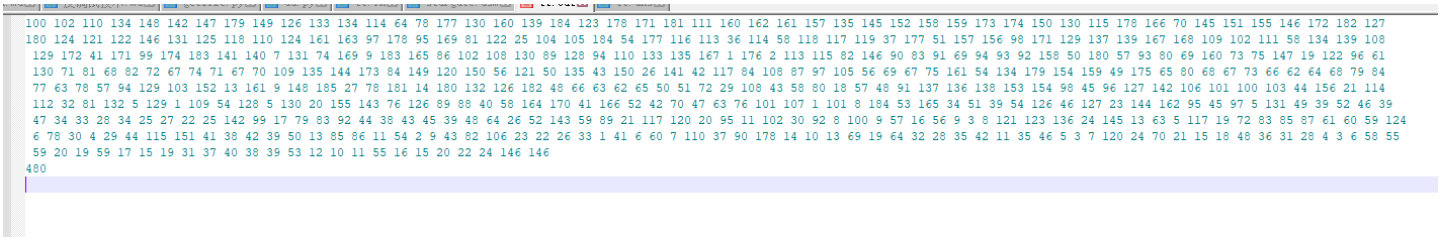
我们利用password来当作点，两点的索引组成一条边，剩下的就是找个脚本跑这个欧拉图就行了

```
185 958 145
2 3
2 4
2 5
2 6
2 7
2 8
9 10
9 11
9 12
9 13
14 15
14 16
14 17
14 18
14 19
14 20
21 22
21 23
21 24
21 25
21 26
21 19
3 2
3 27
3 28
3 29
27 3
27 30
27 31
27 32
27 33
27 34
30 27
30 35
30 18
30 36
37 38
37 39
37 40
37 41
37 42
37 43
38 37
38 44
38 45
38 46
```

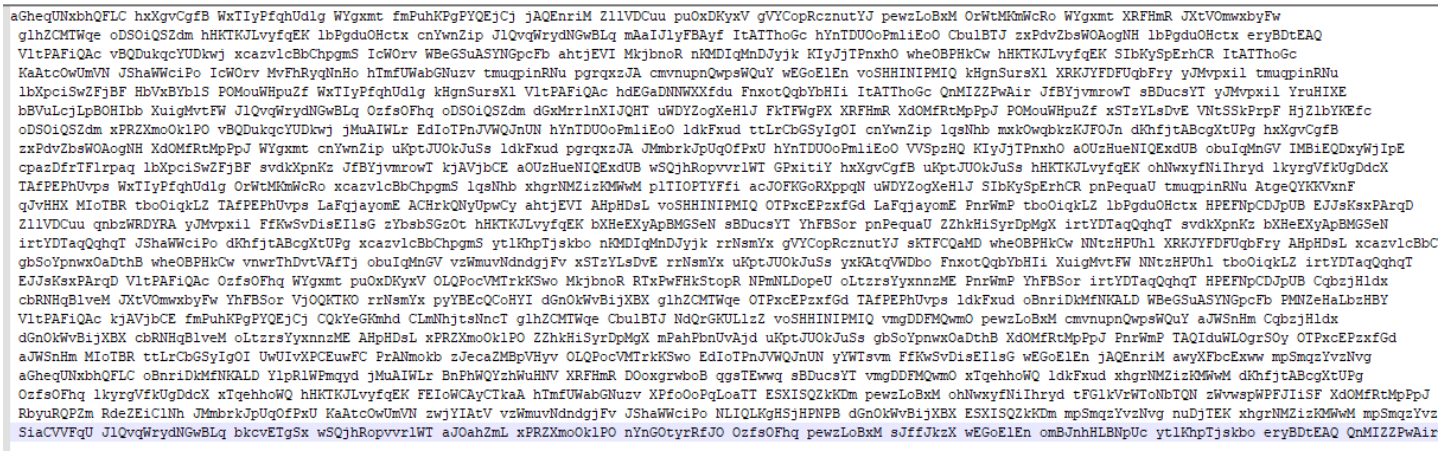
这个是将password变成了索引，一对为边

```cpp
#include <cstdio>
#include <iostream>
#include <vector>
using namespace std;
vector<int> v[1000];
int x,y,n,m,du[5020],sta=0x3f3f3f3f,mp[2520][2520],ans[5200],top;
inline int read()
{
    int x=0,f=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
    while(isdigit(c)){x=(x<<1)+(x<<3)+(c^48);c=getchar();}
    return x*f;
}
int cnt1 = 0;
void dfs(int x)
{
    for(int i=1;i<=n;i++)
    {
        if(mp[x][i])
        {
```

```
            mp[x][i]--;
            mp[i][x]--;
            cnt1 ++;
            dfs(i);
        }
    }
    ans[++top]=x;
}
void dfs1(int p, int t) {
    v[t].push_back(p);
    for(int i = 1; i <= n; i ++ ) {
        if(mp[p][i]) {
            mp[p][i]--;
            cnt1 ++;
            mp[i][p]--;
            dfs1(i, t);
        }
    }
}
int main()
{
    freopen("tt.in", "r", stdin);
    freopen("tt.out", "w", stdout);
    int rt;
    scanf("%d%d%d", &n, &m, &rt);
    rt ++;
    for(int i=1;i<=m;i++)
    {
        scanf("%d%d", &x, &y);
        x ++, y ++;
        mp[x][y] = 1;
        mp[y][x] = 1;
        du[x]++;
        du[y]++;
    }
    dfs(rt);
    while(true) {
        for(int i = 1; i <= n; i ++ ) {
            dfs1(i, i);
        }
        bool flag = 0;
        for(int i = 1; i <= n; i ++ ) for(int j = 1; j <= n; j ++ ) flag |= mp[i][j];
        if(!flag) break;
    }
    for(int i = 1; i <= top; i ++ )
        printf("%d ", ans[i]);
    printf("%d", rt);
    printf("\n%d\n", cnt1);
    return 0;
}
```



这是跑完欧拉后的结果，依次输入对应索引的password就是真正的路径，但是不知道为什么dfs后的值都+1了，懂得师傅可以评论告诉我一下谢谢



替换成password，这么多肯定不是我们手动输入，必须还得使用交互式脚本来

```
print("when ready, press enter to continue.")
ans = input().split(' ')[:-1]
print(ans)
r.recvuntil('Password : ')
for i in range(len(ans)) :
    print(i, ans[i])
    r.sendline(ans[i])
    try :
        print(r.recvuntil(':'))
    except :
        break
r.interactive()
```

最后可以把这个与前面的脚本结合一下：

```python
from pwn import*
import base64
import sys
import subprocess
p=remote('node4.buuoj.cn',26237)
p.recvuntil("Gate")#将Wellcome to Star Gate接收
code=p.recvuntil("==end==")#接收中间的base64
with open("stargate",'wb') as f:
    f.write(base64.b64decode(code[:-8]))#code包含==end==，所以只编码到倒数第八个字符
print("when ready, press enter to continue.")
ans = input().split(' ')[:-1]
print(ans)
p.recvuntil('Password : ')
for i in range(len(ans)) :
    print(i, ans[i])
    p.sendline(ans[i])
    try :
        print(p.recvuntil(':'))
    except :
        break
p.interactive()
```

总体流程：

第一个py生成ELF->IDA生成ASM->第二个py生成边调用dfs, dfs结果输入回第二个py拿到点对应的password, 把password给第一个py和服务器交互, 最后拿到flag

```
482 JEsWYUQAgac
b' Now you in universe JEsWYUQAgac, please enter the password to enter the next universe through the stargate. \n:'
483 tsvUIxBTZR
b' Now you in universe tsvUIxBTZR, please enter the password to enter the next universe through the stargate. \n:'
484 kbavlM
b' Now you in universe kbavlM, please enter the password to enter the next universe through the stargate. \n:'
485 aOfclGDrs
b' Now you in universe aOfclGDrs, please enter the password to enter the next universe through the stargate. \n:'
486 wvYCbBKyDFsW
b' Now you in universe wvYCbBKyDFsW, please enter the password to enter the next universe through the stargate. \n:'
487 JEsWYUQAgac
b' Now you in universe JEsWYUQAgac, please enter the password to enter the next universe through the stargate. \n:'
488 MXHFBHdceaixHo
b' Now you in universe MXHFBHdceaixHo, please enter the password to enter the next universe through the stargate. \n:'
489 bfSxFrsROEyDIil
b' Now you in universe bfSxFrsROEyDIil, please enter the password to enter the next universe through the stargate. \n:'
490 OJlmcjQ
[*] Switching to interactive mode
 flag{e5143a52-3a0d-4d67-ac82-10f3b674fdd2}
Bye Bye!
[*] Got EOF while reading in interactive
$
[*] Interrupted
[*] Closed connection to node4.buuoj.cn port 26237

┌──(root💀kali)-[~/桌面/re]
└─#
```

这题主要还是学习交互式脚本的写法，平时对于这种脚本的使用比较少，导致做题时没什么思路