[XCTF-Reverse] 37-39

原创

石氏是时试 已于 2022-03-21 20:05:13 修改 25 收藏
分类专栏： CTF reverse 文章标签： reverse
于 2022-03-20 00:15:00 首次发布
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/123600457
版权


CTF reverse 专栏收录该内容
64 篇文章 0 订阅
订阅专栏

37，XCTF 4th-QCTF-2018_babymips

mips这个不好搞了，先用ida打开（不能反编译成c）大概能看到他把输入数据左右移

整了半天插件也没安上，然后下了Retdec然后运行

```
py retdec-decompliler.py babymips
```

他会在程序同目录生成.c文件，其实也没比汇编多看多少，而且那个比较的串在.c里没有，还得从ida里看，反复理解就是先跟0x20-i异或再根据奇偶分别把前2位后6位互换/前6后2（除去头）

```
a = bytes.fromhex('52FD16A489BD9280134154A08D451881DEFC95F016791A155B751F')

flag = b"Q|j{g"
b = [0]*27                              #2，5位以后分奇偶前2后6/前6后2交换
for i,v in enumerate(a):
    if i%2 == 0:
        b[i] = v>>6 | v<<2 & 0xff
    else:
        b[i] = v>>2 | v<<6 & 0xff
flag +=bytes(b)
flag = ''.join([chr(v^(0x20-i)) for i,v in enumerate(flag)]) #1,输入值逐位与 0x20-i异或
print(flag)
#qctf{ReA11y_4_B@89_mlp5_4_XmAn_}
```

38，tu-ctf-2016_reverse-for-the-holy-grail-350

这个逆起来有点难度了。

主程序调用validstr和stringMod两个函数，第1个只是检查没啥用，第2个把18个字符每3个一组第1个给定，然后异或1，2得到3。

```
__int64 __fastcall stringMod(__int64 *a1)
{
  __int64 v1; // r9
  __int64 v2; // r10
  __int64 v3; // rcx
  signed int v4; // er8
```

```
int *v5; // rdi
int *v6; // rsi
signed int v7; // ecx
signed int v8; // er9
int v9; // er10
unsigned int v10; // eax
int v11; // esi
int v12; // esi
int v14[24]; // [rsp+0h] [rbp-60h]

memset(v14, 0, 0x48uLL);
v1 = a1[1];
if ( v1 )
{
  v2 = *a1;
  v3 = 0LL;
  v4 = 0;
  do
  {
    v12 = *(char *)(v2 + v3);
    v14[v3] = v12;
    if ( 3 * ((unsigned int)v3 / 3) == (_DWORD)v3 && v12 != firstchar[(unsigned int)v3 / 3] )
      v4 = -1;                                  // 能被3整除的要与firstchar相同
    ++v3;
  }
  while ( v3 != v1 );
}
else
{
  v4 = 0;
}
v5 = v14;
v6 = v14;
v7 = 666;
do
{
  *v6 = v7 ^ *(unsigned __int8 *)v6;
  v7 += v7 % 5;
  ++v6;
}
while ( &v14[18] != v6 );
v8 = 1;
v9 = 0;
v10 = 1;
v11 = 0;
do
{
  if ( v11 == 2 )
  {
    if ( *v5 != thirdchar[v9] )
      v4 = -1;
    if ( v10 % *v5 != masterArray[v9] )
      v4 = -1;
    ++v9;
    v10 = 1;
    v11 = 0;
  }
  else
  {
    v10 *= *v5;
```

```
      if ( ++v11 == 3 )
        v11 = 0;
    }
    ++v8;
    ++v5;
  }
  while ( v8 != 19 );
  return (unsigned int)(v7 * v4);
}
```

这个逆着不好来，每组爆破的方式处理

```
first = [0x41, 0x69, 0x6e, 0x45, 0x6f, 0x61, 0x00]
third = [0x2ef, 0x2c4, 0x2dc, 0x2c7, 0x2de, 0x2fc, 0]
master= [0x1d7, 0xc, 0x244, 0x25e, 0x93, 0x6c]
v7 = [666]*24
for i in range(1,24):
    v7[i] = v7[i-1]+(v7[i-1]%5)
#print(v7)

def get3(i):
    c1 = first[i]
    c3 = third[i]^v7[i*3+2]
    c2 = 0
    #print(c1,c3,v10)
    for tc2 in range(30,255):
        tmp = (c1^v7[i*3])*(tc2^v7[i*3+1])
        #print(tmp,third[i], tmp%third[i],master[i])
        if (tmp%third[i]) == master[i]:
            c2 = tc2
            print(chr(c1)+chr(c2)+chr(c3), end='')
            break

for i in range(6):
    get3(i)

#AfricanOrEuropean?
#tuctf{AfricanOrEuropean?}
```

39，suctf-2016_android-app-100

这个比较坑，是个apk文件，先解包得到class.dex和libadnjni.so前边是flag的打包方法，后边是输入串校验方法。

dex有不会整，jeb安不上，先用dex2jar打成jar包再用jd-jui看，大概意思就是把一个数加上输入串被处理后的返回值加上个空格的md5

```
class a implements View.OnClickListener {
  a(MainActivity paramMainActivity) {}

  public void onClick(View paramView) {
    new String(" ");
    String str = this.a.b.getText().toString();
    Log.v("EditText", this.a.b.getText().toString());
    new String("");
    int i = this.a.processObjectArrayFromNative(str);
    int j = this.a.IsCorrect(str);
    str = String.valueOf(this.a.d + i) + " ";
    try {
      MessageDigest messageDigest = MessageDigest.getInstance("MD5");
      messageDigest.update(str.getBytes());
      byte[] arrayOfByte = messageDigest.digest();
      StringBuffer stringBuffer = new StringBuffer();
      for (i = 0;; i++) {
        if (i >= arrayOfByte.length) {
          if (j == 1 && this.a.e != "unknown")
            this.a.c.setText("Sharif_CTF(" + stringBuffer.toString() + ")");
          if (j == 1 && this.a.e == "unknown")
            this.a.c.setText("Just keep Trying :-)");
          if (j == 0) {
            this.a.c.setText("Just keep Trying :-)");
            return;
          }
          break;
        }
        stringBuffer.append(Integer.toString((arrayOfByte[i] & 0xFF) + 256, 16).substring(1));
      }
    } catch (NoSuchAlgorithmException noSuchAlgorithmException) {
      noSuchAlgorithmException.printStackTrace();
    }
  }
}
```

而这个函数是.so里边，这个函数很长长，是个一大循环，里边用一个大数字字作各个分支的入口，先与指定串比较，然后第3次将一个返回值赋值，然后第6次退出。这个如果有模拟器的话安上apk输入这个给定的值就应该能出结果。如果没有就手工走6次循环打到一个隐藏很深的返回值。

```
'''
onClick->processObjectArrayFromNative
ret = processObjectArrayFromNative( 'ef57f3fe3cf603c03890ee588878c0ec' )
flag = 'Sharif_CTF(' + md5(str(d + ret) + ' ')  + ')'
'''



#v30 = bytes([101,102,53,55,102,51,102,101,51,99,102,54,48,51,99,48,51,56,57,48,101,101,53,56,56,56,55,56,9
#print(v30)
m = str(114366+ 0x57cbbd2)+' '
from hashlib import md5
print('Sharif_CTF('+ md5(m.encode()).hexdigest() +')')
#Sharif_CTF(833489ef285e6fa80690099efc5d9c9d)
```

这个壳是圆的。

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖