Writeup of CTFxNuist (逆向部分

原创

MozhuCY 于 2018-02-02 14:17:22 发布 503 收藏
分类专栏： CTF入门

两个月前，偶然发现了南信大的CTF平台，大师傅们对我们小萌新还是比较友好的.....这个平台上的题目有难有易，逆向部分非常推荐做一做的✧(ﾟ∀ﾟ)و✧，

0x00：签到题

一个1kb的文件，经检查可知为pyc文件，哇。。。直接反编译啊ლ(￣ー￣ლ)。可见输入的flag每位经过特殊运算后和数组key进行比较，由于没有多解的情况，爆破好了（依旧懒得想逆运算hhhhhhhh，

```
import time ##############反编译结果################
key = [10413,11676,9438,10608,15115,12772,12771,9042, 2597, 2716, 2509, 3150, 2713, 2606, 2607, 2306, 2403,
flag = raw_input('Please input the flag:')
index = 0
for i in flag:
    time.sleep(0.1)
    if not ord(i) * ord(i) ^ ord(i) % (31 + index) == key[index]:
        print 'Wrong answer!'
        quit()
        continue
    index += 1
if index == len(key):
    print 'You got it!'
else:
    print 'Wrong answer!'
```

```
key =     [10413,11676,9438,10608,15115,12772,12771,9042,2597,2716,2509,3150,2713,2606,2607,2306,2403,15636
flag = ''
for j in range(len(key)):
    for i in range(33,127):
        if i * i ^ i % (31 + j) == key[j]:
            flag+=chr(i)
print flag
```

0x03:f***k xiaoyang

IDA打开发现居然有壳.....随后脱掉UPX壳后，反编译，可以看到伪代码中有好多f**kxiaoyang3! 类似的字符串，然后是两个输入分别是ID和password，然后将字符串f**kxiaoyang3!和数字3传入加密函数，最后和password进行对比，既然是这样可以用OD下断点或者直接抄一

遍加密函数找到生成的key，这里选择了第二种方法（OD不会用啊(ノ ゜□゜)ノ ┻━┻

```
#脚本

l='fuckxiaoyang3!'
flag=''
for i in range(len(l)):
    if ord(l[i])<65 or ord(l[i])>90 :
        if ord(l[i])<97 or ord(l[i])>122 :
            flag+=l[i]
        else:
            flag+=chr((ord(l[i])+3-97)%26+97)
    else:
        flag+=chr((ord(l[i])+3-65)%26+65)
print(flag)
```

0x05 x64ELF
查看主函数，看到了ptrace函数.....硬怼好了...定位到关键函数处，woc看到了flag，原本以为这样就结束了，提交了一下，GG，还是怪怪的分析函数吧，原来是一个异或运算，打开pycharm开始写脚本，一开始写脚本被坑了一下...忘记了do里面还有一个++*(&v22 + v26)的操作，下面是关键函数和脚本

```
const char *__fastcall sub_4006D6(const char *a1)
{
  int v2; // [sp+10h] [bp-80h]@3
  int v3; // [sp+14h] [bp-7Ch]@3
  int v4; // [sp+18h] [bp-78h]@3
  int v5; // [sp+1Ch] [bp-74h]@3
  int v6; // [sp+20h] [bp-70h]@3
  int v7; // [sp+24h] [bp-6Ch]@3
  int v8; // [sp+28h] [bp-68h]@3
  int v9; // [sp+2Ch] [bp-64h]@3
  int v10; // [sp+30h] [bp-60h]@3
  int v11; // [sp+34h] [bp-5Ch]@3
  int v12; // [sp+38h] [bp-58h]@3
  int v13; // [sp+3Ch] [bp-54h]@3
  int v14; // [sp+40h] [bp-50h]@3
  int v15; // [sp+44h] [bp-4Ch]@3
  int v16; // [sp+48h] [bp-48h]@3
  int v17; // [sp+4Ch] [bp-44h]@3
  int v18; // [sp+50h] [bp-40h]@3
  int v19; // [sp+54h] [bp-3Ch]@3
  int v20; // [sp+58h] [bp-38h]@3
  int v21; // [sp+5Ch] [bp-34h]@3
  __int64 v22; // [sp+60h] [bp-30h]@1
  __int64 v23; // [sp+68h] [bp-28h]@1
  int v24; // [sp+70h] [bp-20h]@1
  char v25; // [sp+74h] [bp-1Ch]@1
  int v26; // [sp+84h] [bp-Ch]@3
  const char *v27; // [sp+88h] [bp-8h]@1
```

```
  v22 = 0x454B4548544D4149LL;
  v23 = 0x6568746D61692159LL;
  v24 = 0x2179656B;
  v25 = 0;
  v27 = malloc(0x15uLL);
  if ( !strcmp(a1, "hehe") )
  {
    v27 = "flag{iamnotthekey}";
  }
  else
  {
    v26 = -1;
    v2 = 44;
    v3 = 46;
    v4 = 47;
    v5 = 50;
    v6 = 50;
    v7 = 50;
    v8 = 36;
    v9 = 47;
    v10 = 41;
    v11 = 75;
    v12 = 25;
    v13 = 16;
    v14 = 11;
    v15 = 20;
    v16 = 5;
    v17 = 13;
    v18 = 9;
    v19 = 31;
    v20 = 91;
    v21 = 95;
    do
    {
      if ( ++v26 > 19 )
        break;
      ++*(&v22 + v26);
    }
    while ( (*(&v22 + v26) ^ a1[v26]) == *(&v2 + v26) );
    if ( v26 == 20 )
      v27 = "You got it right!";
    else
      v27 = "Try Again!";
  }
  return v27;
}
```

```
l='IAMTHEKEY!iamthekey!'
a=[44,46,47,50,50,50,36,47,41,75,25,16,11,20,5,13,9,31,91,95]
flag=''
for i in range(len(l)):
    flag+=chr((ord(l[i])+1)^a[i])
print flag
```

（密码学的最后一题也可以看看啊，我出的
（//▽//），不过那的确是个坑啊，顺便坑了某XYZ.