微信小程序&公众号安全渗透测试指南V1.3

**—— 安全团队**

**目 录**

[一、 概述 2](#_Toc51156203)

[二、 工具介绍 2](#_Toc51156204)

[1. 反汇编/反编译工具 2](#_Toc51156205)

[2. 调试工具 2](#_Toc51156206)

[三、 信息收集 2](#_Toc51156207)

[1. 小程序的ID与域名收集 2](#_Toc51156208)

[2. 反编译 7](#_Toc51156209)

[3. Sessionkey的判断 13](#_Toc51156210)

[4. 传输内容加解密 14](#_Toc51156211)

[5. 数据库破解 15](#_Toc51156212)

[四、 漏洞挖掘 17](#_Toc51156213)

[1. 信息泄露类漏洞 17](#_Toc51156214)

[1.1. AppSecret泄露 17](#_Toc51156215)

[1.2. 请求直接返回SessionKey泄露 19](#_Toc51156216)

[2. Js\_code重复使用 21](#_Toc51156217)

[3. CWRF漏洞 22](#_Toc51156218)

[4. 供应链漏洞 25](#_Toc51156221)

[3.1. 第三方组件问题 25](#_Toc51156223)

[3.2. 第三方授权问题 27](#_Toc51156224)

[5. WEB相关漏洞 28](#_Toc51156225)

[4.1. 需要关注的 28](#_Toc51156227)

[4.2. 不需要关注的 29](#_Toc51156228)

[五、 附件 29](#_Toc51156229)

# 概述

本手册主要针对微信小程序与公众号渗透测试工具、方法进行收集、整理，并结合在线资料及测试实例，旨在为测试人员提供漏洞测试的基本思路，指导微信小程序与公众号渗透测试工作。

# 工具介绍

## 反汇编/反编译工具

1. **夜神：**夜神版本andriod 7.0以下
2. **ADB**
3. **wxappUnpacker**

一款微信小程序反编译工具， 基于nodejs的反编译工具。

链接：<https://github.com/xuedingmiaojun/wxappUnpacker>

## 调试工具

1. **微信小程序开发工具**
2. **Xposed**
3. **Sqlitecipher爆破查看工具**

# 信息收集

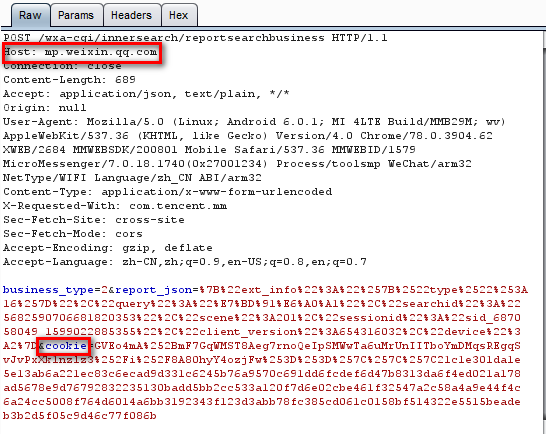
## 小程序的ID与域名收集

（1）批量搜索相关小程序ID

A.获取cookie信息：



抓包获得cookie

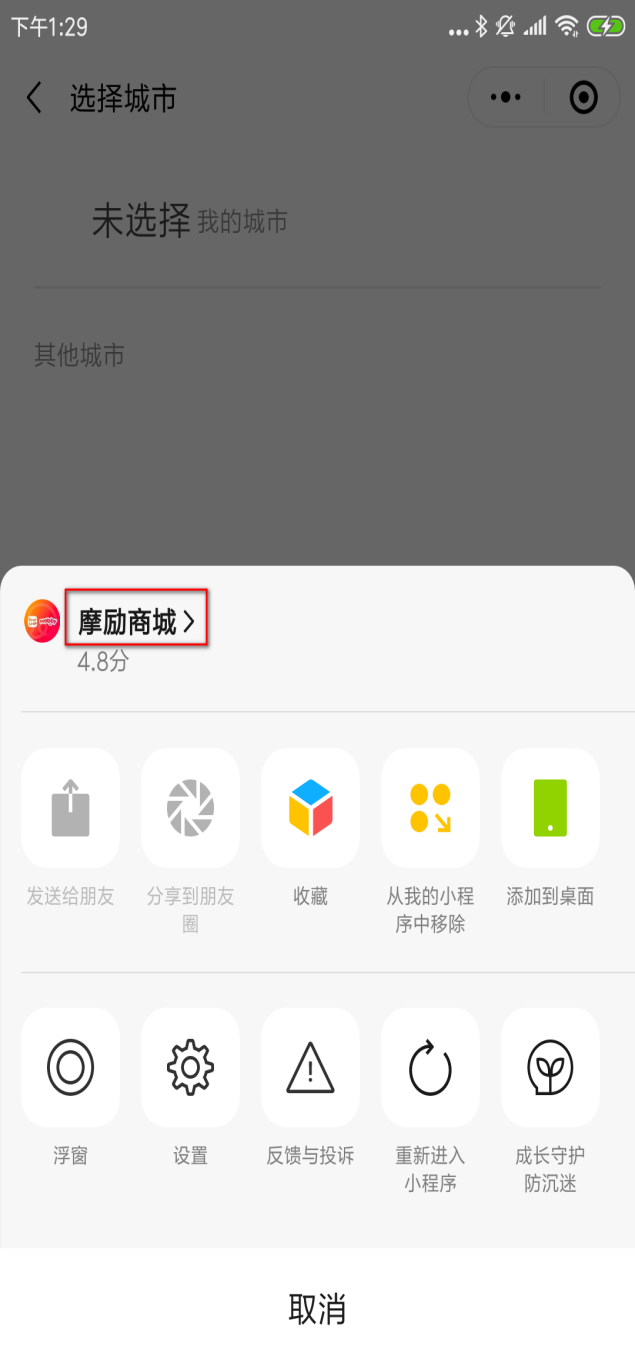


B.执行搜素脚本：获得涉及的小程序名称及appid

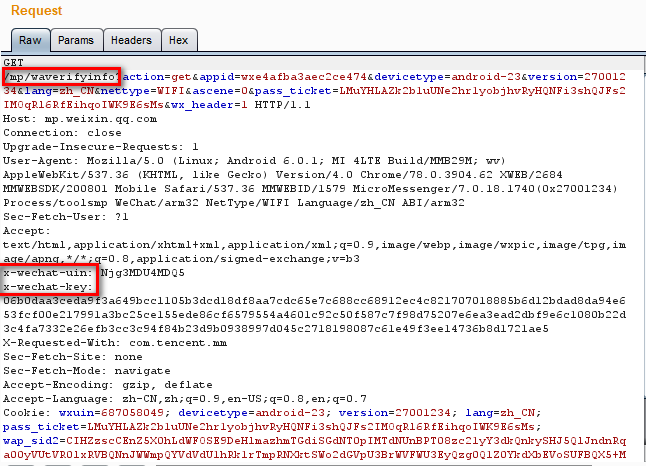


（2）批量搜索小程序对应的域名

A.获取小程序对应域名

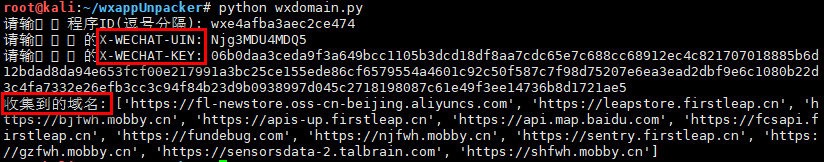






查询接⼝为“https://mp.weixin.qq.com/mp/waverifyinfo” 。其中 有“action”、“appid”、“wx\_header”三个数据，在请求头中还含有“X-WECHAT-KEY”、“X-WECHATUIN”两个微信⾃定义的请求头。“action”为请求类型，这边使⽤默认“get”值；“wx\_header”为是否启⽤ 微信⾃定义头，也使⽤默认参数“1”启⽤即可；“appid”为对应项查询的微信⼩程序ID值；“X-WECHATUIN”为每个客户端固定的身份值；“X-WECHAT-KEY”是微信⽤于验证请求合法性时所⽤的校验值，每次访问会重新⽣成新的校验值，但旧校验值并不会⽴刻失效，任然拥有3⼩时的有效期。

B. 执行搜素脚本：获得小程序涉及的域名

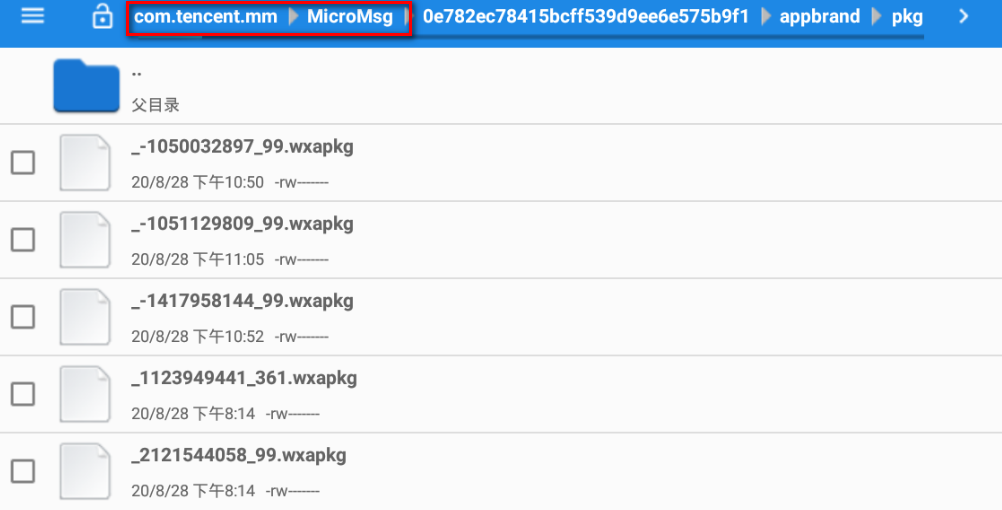


## 反编译

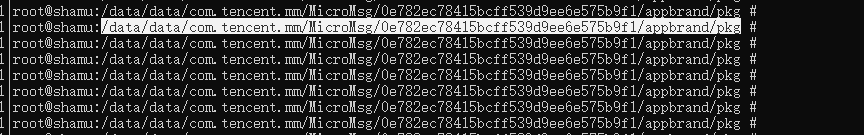
（1）导出wxapkg文件

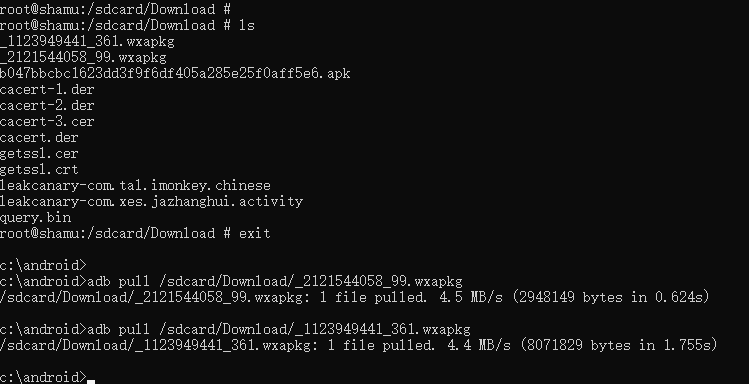
A. 通过微信运行小程序，然后点击所有小程序中的使用功能（目的是将全部wxapkg导出）

B.查看文件导出位置：/data/data/com.tencent.mm/MicroMsg/（小程序对应随机字符串目录）/appbrand/pkg



C.通过ADB导出wxapkg文件：cat \_xxxx.wxapkg > /sdcard/Download/xxxxx.wxapkg

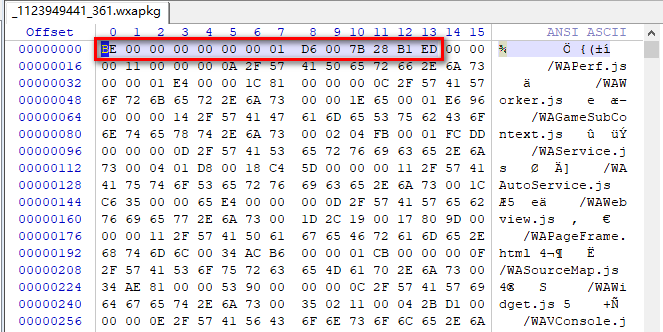




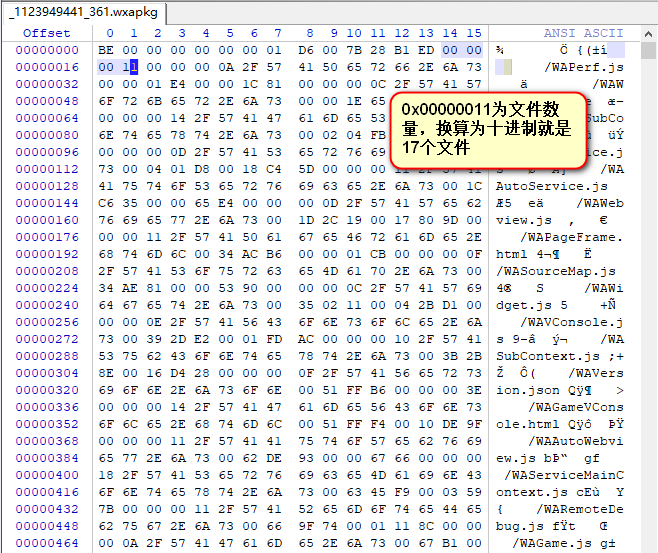
D.文件导出完成

（2）文件结构讲解

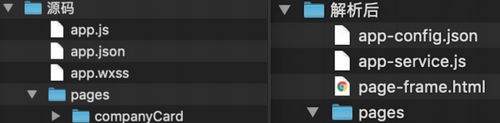
A.⼩程序包的头部段都由 0xBE 幻数（Magic Number）开头以 0xED 幻数做结尾



B. 索引段以占位4字节的“数量块”开头，代表当前包内共由 多少个⽂件



C.小程序组成结构（开发代码与发布后）

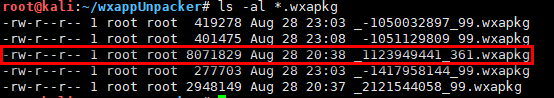


在微信服务器会将⼩程序源码中所有的“js”⽂件压⼊“app-service.js”⽂件中，将所有的“json”⽂件 压⼊“app-config.json”中，将所有的“wxml”⽂件压⼊“page-frame.html”⽂件中，“wxss”则在处理之后 以“html”⽂件的形式存留在对应⻚⾯⽬录之下。

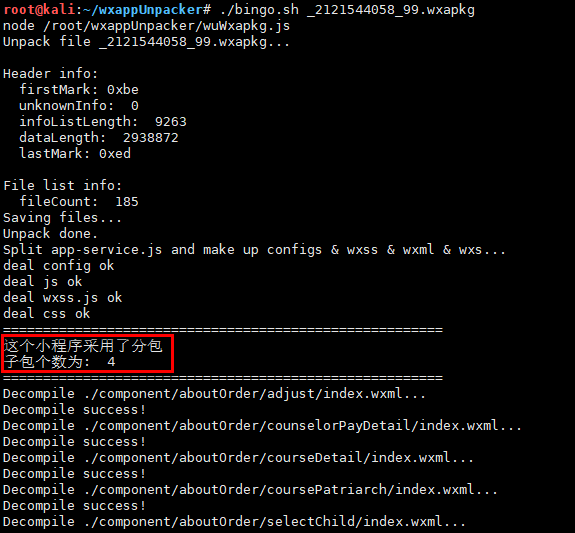
反编译要做的事情：JS数据还原、JSON数据还原、WXSS数据还原、WXML数据还原

（3）使用wxappUnpacker反编译

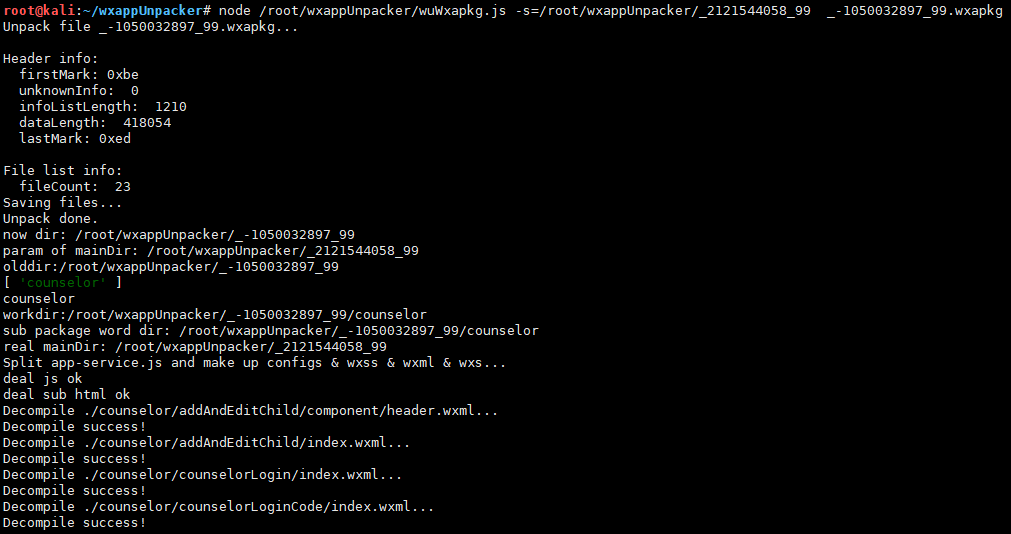
A.导出的文件：大致10M这个为微信基础包不需要反编译，\_-这个形式大致为子包，属于多个包分包情况

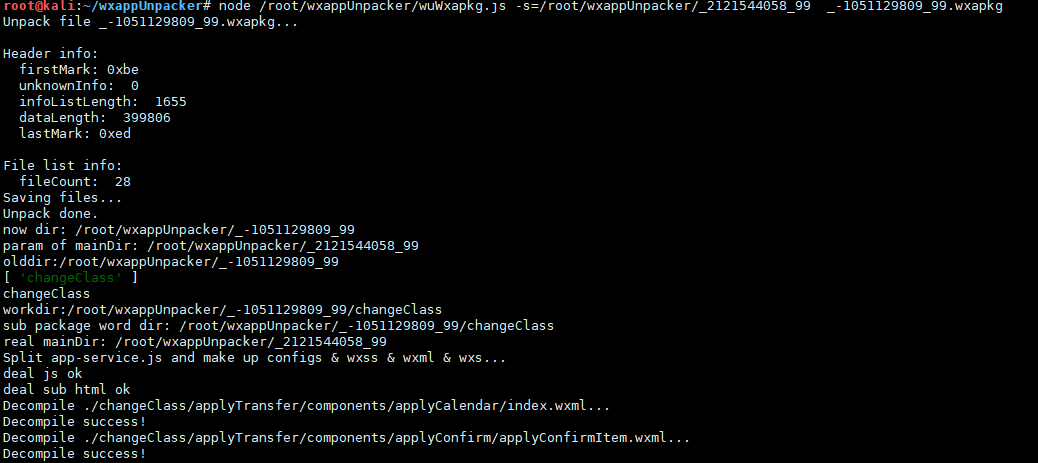


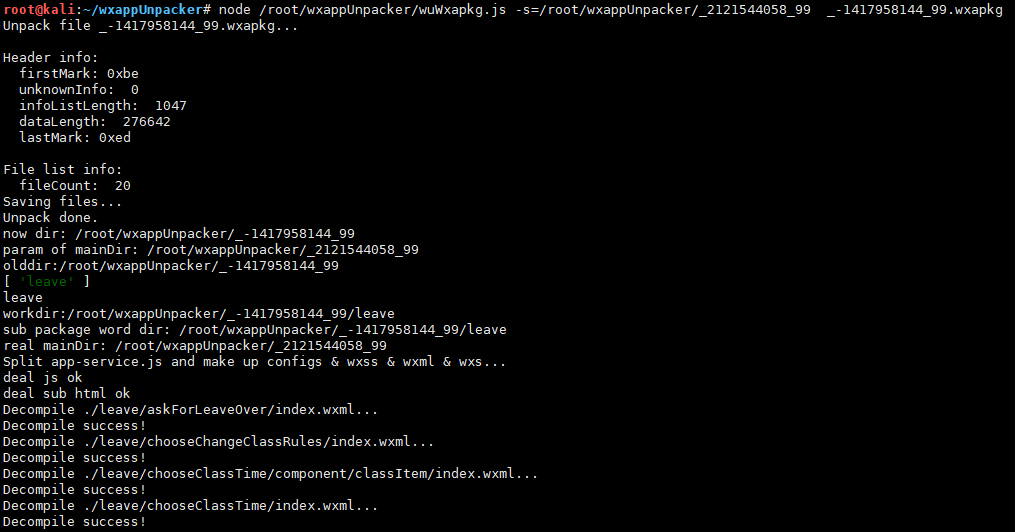
B.对于主包进行反编译，知道存在4个子包的情况（如果发现子包不全，继续操作小程序功能，看是否有导出，很多子包是需要被操作到才会导出的）



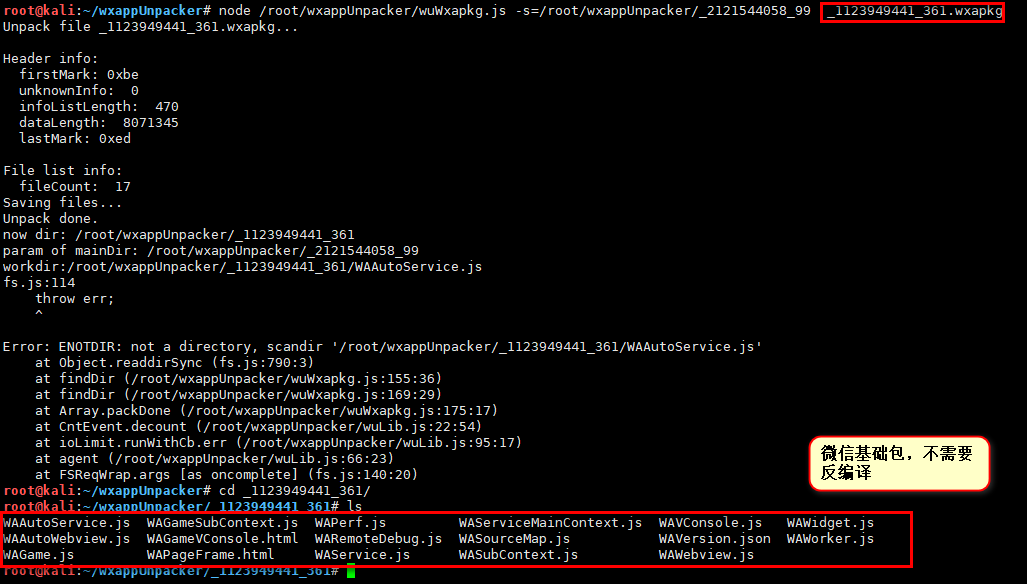
C.对子包进行反编译操作





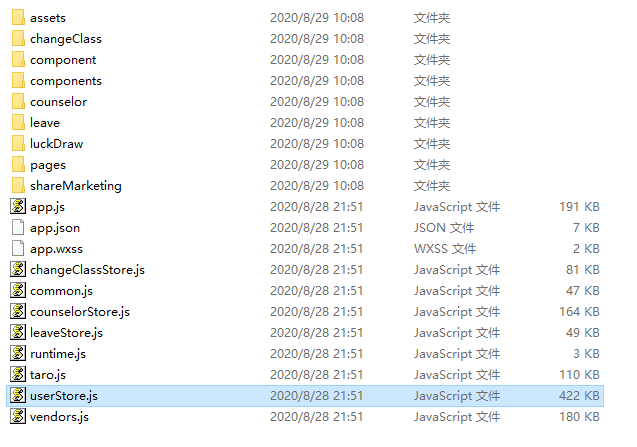


D.微信基础包不需要反编译的判断方法



E.获得反编译后的结果





## Sessionkey的判断

（1）从代码中获得的appid、secret，从抓包中获得js\_code

（2）从抓包中获得js\_code，appid、secret

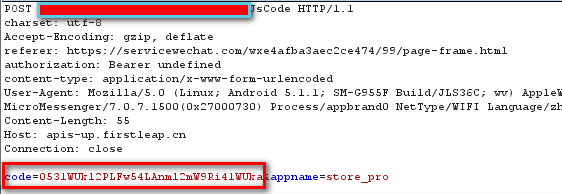
（3）从代码中获得的appid，从抓包中获得js\_code，secret

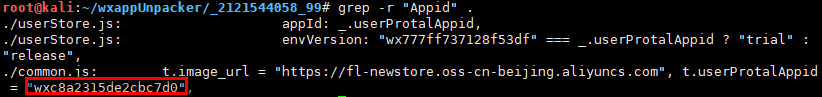
微信官⽅ jscode2session的API接口：

[https://api.weixin.qq.com/sns/jscode2session?appid={0}&secret={1}&js\_code={2}&grant\_type=authorization\_code](https://api.weixin.qq.com/sns/jscode2session?appid=%7b0%7d&secret=%7b1%7d&js_code=%7b2%7d&grant_type=authorization_code)

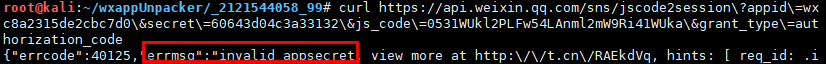
例如：验证secret是否正确:

A.抓包获得js\_code



B.代码中获得appid

C.从代码或抓包中找到secret，这里未找到，构造的secret，提示错误如图1，如果找到确认是正确返回如图2



图[1]



图[2]

## 传输内容加解密

前提：获得到session\_key

微信使⽤的是AES加密，CBC模式采⽤PKCS7填充，数据块⻓ 度为128位，输出使⽤base64编码。AES是⽤于替代DES加密的对称加密算法，对称加密算法最⼤的特 性便是加、解密使⽤的同是⼀串密钥，只要获取到key便可加解⾃如（CBC加密模式下还需要 iv 偏移量参数，⼀般固定不变或作为变量内容传输）

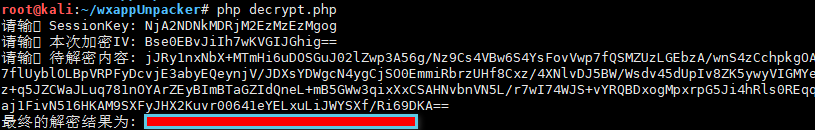
仅在“获取⽤户信息”功能中才会⽣成签名，⽤于防止内容被篡改，其他没有签名，例如：“获取⼿机号”功能中并⽆，因为签名是采⽤ sha1( rawData + sessionkey ) ⽅式⽣成的，若session\_key 泄露则⼀切验证签名的操作均无效的。

例如：解密内容

A.获得IV及加密内容，通过抓包获得



B.运行输入：session\_key、IV、加密内容



## 数据库破解

前提：微信7版本及以下

目的：为了读取数据和篡改聊天记录

聊天记录存放：/data/data/com.tencent.mm/MicroMsg/${userid}/EnMicroMsg.db

微信数据库使用sqlcipher方法进行加密的，EnMicroMsg.db是个加密的SQLite数据库。

破解方法：获取加密密码

1、动态调试获取密码（尽量使用不用的微信号，这个动态调试很容易封号）

（1）iOS 采用Xcode⾃带的 LLDB 调试器下断点调试获得

（2）Andriod 采用xposed hook方式：

Git地址：

<https://github.com/1998lixin/WeChat-database/tree/dev>

<https://github.com/ninecents/AndroidWechatSQLiteDecrypt>

2、本地暴力破解获取

Git地址：<https://github.com/chg-hou/EnMicroMsg.db-Password-Cracker>

3、sqlite3 cipher访问方式

Sqlite3访问工具：

1、<https://github.com/pawelsalawa/sqlitestudio/releases>

2、[https://sqlitebrowser.org/dl/#nightly-builds](https://sqlitebrowser.org/dl/" \l "nightly-builds)

例如：解密数据库，使用动态效果更好一些，本地爆破风险更低一下，这里使用本地爆破。

A.使用adb导出EnMicroMsg.db

聊天记录存放位置：/data/data/com.tencent.mm/MicroMsg/${userid}/EnMicroMsg.db

Userid构成：MD5('mm'+微信UIN)

微信的UIN存放在微信文件夹：“/data/data/com.tencent.mmshared\_prefs/system\_config\_prefs.xml”

找到后生成MD5找到对应的文件夹，通过adb导出到本地

B.使用本地暴力破解工具破解得到密码

C.使用SqLiteStudio访问



# 漏洞挖掘

## 信息泄露类漏洞

### AppSecret泄露

漏洞描述：

在⼩程序包内的配置⽂件中、博⽂、 GitHub仓库等等，找到被泄露的AppSecret。

漏洞场景：

小程序代码。

漏洞级别：

**【高危】**

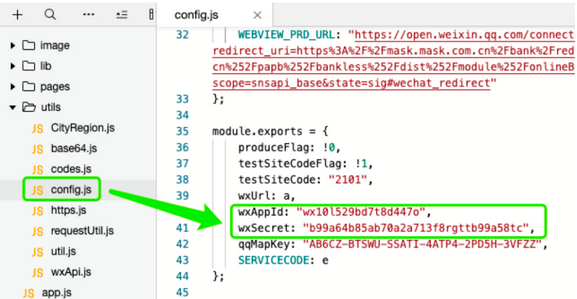
漏洞测试：

1、通过wxappUnpacker反编译工具进行分析，查看代码中是否存在AppSecret情况。

2、通过正则、关键字对代码、配置文件进行检索：检索常见的登录凭证，例如appid、secret、key等；

例如：

A. 对小程序进行反编译，再通过关键字检索，发现代码中的AppSecret。



B. 通过微信官⽅ jscode2session的API接口，判断sessionkey的方式确定此值是否为AppSecret



漏洞修复：

1. 禁止在代码中存储如口令和密钥之类的敏感数据，容易导致泄密，仅加密的密钥可以硬编码在代码中；
2. 开发者服务器可以根据⽤户标识来⽣成⾃定义登录态，⽤于后续业务逻辑中前后端交互时识别⽤户身份；
3. 会话密钥 session\_key 是对⽤户数据进⾏ 加密签名的密钥。为了应⽤⾃身的数据安全，开发者 服务器不应该把会话密钥下发到⼩程序，也不应该对外提供这个密钥。

### 请求直接返回SessionKey泄露

漏洞描述：

当开发者⾮常规范的使⽤第三⽅key来关联session\_key，并且⼀切操作查询均只使⽤第三⽅key，但也可能一些流程出现直接返回SessionKey的情况

漏洞场景：

1、在请求登录或获取openid时直接返回SessionKey，

2、在查询第三⽅key等功能中返回SessionKey

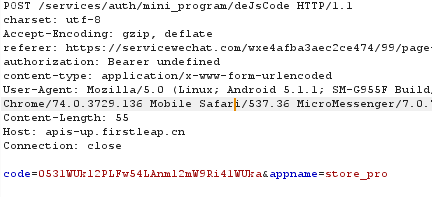
漏洞级别：

**【高危】**

漏洞测试：

例如：

A.正确的方式使用，返回第三⽅key



返回报文：

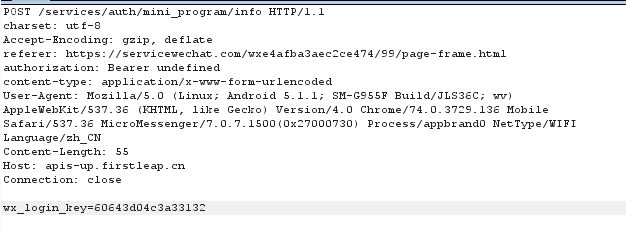


B. 在请求登录或获取openid时直接返回SessionKey

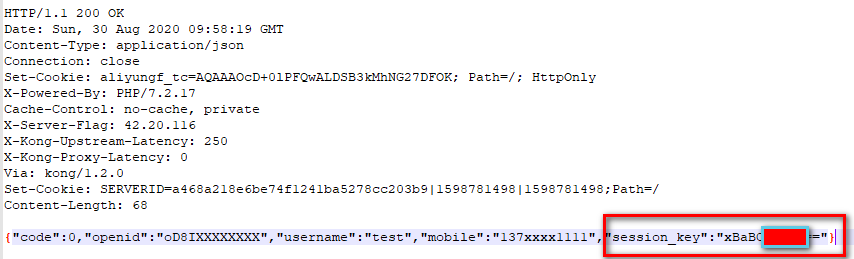
返回报文：



C. 在查询第三⽅key等功能中返回SessionKey



返回报文：



漏洞修复：

1）从微信这边获取到的session\_key不能直接返回给⽤户

2）如果需要返回⽤于校验⽤户身份，使用自己生成的第三⽅key，将session\_key和 自己生成的key在数据库中做个关联

3）禁止在一切查询提交等过程中通过第三⽅key返回session\_key

## Js\_code重复使用

漏洞描述：

当开发者⾮常规范的使⽤第三⽅key来关联session\_key，并且⼀切操作查询均只使⽤第三⽅key，但也可能一些流程出现直接返回SessionKey的情况

漏洞场景：

1、爆破SessionKey

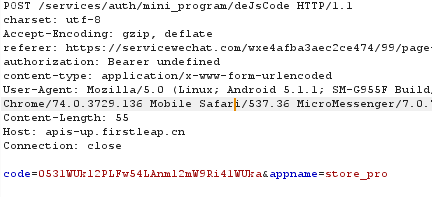
漏洞级别：

**【低危】**

漏洞测试：

例如：

A.抓包获取js\_code



B. 通过jscode2session的API接口猜解secret爆破SessionKey

[https://api.weixin.qq.com/sns/jscode2session?appid={0}&secret={1}&js\_code={2}&grant\_type=authorization\_code](https://api.weixin.qq.com/sns/jscode2session?appid=%7b0%7d&secret=%7b1%7d&js_code=%7b2%7d&grant_type=authorization_code)

返回报文：



漏洞修复：

1）每⼀次调⽤都要⽣成⼀个⼀次性有效的 js\_code 参数

2）⽣成的 js\_code 具有有效时间

3）同时间只能有⼀个 js\_code 有效（例如⼩程序内连续⽣成两 个code A和B，那么⽼的A便不再有效）

## CWRF漏洞

漏洞描述：

跨微信⼩程序请求伪造（Cross WechatAPP Request Forgery），利⽤⼩程序对⽤户微信身份的信任在获取⻚⾯传⼊的参数之后结合⽤户已经登录的身份信息（储存在本地的数据， 或者⽤户的openid）向⼩程序后端发送对应的数据包，从⽽使⽤户在⽆意间（打开对应分享链接时）完成⼀次请求操作。

起因：由于微信的这⼀分享机制可以使开发者掌控哪些⻚⾯可被分享，哪些⻚⾯不能被分享；并且微信⼩程序⻚与⻚直接传递参数的⽅法的确很简单并且⾮常的实⽤，所以在⽇常⽣活中被⼤量的使⽤。

漏洞场景：

小程序可以被分享的界面

前提：破解数据库，修改聊天记录转发

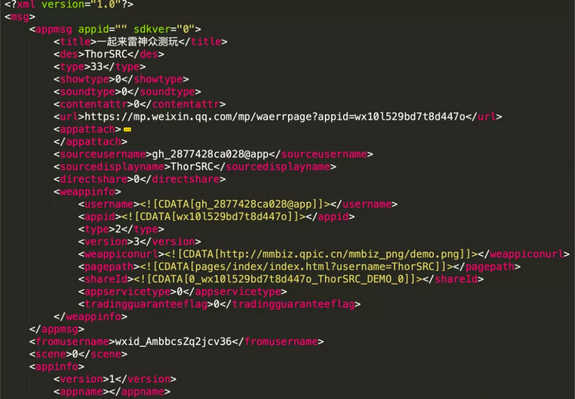
漏洞级别：

**【中危】**

**漏洞测试：**

1、修改微信⼩程序消息记录：

A.在成功进⼊数据库中，找到对应的对话，可以在 msgContent 字段中看⻅微信聊天记录中⼩程序的消 息，他是以XML的形式保存在数据库中的，例如下图：



B.标签说明

<title></title> 决定了⼩程序消息的标题内容；

<sourcedisplayname> </sourcedisplayname> 决定了⼩程序本身的名字； <pagepath></pagepath> 参 数，决定了⽤户点击⼩程序消息之后将携带什么参数前往⼩程序哪个指定的⻚⾯，例如上图示例中的⻚⾯路径及参数为：pages/index/index.html?username=ThorSRC 。

C.要修改 <pagepath></pagepath> 参数的内容就可以不受任何分享限制⾃定义⼩程序路径及传参 内容，这⾥需要注意⼀点：在⼩程序源码中⻚⾯路径为 pages/index/XXX ，⽽在运⾏环境下⼩程序⻚ ⾯后需要添加上“.html”变成 pages/index/XXX.html ，其余均⽆变化。

D.在修改完⼩程序消息的本地消息内容之后，我们只需要将对应的消息转发给他⼈即可

2、搜索⻚⾯间跳转：

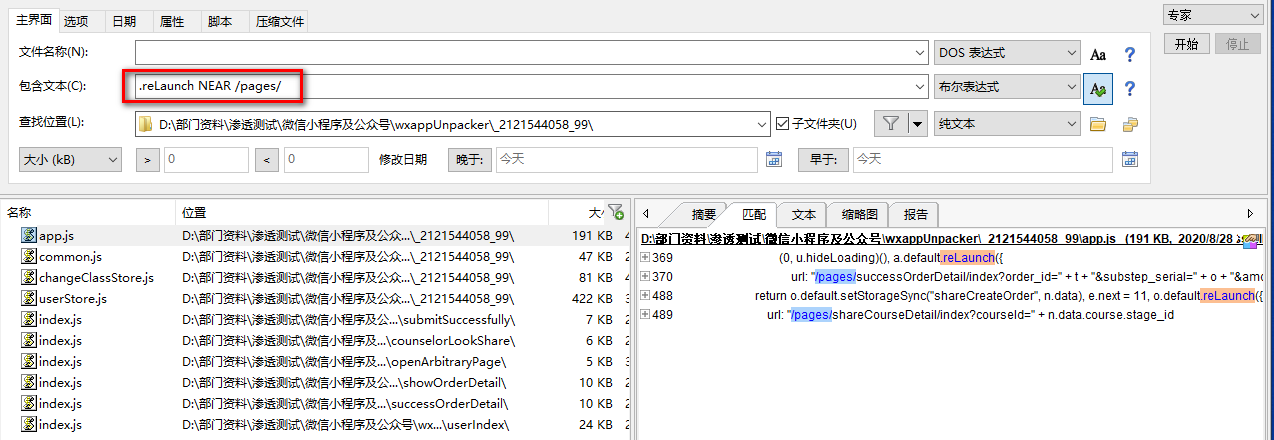
有CWRF漏洞存在的⼩程序⻚⾯，其⻚⾯必然有被有⻚⾯跳转功能的函数所引⽤过，那么只要在项⽬中 找到哪些代码⽚段引⽤了这些函数便可快速判定是否有CWRF漏洞的存在。

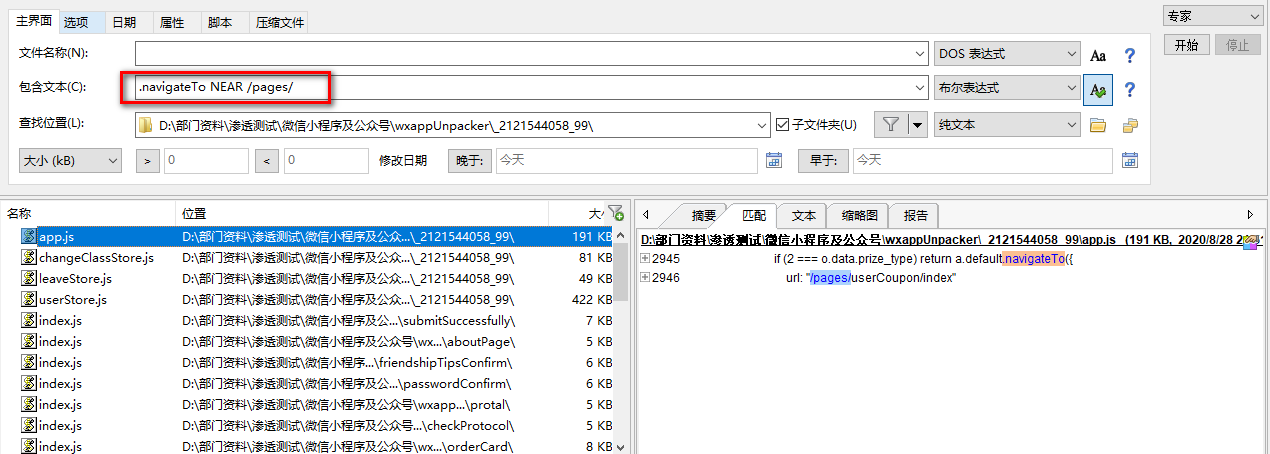
⽂件中有如下三 个可切换⻚⾯的函数：

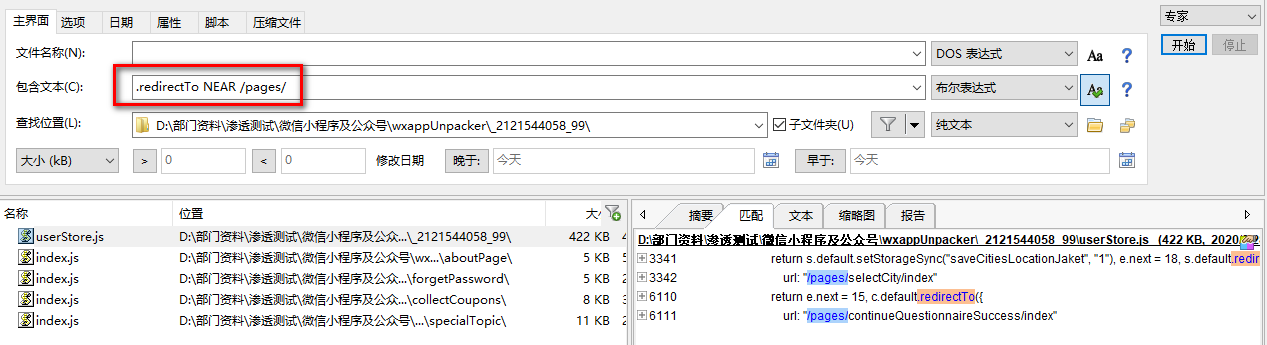
.reLaunch -- 关闭所有⻚⾯，打开到应⽤内的某个⻚⾯

.navigateTo -- 保留当前⻚⾯，跳转到应⽤内的某个⻚⾯

.redirectTo -- 关闭当前⻚⾯，跳转到应⽤内的某个⻚⾯







**漏洞修复：**

1）重点功能点使用动态验证码

2）通过token方式防护

参考链接：https://mp.weixin.qq.com/s?\_\_biz=MzI0NzEwOTM0MA==&mid=2652476494&idx=1&sn=75f1d8184c3764b2a1900202cb8d512c&chksm=f25837fdc52fbeebfa3609237f861070f181675ba3775a2bcab9b04585a9701514a2b81aa676&scene=178#rd



## 供应链漏洞



### 第三方组件问题

漏洞描述：

微信⼩程序本身也是⽀持插件功能的，⼀旦这些插件出现了漏洞也会对⼩程序本身的安全造成⼀定的影响。

漏洞场景：使用第三方外部插件

漏洞级别：

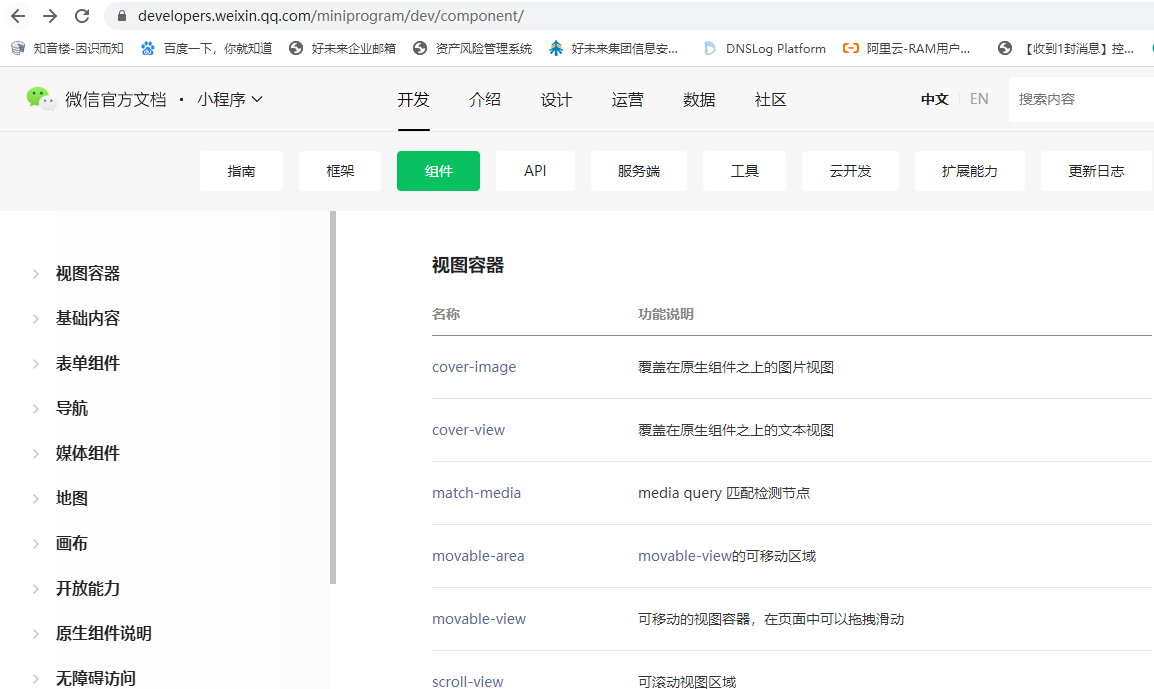
**【中危】**

**漏洞测试：**

1、在⼩程序详情⻚⾯的“服务⽀持”中查看到当前⼩程序使⽤的全部插件名称和他对应的服务商。

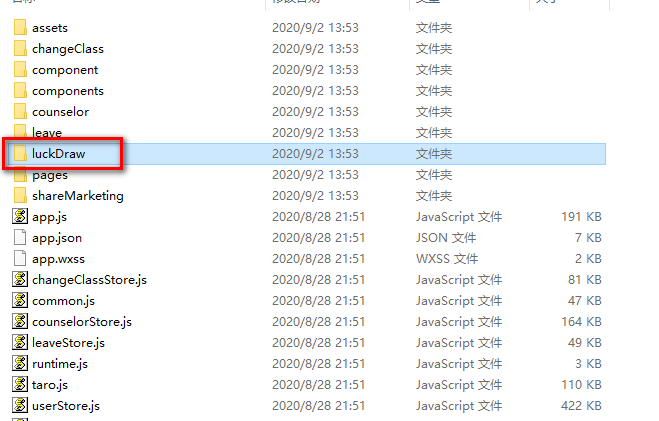
2、官方组件查询地址：

<https://developers.weixin.qq.com/miniprogram/dev/component/>



3、查看反编译后的文件夹及文件

例如：使用第三方的插件库，搜索是否含有漏洞



漏洞修复：

1. 使用微信官方提供的插件库
2. 及时更新已修复的第三方插件版本
3. 插件已停止维护后更换替代插件，删除已有插件

### 第三方授权问题

漏洞描述：

微信提供了⼩程序账户第三⽅授权功能功能。通过此功能将对于的⼩程序账户授权给第三⽅平台使⽤，⼀旦授权第三⽅平台等于接管了你的账户可以执⾏任何操作（视实际给予权限⽽定）。

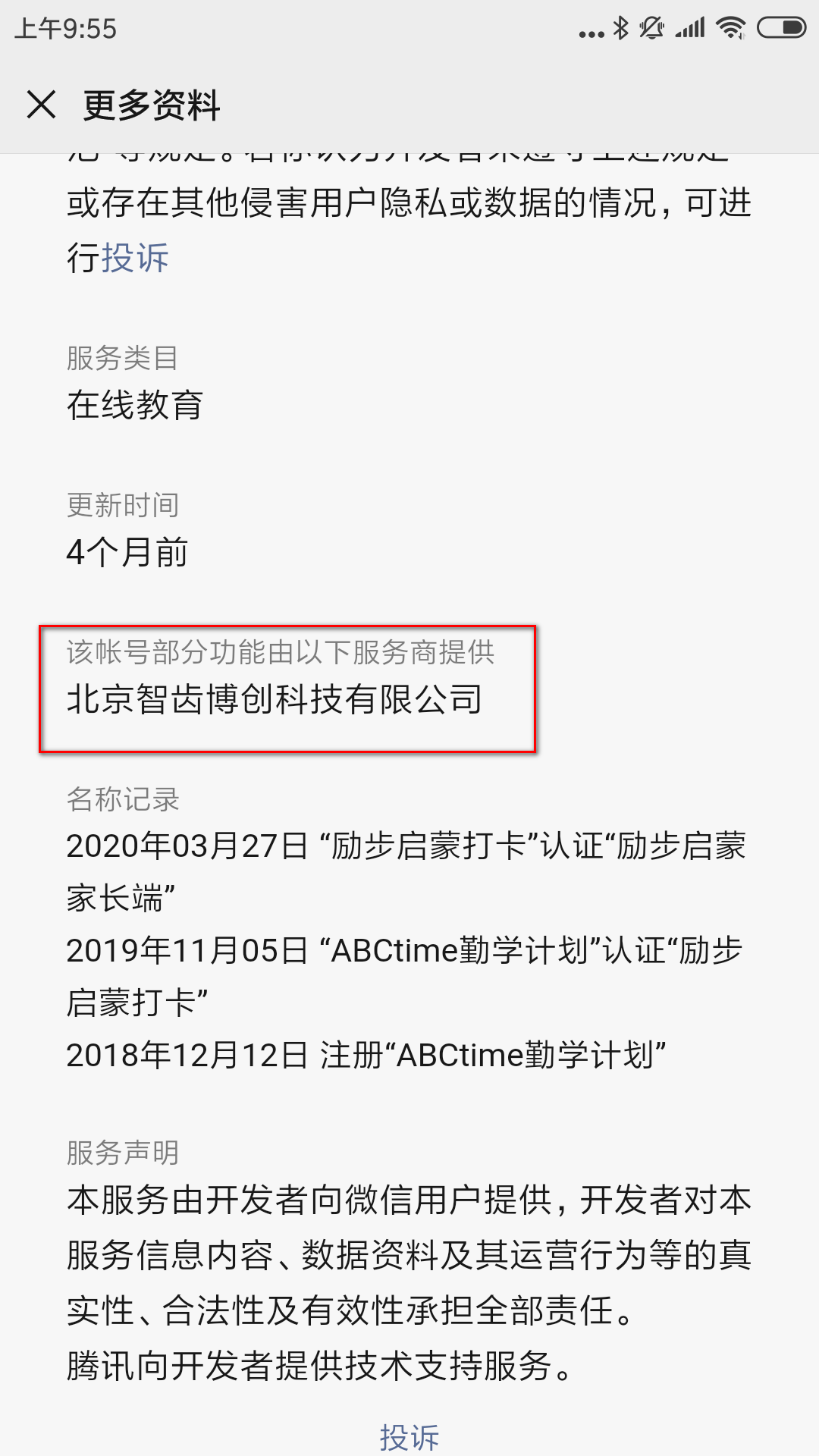
漏洞场景：授权给第三方平台

漏洞级别：

**【高危】**

漏洞测试：

可以在⼩程 序详情⻚⾯的“该账号的部分功能由以下服务商提供”中查看到当前⼩程序账户被授权平台对应的公司名称，通过搜索公司名称，可以很快的找到其对应被授权的平台。那么如果这个平台存在某个越权漏洞，便可以轻松接管被授权在此平台上的所有微信⼩程序。



漏洞修复：

1、做好严格的访问控制及授权管理。

## WEB相关漏洞



### 需要关注的

描述：

各类逻辑漏洞、水平越权、SQL注入、任意文件上传、SSRF、通用组件漏洞、命令执行、反序列化

### 不需要关注的

描述：

XSS（分享朋友圈方式可以尝试）、CSRF、JSON劫持、CORS劫持、URL跳转、CRLF注⼊、目录遍历。

**原因：**

1、微信⼩程序的特性是不能执⾏动态脚本；

2、⼩程序内⽤户的⼀切操作 都是在微信沙盒内完成的，攻击者⽆法通过浏览器利⽤⽤户在⼩程序内的身份凭据来完成攻击；

3、后端API的特性（⼤多以JSON格式返回数据）

# 附件

