几道攻防世界安卓逆向题要点总结 (1)

作者：jyl
原文链接：https://ld246.com/article/1622162372546
来源网站：链滴

备忘一下，逆向做不出来了，这两天耍了几道安卓逆向。

不知不觉攻防世界等级都刷到高手了，400多分，不过有四分之三都是看wp抄出来的吧
weat_smile

APK逆向

应该是最简单的安卓逆向题了，条件都在mainactivite里给了，用一段字符串去md5获得hex后，每两位取值。

```
# author jinyunlong
# createtime 2021/5/16 23:23
# 职业   ICBC锅炉房保安
import hashlib
import string

def encrypt_md5(str):
    md=hashlib.md5()
    md.update(str.encode(encoding='utf-8'))
    print(md.hexdigest())
    # return md.digest()  #byte
    return md.hexdigest()  #hex

s = 'Tenshine'
s = encrypt_md5(s)

flag = ''
for i in range(0,len(s),2):
    flag +=str(s[i])
print(flag)
```

人民的名义-抓捕赵德汉1-200

关键代码在CheckPass类里，是段md5，解了密就是inputstring

```
public class CheckPass implements CheckInterface {
  public boolean checkPassword(String input) {
    MessageDigest md5Obj = null;
    try {
      md5Obj = MessageDigest.getInstance("MD5");
    } catch (NoSuchAlgorithmException e) {
      System.out.println("Hash Algorithm not supported");
      System.exit(-1);
    }
    byte[] hashBytes = new byte[40];
    md5Obj.update(input.getBytes(), 0, input.length());
    hashBytes = md5Obj.digest();
    return byteArrayToHexString(hashBytes).equals("fa3733c647dca53a66cf8df953c2d539");
  }
```

基础android

考察了安卓广播方法的使用:sendBroadcast(发广播方法) receiver(注册广播) action android:name= xxxxxxxxxxxxxxxx(输入内容)"

在这之前的考点是在12次循环中的pass[len] = (char) (((255 - len) - 100) - pass[len]);表达式计算之每一位的pass[len]的值必须等于字符‘0’的char值。'0'的十进制就是48，用这个算出的12位key进第一次提交到第二个onclick事件，第二个事件触发的就是发广播sendBroadcast方法了，然后输入注广播内的action android:name="xxxxxxxxxxxxxxxx(输入内容)"就会跳到给flag的方法了

其实这题还有很多其他解法，可以看大佬这篇攻防世界-Mobile-基础android

有时间是要做下安卓开发，还挺好玩的。

Android2.0

一道so库逆向，用ida打开有几个明显函数分别是first，second，third对应三段字符串，有个坑是这段伪代码都是处理四位字符，剩下的位数不处理，写脚本时要注意，最后在init方法里按位置拼接了。

```
# author jinyunlong
# createtime 2021/5/17 10:14
# 职业   ICBC锅炉房保安
s1 = ''
t1 = 'LN^d'

for i in t1:
    s1 += chr((ord(i) ^ 0x80) // 2)
s1 += 'l'
print(s1)

s2 = ''
t2 = [0x20, 0x35, 0x2d, 0x16]
for i in range(len(t2)):
    s2 += chr(ord(t1[i]) ^ t2[i])
s2 += chr(0x61)
```

```
print(s2)

s3 = ''
t3 = 'AFBo'
for i in range(len(t3)):
    s3 += chr(t2[i] ^ ord(t3[i]))
s3 += '}'
print(s3)

flag = ''
for i in range(5):
    flag += s1[i] + s2[i] + s3[i]
print(flag)
```

APK逆向-2

逆向里也有道题叫这名字，谁能想到这道题是改mainfest.xml文件呢。文件格式的事。反编译apk后堆花里胡哨的东西，然后装apk也装不上，看了wp写是mainfest.xml文件加载失败了，用jadx一看还是，根本没加载出来，然后参考几个正常的mainfest文件，用010十六进制打开发现这个文件的第一第8到第11个字符和第三行的第三个字符都不对劲，修改后重新放到apk目录结构里，重新打回apk包编译打开可正常显示了action android:name="8d6efd232c63b7d2"就是答案

app3

没做出来，不过学会了使用Androidbackup_extractor去反查ab文件，jadx的反混淆。

easyjni和easy-so

放一块说吧两道题一样(easy-so比easyjni还简单，做出来的人却更少。。？？)

就拿so说吧

主要逻辑在so库文件里，关键代码在Java_com_testjava_jack_pingan2_cyberpeace_CheckString有两段字符串操作一段是前后16个字节换位置，还一个是两两互换。还原回来就是flag

easyjni有个base64码表替换后加密的步骤

贴下各种换位置的py

open_mouth

```
# author jinyunlong
# createtime 2021/5/17 19:30
# 职业   ICBC锅炉房保安
s = 'f72c5a36569418a20907b55be5bf95ad'
temp=[]
for i in range(0,len(s)):
    temp.append(s[i])
print(temp)

for j in range(0,len(temp),2):
    k = ''
    k = temp[j]
```

```
    temp[j] = temp[j+1]
    temp[j+1] = k
print(temp)

for m in range(0,len(temp)//2):
    n = ''
    n = temp[m]
    temp[m] = temp[m+16]
    temp[m+16] = n

flag = ''.join(temp)
print(flag)
```

哎，记一下吧，怕忘了，今天看，明天忘，明天再看，后天再忘，可能年纪轻轻就也该到了吃脑白金年龄了doge doge doge。