安卓逆向-反编译-修改-打包-签名-解决夜神模拟器usb调试找不到安卓apk的问题，无法进行动态调试的解决方案。

原创

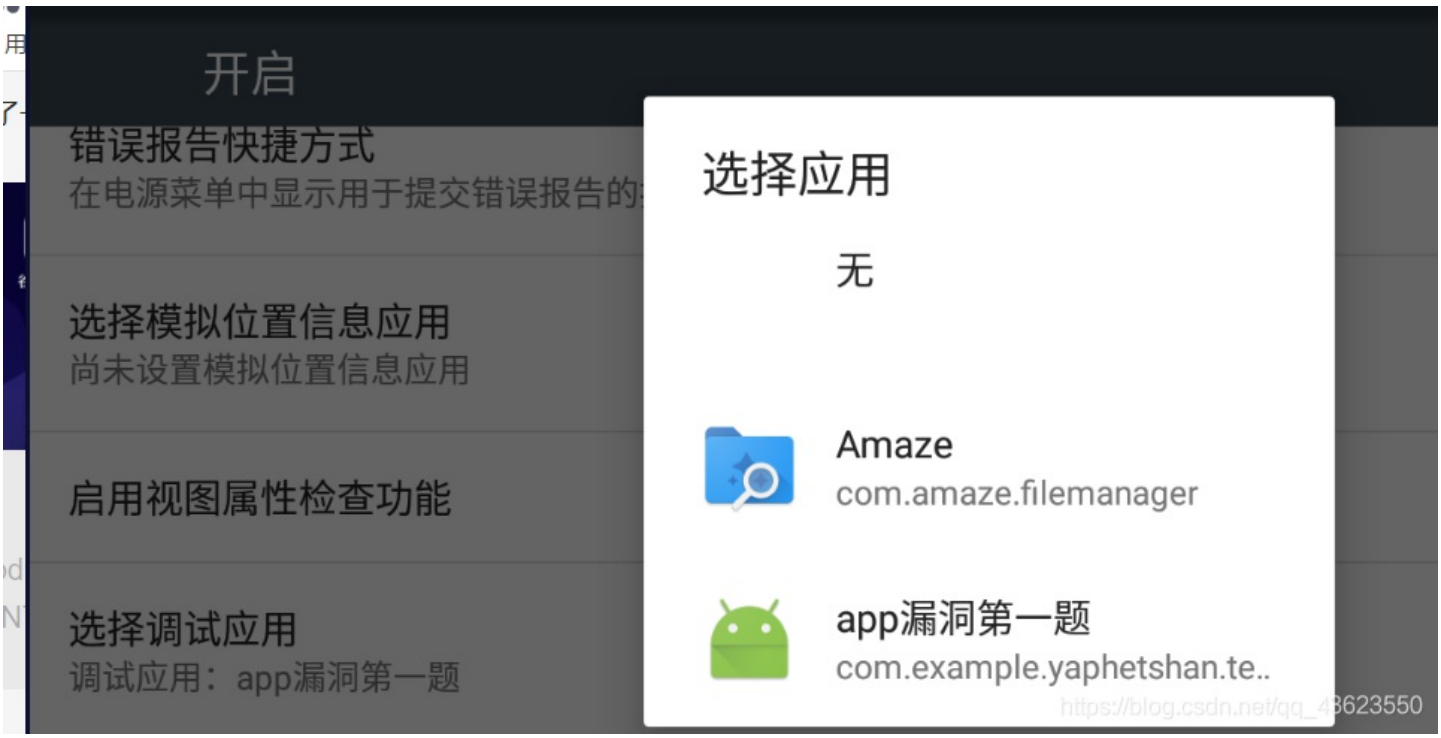
啦啦啦三杀了 于 2021-03-09 14:31:57 发布 356 收藏 2

文章标签： 反编译 安卓 android apk debug

本文链接：https://blog.csdn.net/qq_43623550/article/details/114579081

问题来由，在进行了一定时间的安卓逆向的学习，遇到了一个问题如图：

这么多个安卓apk，打开开发者模式，



调试应用只有一个，这不是离谱，按照之前的文章，这就不能进行动态调试。既然有了问题就要想解决办法，问了带我的师傅：我不是用你这个方法的，给我演示了一下他的手法（cmd一顿操作，我没看懂）。没办法只能自己想办法，在writeup中看到有人提到了这个问题，原来是说APK没开启动态调试，就是xml中没开启调试功能，所以检测不到。（嘿，别问我怎么get到这个点的）

找到了原因解析来就是如何给他添加上这个功能。总所周知，安卓开发现在为了安全如果逆向后修改他的源文件需要加上其对应

找到了原因解析来就是如何给他添加上这个功能，总所周知，安卓开发现在为了安全如果逆向后修改他的源文件需要加上其对应的签名（部分开发者防护做的很差，所以经常会被人加入恶意代码后发布到应用商城，以此进行传播在top10中叫做：代码篡改）

好了接下来就是实战了：

首先对于apk文件进行反编译，这里使用apktool（没有的自行百度）。

```
C:\Users\jjh\Desktop\ctf\android反编译三件套>java -jar apktool_2.3.4.jar d -f f6adc401d0eb472892a4ac4481f76a85.apk -o go
ngfangshijiecrackme
I: Using Apktool 2.3.4 on f6adc401d0eb472892a4ac4481f76a85.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (C:\Users\jjh\AppData\Local\apktool\framework), using C:\Users\jjh\AppData\Local\Temp\ in
stead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the defa
ult storage directory is unavailable
I: Loading resource table from file: C:\Users\jjh\AppData\Local\Temp\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```


dex2jar-2.0	2021/3/3 17:20	文件夹	
gongfangshijiecrackme	2021/3/9 11:44	文件夹	
apktool_2.3.4.jar	2019/2/12 20:21	Executable Jar File	10,746 KB
apktool使用方法.txt	2021/3/4 9:33	文本文档	1 KB
f6adc401d0eb472892a4ac4481f76a8...	2021/3/8 17:10	APK 文件	1,048 KB
jd-gui.exe	2015/8/8 15:54	应用程序	8,689 KB
看下几个工具的区别.txt	2021/3/3 17:22	文本文档	1 KB

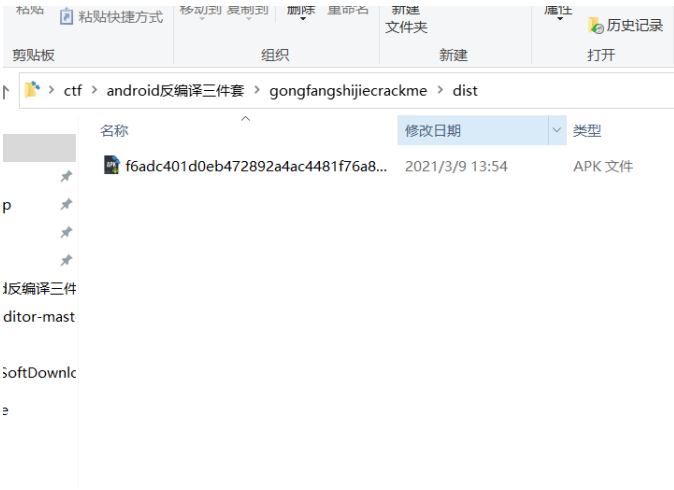
再往下需要修改AndroidManifest.xml为其添加上debug=true的选项

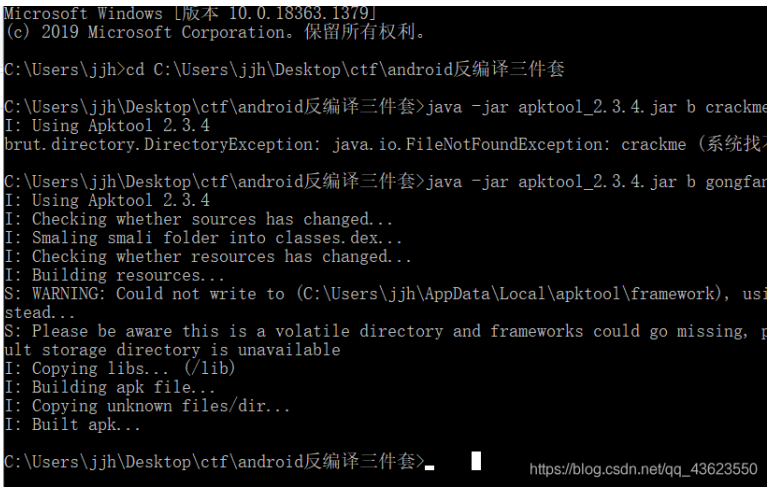
有两种操作

1.直接解压apk得到AndrodiManifest.xml 这时候打开会发现是乱码，接下来要使用AXML工具来添加 android:debuggable="true"

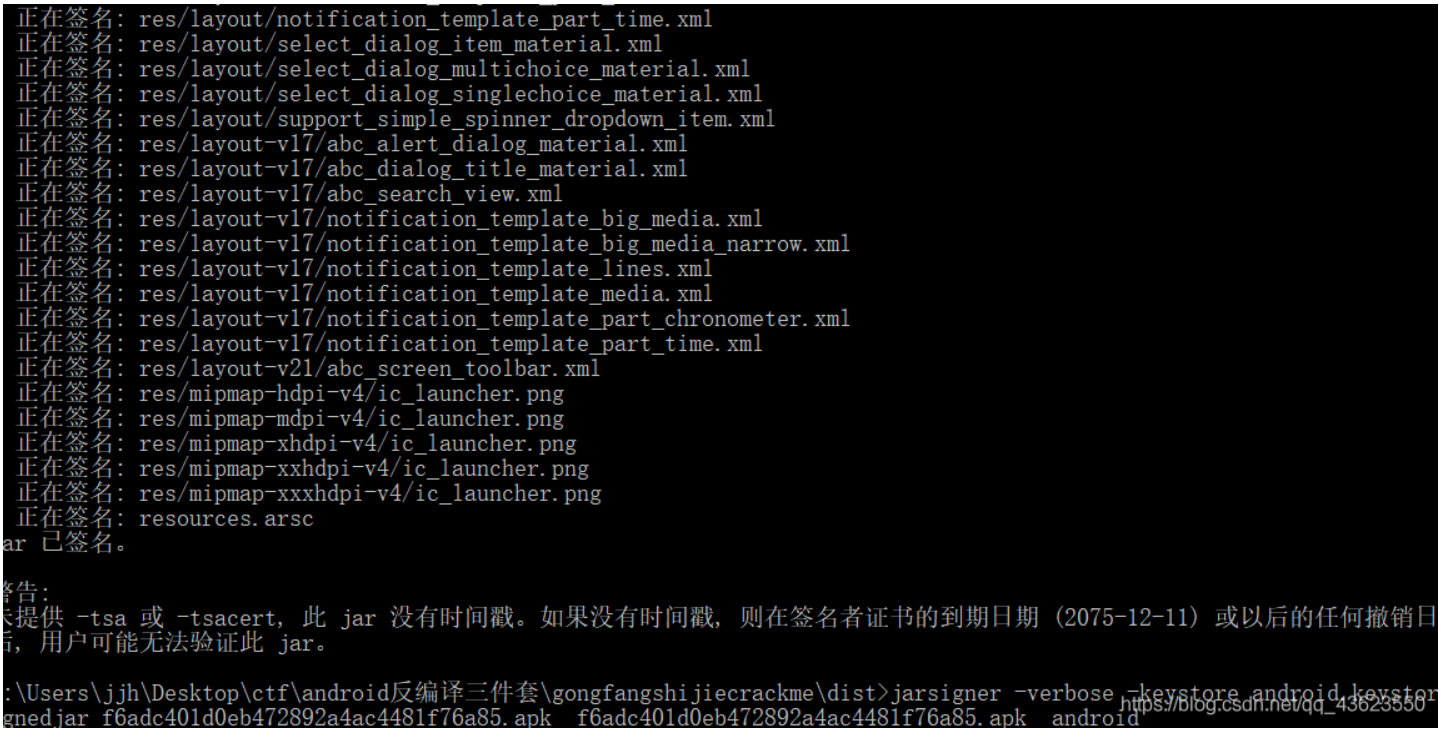
2.用apktool反编译出来的 AndrodiManifest.xml ，发现是正常的，直接添加即可。

现在已经改好了需要的debug=true了，接下来就是打包
使用apktool进行打包，cmd执行如下代码
java -jar apktool_2.3.4.jar b gongfangshijiecrackme



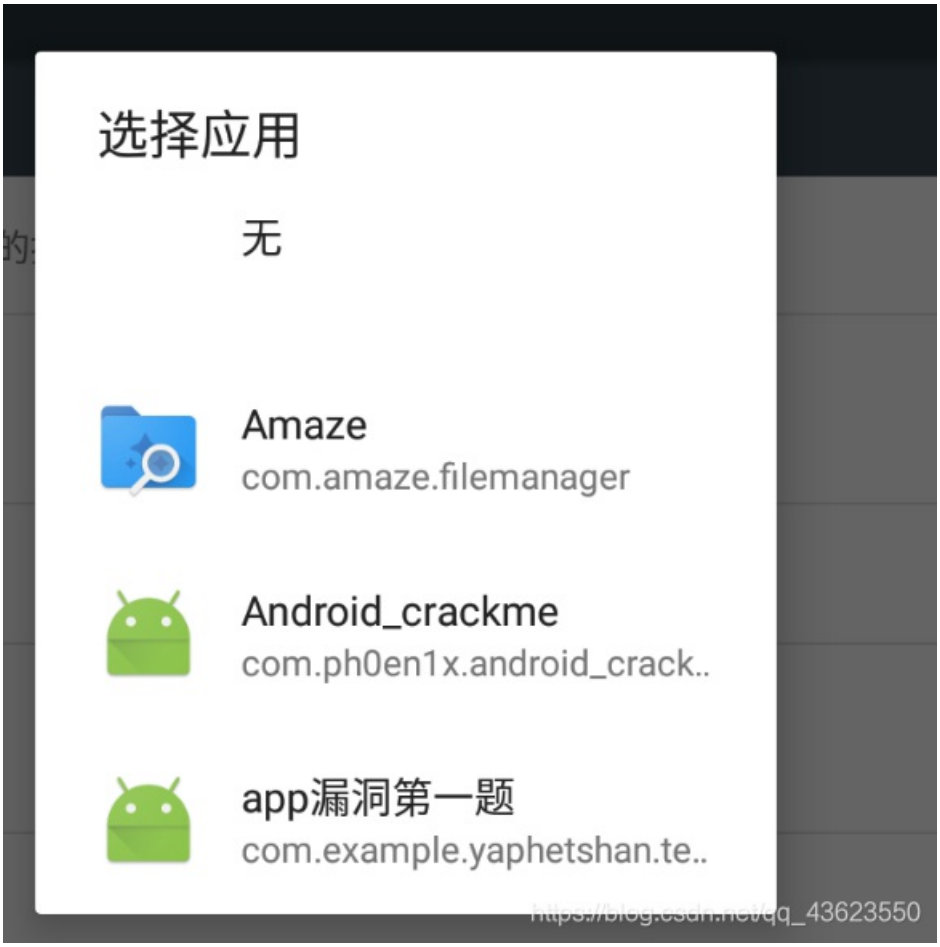


接下来没有签名，让我们来看看反编译的效果，可以根本不能够打开跑在模拟器上面，接下里就是写签名。
这里使用jdk自带的keytool 根据网上的教程生成自己的android.ketstore ，然后把要签名的和android.keystore放在一起执行如下命令
jarsigner -verbose -keystore android.keystore -signedjar f6adc401d0eb472892a4ac4481f76a85.apk f6adc401d0eb472892a4ac4481f76a85.apk android



这个时候签名完成了。

打开模拟器看看



emmm，针不错

签名操作很繁琐可以观看一下网址
https://blog.csdn.net/akmzpzn69761/article/details/101638539

https://blog.csdn.net/rrkdddd33/article/details/80067589?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242