[XCTF-Reverse] 70 XX

原创

石氏是时试 于 2022-03-26 20:36:42 发布 154 收藏
分类专栏： CTF reverse 文章标签： reverse
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/weixin_52640415/article/details/123762679
版权


CTF reverse 专栏收录该内容
64 篇文章 0 订阅
订阅专栏

xxtea加密第一次遇见，不看不知道啊。

很容易跟到主函数，分N步加密，太狠了

第1步读入检查长度：从标准输入读入flag，然后计算长度，应为19。（这里的myinput是我改的，原来叫Code与全局变量重名，容易混淆）

```
*(_OWORD *)myinput = 0i64;
v33 = 0;
sub_13F8B18C0(std::cin, a2, (__int64)myinput);// 读入Code
myinput_len = -1i64;
v3 = -1i64;
do
  ++v3;
while ( *((_BYTE *)myinput + v3) );
if ( v3 != 19 )
{
  sub_13F8B1620(std::cout, "error\n");
  _exit((unsigned __int64)myinput);
}
```

第2步取密钥：先把前4字节与已知范围比较，key必需是在这个范围内（小写字母和数字）

```
v4 = (__int128 *)sub_13F8B1E5C(5ui64);
t_code = *(_QWORD *)&Code;                      // "qwertyuiopasdfghjklzxcvbnm1234567890"
v6 = v4;
v7 = 0;
v8 = v4;
do
{
  v9 = *((_BYTE *)v8 + (char *)myinput - (char *)v4);
  v10 = 0;
  *(_BYTE *)v8 = v9;
  v11 = 0i64;
  v12 = -1i64;
  do
    ++v12;
  while ( *(_BYTE *)(t_code + v12) );
  if ( v12 )
  {
    do
    {
      if ( v9 == *(_BYTE *)(t_code + v11) )
        break;
      ++v10;
      ++v11;
    }
    while ( v10 < v12 );
  }
  v13 = -1i64;
  do
    ++v13;
  while ( *(_BYTE *)(t_code + v13) );
  if ( v10 == v13 )
    _exit(t_code);
  v8 = (__int128 *)((char *)v8 + 1);
}
while ( (char *)v8 - (char *)v4 < 4 );          // 验证前4字节都在Code串中
*((_BYTE *)v4 + 4) = 0;                         // 保存前4位
```

第3步把key填充至16字节（一个简单的活用一堆指针来处理，真够意思，按代码行数给钱吧）

```
do
  ++myinput_len;
while ( *((_BYTE *)myinput + myinput_len) );
v14 = 0i64;
key = *v6;
while ( *((_BYTE *)&key + v14) )
{
  if ( !*((_BYTE *)&key + v14 + 1) )
  {
    ++v14;
    break;
  }
  if ( !*((_BYTE *)&key + v14 + 2) )
  {
    v14 += 2i64;
    break;
  }
  if ( !*((_BYTE *)&key + v14 + 3) )
  {
    v14 += 3i64;
    break;
  }
  v14 += 4i64;
  if ( v14 >= 0x10 )
    break;
}
for ( i = v14 + 1; i < 0x10; ++i )            // 将key填充0至16位
  *((_BYTE *)&key + i) = 0;
```

第4步xxtea加密（没看懂，不过名字叫xx估计这就是提示）

```
v16 = xxtea_13F8B1AB0((__int64)myinput, myinput_len, (unsigned __int8 *)&key, &Size);// 数据加密 xxtea
```

第5步修改顺序

```
*v19 = v18[2];
ptr_v19_1 = v19 + 1;
v19[1] = *v18;
v19[2] = v18[3];
v19[3] = v18[1];
v19[4] = v18[6];
v19[5] = v18[4];
v19[6] = v18[7];
v19[7] = v18[5];
v19[8] = v18[10];
v19[9] = v18[8];
v19[10] = v18[11];
v19[11] = v18[9];
v19[12] = v18[14];
v19[13] = v18[12];
v19[14] = v18[15];
v19[15] = v18[13];
v19[16] = v18[18];
v19[17] = v18[16];
v19[18] = v18[19];
v19[19] = v18[17];
v19[20] = v18[22];
v19[21] = v18[20];
v19[22] = v18[23];
```

第6步与前n/3个字节异或

```
for ( v19[23] = v18[21]; v20 < v17; ++ptr_v19_1 )// v20从1开始实际是从3开始
{
  v22 = 0i64;
  if ( v20 / 3 > 0 )
  {
    v23 = *ptr_v19_1;
    do
    {
      v23 ^= v19[v22++];
      *ptr_v19_1 = v23;
    }
    while ( v22 < v20 / 3 );                  // 与前n/3个异或
  }
  ++v20;
}
```

第7步目标串（两个8字节+两个4字节）

```
*(_QWORD *)&key = -456968194084773969 8i64;
v24 = v19 - (_BYTE *)&key;
*((_QWORD *)&key + 1) = 381988763664492 8495i64;
v25 = 0i64;
v30 = -939386845;
v31 = -95004953;
```

然后逆向(xxtea加解密函数从网上抄的，我不可能会)

```
#1 比较
```

```
#1,比较
v29 = list(b'\xce\xbc@k|:\x95\xc0\xef\x9b  \x91\xf7\x025#\x18\x02\xc8\xe7VV\xfa')
print(1, bytes(v29))
#2,异或
for i in range(23,0,-1):
    for j in range(i//3):
        v29[i]^=v29[j]
print(2, bytes(v29))
#3,换位
v19_v18 = [2,0,3,1,6,4,7,5,10,8,11,9,14,12,15,13,18,16,19,17,22,20,23,21]
v18 = [0]*len(v29)
for i,j in enumerate(v29):
    v18[v19_v18[i]] = j
print(3, bytes(v18))

#4,xxtea 解密
import struct
_DELTA = 0x9E3779B9

def _long2str(v, w):
    n = (len(v) - 1) << 2
    if w:
        m = v[-1]
        if (m < n - 3) or (m > n): return ''
        n = m
    s = struct.pack('<%iL' % len(v), *v)
    return s[0:n] if w else s



def _str2long(s, w):
    n = len(s)
    m = (4 - (n & 3) & 3) + n
    s = s.ljust(m, b"\0")
    v = list(struct.unpack('<%iL' % (m >> 2), s))
    if w: v.append(n)
    return v

def xxtea_encrypt(str, key):
    if str == '': return str
    v = _str2long(str, True)
    k = _str2long(key.ljust(16, b"\0"), False)
    n = len(v) - 1
    z = v[n]
    y = v[0]
    sum = 0
    q = 6 + 52 // (n + 1)
    while q > 0:
        sum = (sum + _DELTA) & 0xffffffff
        e = sum >> 2 & 3
        for p in  range(n):
            y = v[p + 1]
            v[p] = (v[p] + ((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z))) & 0xff
            z = v[p]
        y = v[0]
        v[n] = (v[n] + ((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4) ^ (sum ^ y) + (k[n & 3 ^ e] ^ z))) & 0xffffff
        z = v[n]
        q -= 1
    return _long2str(v, False)

def xxtea_decrypt(str, key):
```

```
    if str == '': return str
    v = _str2long(str, False)
    k = _str2long(key.ljust(16, b"\0"), False)
    n = len(v) - 1
    z = v[n]
    y = v[0]
    q = 6 + 52 // (n + 1)
    sum = (q * _DELTA) & 0xffffffff
    while (sum != 0):
        e = sum >> 2 & 3
        for p in range(n, 0, -1):
            z = v[p - 1]
            v[p] = (v[p] - ((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z))) & 0xff
            y = v[p]
        z = v[n]
        v[0] = (v[0] - ((z >> 5 ^ y << 2) + (y >> 3 ^ z << 4) ^ (sum ^ y) + (k[0 & 3 ^ e] ^ z))) & 0xffffff
        y = v[0]
        sum = (sum - _DELTA) & 0xffffffff
    return _long2str(v, True)

x = xxtea_decrypt(bytes(v18), b'flag')
print(x)
'''
1 b'\xce\xbc@k|:\x95\xc0\xef\x9b  \x91\xf7\x025#\x18\x02\xc8\xe7VV\xfa'
2 b'\xce\xbc@\xa5\xb2\xf4\xe7\xb2\x9d\xa9\x12\x12\xc8\xae[\x10\x06=\x1d\xd7\xf8\xdc\xdcp'
3 b'\xbc\xa5\xce@\xf4\xb2\xb2\xe7\xa9\x12\x9d\x12\xae\x10\xc8[=\xd7\x06\x1d\xdcp\xf8\xdc'
b'flag{CXX_and_++tea}'
'''
```

原比赛里flag的格式应该有声名。