2018上海市大学生网络安全大赛 逆向 cpp

原创

snowleopard_bin 于 2018-11-06 20:22:51 发布 625 收藏
分类专栏： CTF

本文链接：https://blog.csdn.net/snowleopard_bin/article/details/83793381

一道简单的逆向，就是两个函数加密，反着来一遍就行，不用爆破。

```
for ( i = 0; ; ++i )
  {
    LODWORD(v1) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(a1);
    if ( i >= v1 )
      break;
    LODWORD(v2) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](
    v3 = v2;
    LODWORD(v4) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](
    v5 = 4 * *v4;
    LODWORD(v6) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[](
    *v3 = ((*v6 >> 6) | v5) ^ i;
  }
```

```
for ( i = 0; i <= 3; ++i )
  {
    for ( j = 1; j < strlen(s); ++j )
    {
      LODWORD(v1) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[
      v2 = v1;
      LODWORD(v3) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[
      v4 = *v3;
      LODWORD(v5) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[
      v6 = *v5 | v4;
      LODWORD(v7) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[
      v8 = *v7;
      LODWORD(v9) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::operator[
      *v2 = v6 & ~(v8 & *v9);
    }
  }
```

```
#!/usr/bin/env python
#coding=utf-8
la = [153,176,135,158,112,232,65,68,
      0x5,0x4,0x8b,0x9a,0x74,0xbc,0x55,0x58,
      0xb5,0x61,0x8e,0x36,0xac,0x9,0x59,0xe5,
      0x61,0xdd,0x3e,0x3f,0xb9,0x15,0xed,0xd5]
flag=''
def last():
    for j in range(4):
        for i in range(1,32):
            d=la[32-i]
            e = la[32-i-1]
            f = d^e
            la[32-i]=f

def first():
    for k in range(len(la)):
        #la[i]=(la[i]>>6|4*la[i])^i
        la[k]=la[k]^k
        la[k]=(la[k]<<6)&0xff|(la[k]>>2)
last()
first()
for i in range(len(la)):
    flag+=chr(la[i])
print flag
```

网上有些writeup写last()函数时照搬s[j]=s[j-1]|s[j]&(~(s[j]&s[j-1]))，其实它就相当于抑或，这个可以自己证明的。

但这题给我的警示是打比赛不要理所当然转化这种相当的运算公式，如果是错的，自己都发现不了。。（比如first()）