作者 0h1in9e 2020.04.16 09:42:00
写了39篇文章,回复129人,

Wechat&Alipay小程序源码反编译

0x01 前言

随着小程序的火爆，越来越多的企业开始开发自身的小程序，小程序便成了我们挖洞的一个切入点。常用的平台有微信、支付宝、百度app等。日常渗透、众测中经常会遇到小程序，在获取到小程序客户端源码的情况下，对于Bypass数据包签名、GET敏感接口等都有一定的帮助。这里对最常遇到的微信小程序、支付宝小程序的源码反编译做个方法记录。

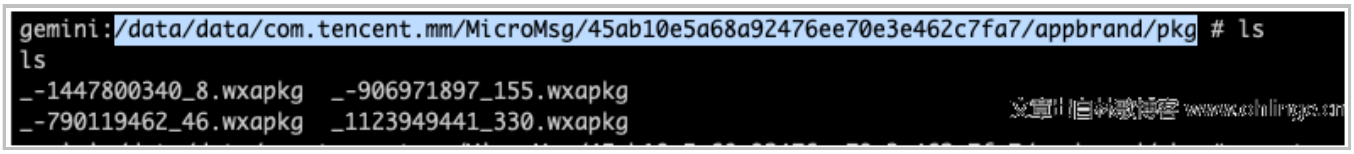
0x02 需要的准备

1. 安卓手机（需ROOT）
2. PC安装adb套件

0x03 微信小程序

1. 微信使用小程序
2. 手机连接电脑，注意手机开启USB调试模式，可用电脑命令行下运行 `adb devices` 来确认是否连接成功。
3. 进入微信小程序.wxapkg文件存放位置：

```
➜  ~ adb shell
    $: su
    # cd /data/data/com.tencent.mm/MicroMsg/{user}/appbrand/pkg
// 注意这里的{user}通常是一段hash
    # ls -lt
// 利用lt排序可以将最近使用的小程序排到前面
```



4. 导出小程序：

> 通常利用android手机的/sdcard/作为中间目录来导出wxapkg文件

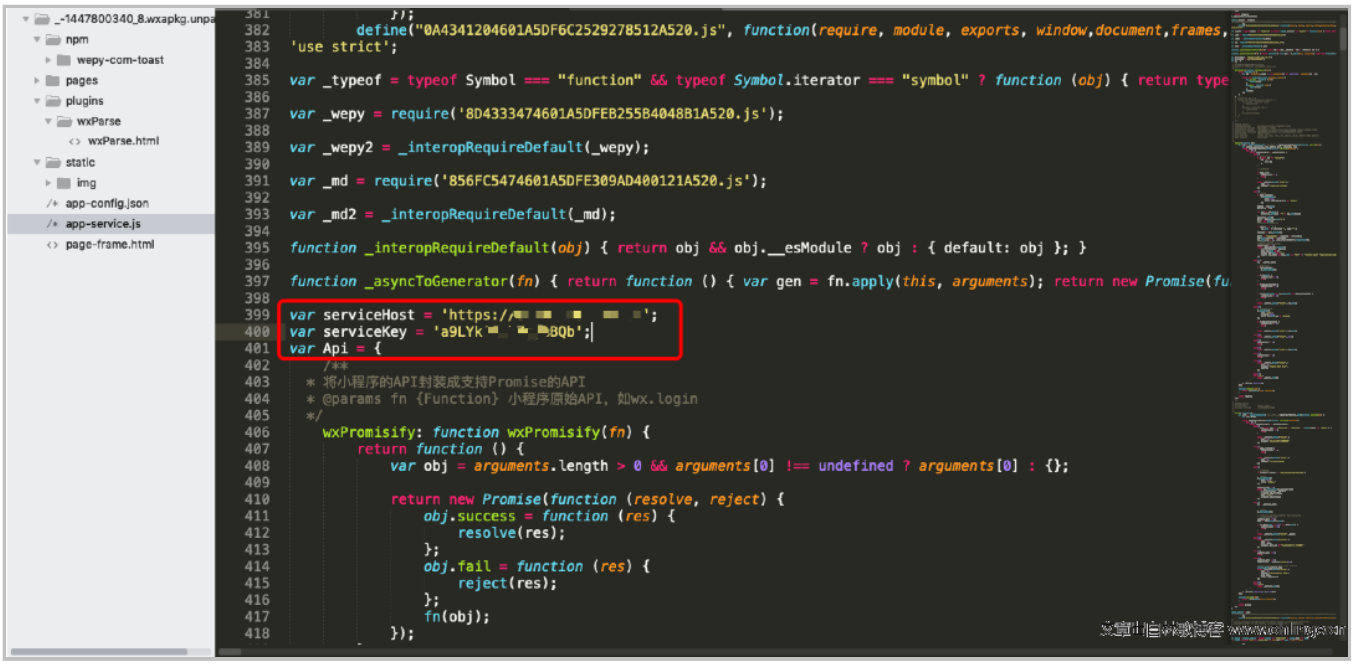
```
    # cp _-1447800340_8.wxapkg /sdcard/
// 退出adb shell
```

3. .wxapkg文件反编译

> 如上已经将wxapkg文件导出到电脑本地，此时需要利用脚本将其解压，即可得到最终的源码文件

```
#!/usr/bin/python

# usage python wxapkg_unpack.py filename, unpack at filename.unpack

import sys,os
import struct

class WxapkgFile:
  nameLen = 0
  name = ""
  offset = 0
  size = 0

with open(sys.argv[1], "rb") as f:

  root = os.path.dirname(os.path.realpath(f.name))
  name = os.path.basename(f.name)

  #read header

  firstMark = struct.unpack('B', f.read(1))[0]
  print 'first header mark = ' + str(firstMark)

  info1 = struct.unpack('>L', f.read(4))[0]
  print 'info1 = ' + str(info1)

  indexInfoLength = struct.unpack('>L', f.read(4))[0]
  print 'indexInfoLength = ' + str(indexInfoLength)

  bodyInfoLength = struct.unpack('>L', f.read(4))[0]
  print 'bodyInfoLength = ' + str(bodyInfoLength)

  lastMark = struct.unpack('B', f.read(1))[0]
  print 'last header mark = ' + str(lastMark)

  if firstMark != 190 or lastMark != 237:
    print 'its not a wxapkg file!!!!!'
    exit()

  fileCount = struct.unpack('>L', f.read(4))[0]
  print 'fileCount = ' + str(fileCount)

  #read index

  fileList = []

  for i in range(fileCount):

    data = WxapkgFile()
    data.nameLen = struct.unpack('>L', f.read(4))[0]
    data.name = f.read(data.nameLen)
    data.offset = struct.unpack('>L', f.read(4))[0]
    data.size = struct.unpack('>L', f.read(4))[0]

    print 'readFile = ' + data.name + ' at Offset = ' + str(data.offset)

    fileList.append(data)

  #save files

  for d in fileList:
```

```
65        if not os.path.exists(path):
66          os.makedirs(path)
67
68        w = open(root + d.name, 'w')
69        f.seek(d.offset)
70        w.write(f.read(d.size))
71        w.close()
72
73        print 'writeFile = ' + root + d.name
74
75      f.close()
```



0x04 支付宝小程序

大致思路与微信小程序一致，需要注意的是支付宝小程序的存放位置不同,位置如下。另外具体目录名为小程序tinyAppId值，其中的tar包即为源码文件。tar包未加密，adb pull出来之后直接解压即可

```
1  # cd /data/data/com.eg.android.AlipayGphone/files/nebulaInstallApps/
```

```
gemini:/data/data/com.eg.android.AlipayGphone/files/nebulaInstallApps/2018053160289430/ada65aac4c128
efe46c6112769655abd # ls -la
ls -la
total 884
drwx------ 3 u0_a123 u0_a123   4096 2020-02-17 23:00 .
drwx------ 3 u0_a123 u0_a123   4096 2020-02-17 23:00 ..
-rw------- 1 u0_a123 u0_a123 880640 2020-02-17 23:00 2018053160289430.tar
-rw------- 1 u0_a123 u0_a123    595 2020-02-17 23:00 CERT.json
-rw------- 1 u0_a123 u0_a123    179 2020-02-17 23:00 Manifest.xml
-rw------- 1 u0_a123 u0_a123    754 2020-02-17 23:00 SIGN.json
drwx------ 2 u0_a123 u0_a123   4096 2020-02-17 23:00 ariver_ext_0.2.2002061904.37
gemini:/data/data/com.eg.android.AlipayGphone/files/nebulaInstallApps/2018053160289430/ada65aac4c128
efe46c6112769655abd #
```

导出方法一致，最终导出来之后效果图如下：

0x05 js美化

小程序源码使用的是js，而通常会加密、混淆等，如上图的js文件，对于我们分析会有一定的阻碍。方法还是有的，我们只需要对js进行美化即可帮助我们分析，这里推荐一个js美化在线工具：

https://tool.lu/js/

0x06 参考链接

https://www.52pojie.cn/thread-1050690-1-1.html

http://lrdcq.com/me/read.php/66.htm

https://www.jianshu.com/p/3678c5b41636

标签： 小程序