

Wechat&Alipay小程序源码反编译

阅读: 1681 · 评论: 0 · 喜欢: 5

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:

```
1 → ~ adb shell

2 $: su

3 # cd /data/data/com.tencent.mm/MicroMsg/{user}/appbrand/pkg

4 // 注意这里的{user}通常是一段hash

5 # ls -lt

6 // 利用lt排序可以将最近使用的小程序排到前面
```

```
gemini:/data/data/com.tencent.mm/MicroMsg/45ab10e5a68a92476ee70e3e462c7fa7/appbrand/pkg # ls
ls
_-1447800340_8.wxapkg _-906971897_155.wxapkg
_-790119462_46.wxapkg _1123949441_330.wxapkg
```

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

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

J. .WAUPING ATT AMET

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

```
1
    #!/usr/bin/python
 2
 3
     # usage python wxapkg unpack.py filename, unpack at filename.unpack
 4
 5
     import sys, os
 6
     import struct
 7
 8
    class WxapkgFile:
 9
      nameLen = 0
       name = ""
10
11
       offset = 0
12
       size = 0
13
14
    with open(sys.argv[1], "rb") as f:
15
16
       root = os.path.dirname(os.path.realpath(f.name))
17
       name = os.path.basename(f.name)
18
19
       #read header
20
21
       firstMark = struct.unpack('B', f.read(1))[0]
22
       print 'first header mark = ' + str(firstMark)
23
24
       info1 = struct.unpack('>L', f.read(4))[0]
25
       print 'info1 = ' + str(info1)
26
27
       indexInfoLength = struct.unpack('>L', f.read(4))[0]
28
       print 'indexInfoLength = ' + str(indexInfoLength)
29
30
       bodyInfoLength = struct.unpack('>L', f.read(4))[0]
31
       print 'bodyInfoLength = ' + str(bodyInfoLength)
32
33
       lastMark = struct.unpack('B', f.read(1))[0]
34
       print 'last header mark = ' + str(lastMark)
35
36
       if firstMark != 190 or lastMark != 237:
37
        print 'its not a wxapkg file!!!!!'
38
         exit()
39
40
       fileCount = struct.unpack('>L', f.read(4))[0]
       print 'fileCount = ' + str(fileCount)
41
42
43
       #read index
44
45
       fileList = []
46
       for i in range(fileCount):
47
48
49
         data = WxapkgFile()
50
         data.nameLen = struct.unpack('>L', f.read(4))[0]
51
         data.name = f.read(data.nameLen)
         data.offset = struct.unpack('>L', f.read(4))[0]
52
53
         data.size = struct.unpack('>L', f.read(4))[0]
54
55
         print 'readFile = ' + data.name + ' at Offset = ' + str(data.offset)
56
57
         fileList.append(data)
58
59
       #save files
60
       for d in fileList:
61
```

0 最新 web安全 探索 **Q ♀** 登录

```
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()
```

```
;;;
define("0A4341204601A5DF6C2529278512A520.js", function(require, module, exports, window,document,frames,
'use strict';
▼ 🚞 _-1447800340_8.wxapkg.unp
  var _typeof = typeof Symbol === "function" && typeof Symbol.iterator === "symbol" ? function (obj) { return typeof
 ▶ IIII pages
 v 🚞 plugins
                                       var _wepy = require('8D4333474601A5DFEB255B4048B1A520.js');
   ▼ 🚞 wxParse
       wxParse.html
                                       var wepy2 = interopRequireDefault( wepy);
 v 🚞 static
   ⊧ 📗 img
                                       var _md = require('856FC5474601A5DFE309AD400121A520.js');
   /* app-config.json
                                       var _md2 = _interopRequireDefault(_md);
  /* app-service.js
   page-frame.html
                                       function _interopRequireDefault(obj) { return obj && obj.__esModule ? obj : { default: obj }; }
                                       function _asyncToGenerator(fn) { return function () { var gen = fn.apply(this, arguments); return new Promise(fu
                                398
399
400
401
402
403
404
405
406
407
408
410
411
412
413
414
415
416
417
418
                                      var serviceHost = 'https://■■■■
var serviceKey = 'a9LYk ■ '■ B0b';
var Api = {
                                            wxPromisify: function wxPromisify(fn) {
   return function () {
      var obj = arguments.length > 0 66 arguments[0] !== undefined ? arguments[0] : {};
                                                           urn new Promise(function (resolve, reject) {
  obj.success = function (res) {
    resolve(res);
                                                            };
obj.fail = function (res) {
    reject(res);
                                                            };
fn(obj);
                                                                                                                                                                   交重出自然激展客 www.codingso
```

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tinyAppld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
                              4096 2020-02-17 23:00 ariver_ext_0.2.2002061904.37
drwx----- 2 u0_a123 u0_a123
gemini:/data/data/com.eg.android.AlipayGphone/files/nebulaInstallApps/2018053146028943044da65aac4c128
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

添加新评论

如评论不显示,请等候管理员人工审核			
	称呼 *		提交评论
	邮箱 *		
http://	网站		

0