

2021年春秋杯网络安全联赛秋季赛逆向snake.exe Writeup

原创

bin_cat 于 2021-11-28 20:44:17 发布 9894 收藏 1

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/m0_58348028/article/details/121589636

版权



下载附件snake.exe，看题目描述是一个游戏那就打开玩玩，发现这是一个贪吃蛇游戏，游戏大概是通过玩贪吃蛇打到一定的分数之后就会有一个flag校验的窗口，所以基本可以判断出不出flag和那个贪吃蛇游戏基本没关系，可以跳过游戏直接去找flag校验的过程

下面动手，先查壳



一个32位的程序，有一个UPX壳，把壳脱掉扔进ida pro找到main函数

```
6 while ( 1 )
{
    7     system("cls");
    8     byte_406040 = 49;
    9     byte_406048 = 0;
   10     dword_406044 = 0;
   11     dbl_406050 = 0;
   12     dbl_406058 = 0;
   13     dbl_40605E = 0.01;
   14     dword_406044 = 1;
   15     byte_40606C = 119;
   16     sub_401DF();
   17     sub_401D2();
   18     if ( sub_4024C() )
   19         break;
   20     if ( byte_406048 != 121 && byte_406048 != 89 )
   21     {
   22         puts((byte_405032));
   23         system("pause");
   24         return 0;
   25     }
   26 }
   27 return 0;
}
```

猜测这里是一个对贪吃蛇游戏的描述，我们可以跳过这个去找字符串窗口看看能不能找到什么

```

Line 1 of 53
0x00404060: 7G5d5bAy+7MdLWl5CdMTlcJmwkNUgb2AQl3CcmPpVf6DAp72scOS1b
0x00404061: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=0
0x00404062: 不知道下次再来吧-
0x00404063: 呀! 你真厉害!
0x00404064: color 0B
0x00404065: pause
0x00404066: /*-----
0x00404067: WELCOME TO THE GAME OF RETRO SNAKE
0x00404068: -----
0x00404069: hello
0x0040406A: .welcome to play
0x0040406B: %0.2f
0x0040406C: Unknown error
0x0040406D: _mother(%s in %s(%g, %g)) (retval=%g)\n
0x0040406E: Argument domain error (DOMAIN)
0x0040406F: Argument singularity (SIGN)
0x00404070: Overflow range error (OVERFLOW)
0x00404071: The result is too small to be represented (UNDERFLOW)
0x00404072: Total loss of significance (TLOSS)
0x00404073: Partial loss of significance (PLLOSS)
0x00404074: Mingw-w64 runtime failure\n
0x00404075: Address Rip has no image-section
0x00404076: VirtualQuery failed for 81d bytes at address %p

```

字符串窗口找到1号可疑字符，2号字符猜测是一个base64表后期有没有用目前未知，先跟1号进去

```

j_00404060: 7G5d5bAy+7MdLWl5CdMTlcJmwkNUgb2AQl3CcmPpVf6DAp72scOS1b
j_00404061: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=0
j_00404062: 不知道下次再来吧-
j_00404063: 呀! 你真厉害!
j_00404064: color 0B
j_00404065: pause
j_00404066: /*-----
j_00404067: WELCOME TO THE GAME OF RETRO SNAKE
j_00404068: -----
j_00404069: hello
j_0040406A: .welcome to play
j_0040406B: %0.2f
j_0040406C: Unknown error
j_0040406D: _mother(%s in %s(%g, %g)) (retval=%g)\n
j_0040406E: Argument domain error (DOMAIN)
j_0040406F: Argument singularity (SIGN)
j_00404070: Overflow range error (OVERFLOW)
j_00404071: The result is too small to be represented (UNDERFLOW)
j_00404072: Total loss of significance (TLOSS)
j_00404073: Partial loss of significance (PLLOSS)
j_00404074: Mingw-w64 runtime failure\n
j_00404075: Address Rip has no image-section
j_00404076: VirtualQuery failed for 81d bytes at address %p

```

找到引用它的函数继续跟函数

```

0x00404060: 7G5d5bAy+7MdLWl5CdMTlcJmwkNUgb2AQl3CcmPpVf6DAp72scOS1b
0x00404061: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=0
0x00404062: 不知道下次再来吧-
0x00404063: 呀! 你真厉害!
0x00404064: color 0B
0x00404065: pause
0x00404066: /*-----
0x00404067: WELCOME TO THE GAME OF RETRO SNAKE
0x00404068: -----
0x00404069: hello
0x0040406A: .welcome to play
0x0040406B: %0.2f
0x0040406C: Unknown error
0x0040406D: _mother(%s in %s(%g, %g)) (retval=%g)\n
0x0040406E: Argument domain error (DOMAIN)
0x0040406F: Argument singularity (SIGN)
0x00404070: Overflow range error (OVERFLOW)
0x00404071: The result is too small to be represented (UNDERFLOW)
0x00404072: Total loss of significance (TLOSS)
0x00404073: Partial loss of significance (PLLOSS)
0x00404074: Mingw-w64 runtime failure\n
0x00404075: Address Rip has no image-section
0x00404076: VirtualQuery failed for 81d bytes at address %p

```

这逻辑就出来了先是输入v1在经过两次处理之后和我们这个可疑字符串进行了对比，显而易见v1即为我们要找的flag，所以我们只要逆推就可以了找到两次处理

这是第一次

```

0x00404060: 7G5d5bAy+7MdLWl5CdMTlcJmwkNUgb2AQl3CcmPpVf6DAp72scOS1b
0x00404061: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=0
0x00404062: 不知道下次再来吧-
0x00404063: 呀! 你真厉害!
0x00404064: color 0B
0x00404065: pause
0x00404066: /*-----
0x00404067: WELCOME TO THE GAME OF RETRO SNAKE
0x00404068: -----
0x00404069: hello
0x0040406A: .welcome to play
0x0040406B: %0.2f
0x0040406C: Unknown error
0x0040406D: _mother(%s in %s(%g, %g)) (retval=%g)\n
0x0040406E: Argument domain error (DOMAIN)
0x0040406F: Argument singularity (SIGN)
0x00404070: Overflow range error (OVERFLOW)
0x00404071: The result is too small to be represented (UNDERFLOW)
0x00404072: Total loss of significance (TLOSS)
0x00404073: Partial loss of significance (PLLOSS)
0x00404074: Mingw-w64 runtime failure\n
0x00404075: Address Rip has no image-section
0x00404076: VirtualQuery failed for 81d bytes at address %p

```

这是第二次

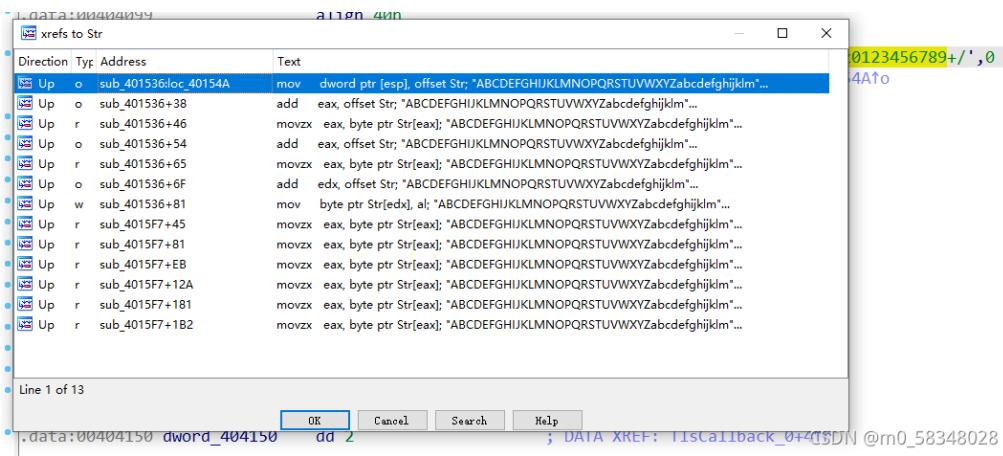
```

1: v9 = (16 * *(BYTE*)(v14 + a1)) & 0x30;
2: if ( v14 + 1 >= a3 )
3: {
4:     *(BYTE*)(a2 + v12) = Str[v9];
5:     *(BYTE*)(v12 + 1 + a2) = 61;
6:     v4 = v12 + 2;
7:     v11 = v12 + 3;
8:     *(BYTE*)(v4 + a2) = 61;
9:     break;
10: }
11: v5 = v12;
12: v13 = v12 + 1;
13: *(BYTE*)(a2 + v5) = Str[((int)*(unsigned __int8*)(v14 + 1 + a1) >> 4) & 0xF | v9];
14: v10 = (4 * *(BYTE*)(v14 + 1 + a1)) & 0x3C;
15: if ( v14 + 2 >= a3 )
16: {
17:     *(BYTE*)(a2 + v13) = Str[v10];
18:     v6 = v13 + 1;
19:     v11 = v13 + 2;
20:     *(BYTE*)(v6 + a2) = 61;
21:     break;
22: }
23: *(BYTE*)(a2 + v13) = Str[((int)*(unsigned __int8*)(v14 + 2 + a1) >> 6) & 3 | v10];

```

00000A56 sub_4015F7:22 (401E56) CSDN @rn0_58348028

观察第二次处理方式不难发现这是一个base64加密对应上了我们一开始在字符串窗口看到的那个base64表，返回字符串窗口看看base64表有没有被改



发现base64表有一个写操作，跟进去就发现表被改了

```

1 int sub_401536()
2{
3    int result; // eax
4    char v1; // [esp+13h] [ebp-15h]
5    signed int v2; // [esp+14h] [ebp-14h]
6    int j; // [esp+18h] [ebp-10h]
7    int i; // [esp+1Ch] [ebp-Ch]
8
9    result = dword_406060;
10   if ( !dword_406060 )
11   {
12       v2 = strlen(Str);
13       for ( i = 0; v2 / 2 > i; ++i )
14       {
15           for ( j = 0; v2 - i - 1 > j; ++j )
16           {
17               if ( Str[j] > Str[j + 1] )
18               {
19                   v1 = Str[j];
20                   Str[j] = Str[j + 1];
21                   Str[j + 1] = v1;
22               }
23           }
24       }
25   }
26 }
```

00000A5C sub_401536:4 (40153C) CSDN @rn0_58348028

外面的都不重要，红色圈才是真正对base64原表下手，按照他的逻辑看看他把表变成什么样了

```

a='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+'
a=list(a)
ss=len(a)
i=0

while(ss / 2 > i):
    j=0
    while(ss - i - 1>j):
        if (a[j] > a[j + 1]):
            v1 = a[j]
            a[j] = a[j + 1]
            a[j + 1] = v1
        j+=1
    i+=1
aa=''
for xxxx in a:
    aa+=xxxx

print(aa)

```

发现表变成了这样ABCDEF~~GHIJKLMNOPQRST0123456789+/UVWXYZ~~abcdefghijklmnopqrstuvwxyz

后面只需要直接写代码逆就ok了

```

import base64
xx = "7G5d5bAy+TMdLWlu5CdkMTlcJnwkJNugb2AQL3CcmPpVf6DAp72sc0S1b"
string1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/UVWXYZ"
string2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
print (base64.b64decode(xx.translate(str.maketrans(string1,string2))))
xxx=list(base64.b64decode(xx.translate(str.maketrans(string1,string2))))
for a in range(1, 11):
    for j in range(42):
        if len(xxx) % a:
            xxx[j] ^= (a + j)
        else:
            xxx[j] ^= ((j % a) + j)
        j += 1

for i in xxx:
    print(chr(i), end="")

```

出flag

flag{5e2200bc-f21a-5421-a90b-57dec19fe196}