

关于android:sharedUserId= "android.uid.system" 的使用

原创VIP文章MYBOYER已于 2022-03-11 19:35:52 修改20417 27收藏版权

分类专栏: android进阶 文章标签: android eclipse java

有时候我们需要在自己的应用中使用一些系统相关权限，比如USB权限，如果我们自己的应用所在进程和系统进程为同一个UID，就会默认拥有该权限，不需要用户自己去授予，很多情况下会方便很多。最近在做的项目，有好大一部分都用到这个权限，修改系统时间啊，调用隐藏方法啊，系统关机重启啊，静默安装升级卸载应用等等，刚开始的时候，直接添加权限，运行就报错，无论模拟器还是真机，在logcat中总会得到"Unable to open alarm driver: Permission denied"。这个函数需要root权限或者运行于系统进程中才可以用。

网上搜索了好久，发现有两种方法可以解决：

一种是需要在Android系统源码的环境下用make来编译：

1. 在应用程序的AndroidManifest.xml中的manifest节点中加入android:sharedUserId="android.uid.system"这个属性。
2. 修改Android.mk文件，加入LOCAL_CERTIFICATE := platform这一行
3. 使用mm命令来编译，生成的apk就有修改系统时间的权限了。

不会.mk编译这种方法，所以就只好参考第二种了：

1. 加入android:sharedUserId="android.uid.system"这个属性。
2. 使用eclipse编译出未加签名的apk文件，但是这个apk文件是不能用的。
3. 使用目标系统的platform密钥来重新给apk文件签名。

这步比较麻烦，首先找到密钥文件，在我的Android源码目录中的位置是"build/target/product/security"，

下面的platform.pk8和platform.x509.pem两个文件。然后用Android提供的Signapk工具来签名，

signapk的源代码是在"build/tools/signapk"下，用法为"signapk platform.x509.pem platform.pk8 input.apk output.apk"，

文件名最好使用绝对路径防止找不到，也可以修改源代码直接使用。这样最后得到的apk和第一个方法是一样的。

最后解释一下原理，首先加入android:sharedUserId="android.uid.system"这个属性。

通过Shared User id,拥有同一个User id的多个APK可以配置成运行在同一个进程中。

那么把程序的UID配成android.uid.system，也就是要让程序运行在系统进程中，这样就有权限来修改系统时间了。

只是加入UID还不够，如果这时候安装APK的话发现无法安装，提示签名不符，

原因是程序想要运行在系统进程中还要有目标系统的platform key，就是上面第二个方法提到的platform.pk8和platform.x509.pem两个文件。

用这两个key签名后apk才真正可以放入系统进程中。第一个方法中加入LOCAL_CERTIFICATE := platform其实就是用这两个key来签名。

这也有一个问题，就是这样生成的程序只有在原始的Android系统或者是自己编译的系统中才可以用，因为这样的系统才可以拿到platform.pk8和platform.x509.pem两个文件。



MYBOYER

关注



要是别家公司做的Android上连安装都安装不了。试试原始的Android中的key来签名，程序在模拟器上运行OK，不过放到G3上安装直接提示"Package ... has no signatures that match those in shared user android.uid.system"，这样也是保护了系统的安全。

最后还说过，这个android:sharedUserId属性不只可以把apk放到系统进程中，也可以配置多个APK运行在一个进程中，这样可以

共享数据，应该会很有用的。

什么是UID

众所周知，Pid是进程ID，Uid是用户ID，只是Android和计算机不一样，计算机每个用户都具有一个Uid，哪个用户start的程序，这个程序的Uid就是那个那个用户，而Android中每个程序都有一个Uid，默认情况下，Android会给每个程序分配一个普通级别互不相同的Uid，如果用互相调用，只能是Uid相同才行，这就使得共享数据具有了一定安全性，每个软件之间是不能随意获得数据的。

设置自己的应用UID为系统进程的UID

系统进程UID有三种：

- android:sharedUserId="android.uid.system"
 - android:sharedUserId="android.uid.shared"
 - android:sharedUserId="android.media"
- 我们这里取第一种来实验下，其他两种实现的方式都一样。

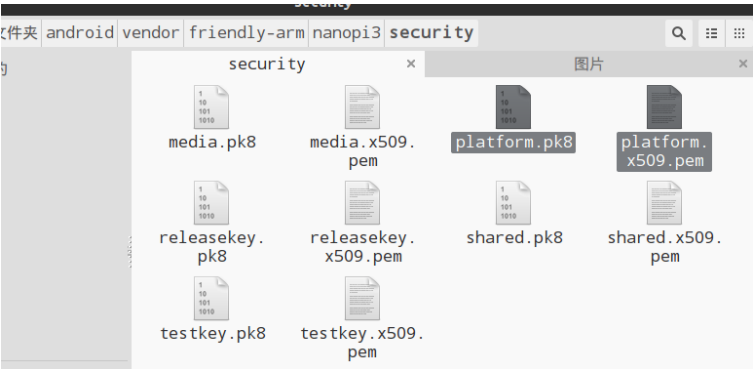
声明UID

首先我们需要在manifest中声明该应用的UID：

```
1 | <?xml version="1.0" encoding="utf-8"?>
2 |
3 | <manifest xmlns:android="http://schemas.android.com/apk/res/android"
4 |     package="cn.izis.chessdeskrobot" 4
5 |     android:sharedUserId="android.uid.system" 5
6 |     ...
7 | </manifest>
```

签名打包

签名打包的前提是要有系统的签名文件，一般位于源码的build/target/product/security目录，有的厂商定制的系统会放在其他目录，比如我用的Nanopc3在如下目录，具体可以全局搜索一下，一开始我也使用的是build目录下的，结果打包后怎么也装不上，后来发现vendor目录下也有一个/(T o T)/~~~：



**MYBOYER**

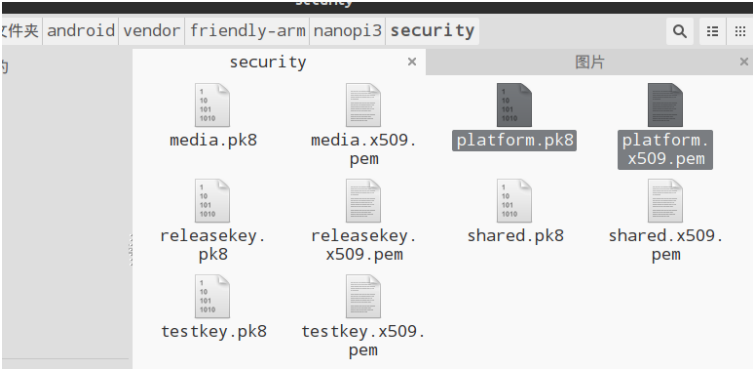
关注



正在上传... 重新上传

取消 正在上传... 重新上传

取消



签名文件.png

其中android:sharedUserId="android.uid.system"对应platform.pk8、platform.x509.pem两个文件。

下面说一下导入系统签名常用的三种方式：

- 对已经生成的apk重签名
- 对应用自身的签名文件重签名
- 通过mk文件编译

已有apk

对于已经生成的apk文件可以通过源码/out/host/linux-x86/framework目录下的signapk.jar文件通过以下命令进行重新签名。

```
java -jar signapk.jar platform.x509.pem platform.pk8 my.apk new.apk
```

注意使用该指令的话signapk.jar、platform.x509.pem、platform.pk8、my.apk四个文件要在同一路径，否则需要自己更改路径。

重新生成签名

生成自己的key文件后利用keytool-importkeypair工具使用如下命令重新生成签名：

```
keytool-importkeypair -k demo.jks -p 123456 -pk8 platform.pk8 -cert platform.x509.pem -alias demo
```

- demo.jks : 签名文件
- 123456 : 签名文件密码
- platform.pk8、platform.x509.pem : 系统签名文件
- demo : 签名文件别名


keytool-importkeypair工具可以从[这里](#)下载。

利用mk文件在源码环境下编译

这种方式我暂时没有测试，因为对mk文件不是很了解。一般情况下也不会用到，毕竟现在基本都是使用ide工具来开发的，上面两种方式足够使用了。如果有需要，以后再添加。

文章知识点与官方知识档案匹配，可进一步学习相关知识



 android:sharedUserId="android.uid.system" 的... m0_38094058的博客 1万+


系统权限使用 android:sharedUserId="android.uid.system" 通过Shared User id,拥有同一个User id的多个APK可以配置成运行在同一个进程中。那么把程序的UID配成android.uid.sy...

 Android中sharedUserId的用处 热门推荐 嵌入式软件开发交流 2万+


不同APK之间如何互相访问数据? Android为我们提供了两种数据互访的方法: (1) 使用Share Preference(设置为共有)或Content Provider APK通过指定接口和数据供其它APK读取...

评论 4条 > 写评论

 qq_20523479 热评 开通文章 解锁VIP

 android:sharedUserId作用 最新发布 Jason_Lee155的博客 3527

前言 Android给每个APK进程分配一个单独的空间, manifest中的userid就是对应一个分配的Linux用户ID, 并且为它创建一个沙箱, 以防止影响其他应用程序 (或者被其他应用程序...

 Android中sharedUserId的用处_深海白玉的博客_android... 1-1

android:sharedUserId作用 Jason_Lee155的博客 3510 前言Android给每个APK进程分配一个单独的空间,manifest中的userid就是对应一个分配的Linux用户ID,并且为它创建一个沙...

 ...sharedUserId_L-Lawliet的博客_android:shareduserid 1-3

通过Shared User id,拥有同一个User id的多个APK可以配置成运行在同一个进程中.所以默认就是可以互相访问任意数据. 也可以配置成运行成不同的进程, 同时可以访问其他APK的...

“相关推荐”对你有帮助么?

-  非常没帮助
-  没帮助
-  一般
-  有帮助
-  非常有帮助

©2022 CSDN 皮肤主题: 大白 设计师: CSDN官方博客 返回首页

关于我 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2023北京创新乐知网络技术有限公司

 MYBOYER
码龄10年 暂无认证

265 原创

2万+ 周排名

3058 总排名

91万+ 访问

 等级

1万+ 积分

232 粉丝

92 获赞

82 评论

359 收藏









私信

关注

搜博文文章 

热门文章

- 【无人机开发】通讯协议MavLink详解 

29287
- win10下Miracast无线投屏使用教程及异常解决方案(超详细) 

21577
- 关于 android:sharedUserId="android.uid.system" 的使用 

20404
- hellocharts-android开源图表库 (效果非常好) 

13226
- Spinner 取消默认选中 

11092

 MYBOYER

关注



 人工智能

1篇

 JetPack实践

6篇

 HarmonyOS

 Android Kotlin实践

5篇

 Kotlin实践

 Java

36篇

▼

最新评论

关于android:sharedUserId="android.uid....
qq_20523479: 开通文章 解锁VIP

关于android:sharedUserId="android.uid....
qq_20523479: 开通VIP 解锁文章

Jetpack之使用 Jetpack Compose 更快地...
golang人工智能框架: 我知道，不是不理
解，懂意思就行了，就是原来的xml写法 ...

Jetpack之使用 Jetpack Compose 更快地...
MYBOYER: 好好理解一下，这不是语言之
间的比对

阿里淘系自研标准化协议库 XQUIC
macWhale: OK

您愿意向朋友推荐“博客详情页”吗？











强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

AI之神经网络

Android多进程:生成多进程(android:process
属性)

Android 13 现已正式发布，看看有哪些更
新!

2022年 5篇

2021年 9篇

2020年 25篇

2019年 20篇

2018年 18篇

2017年 28篇

2016年 17篇

2015年 70篇

2014年 140篇

2013年 431篇

目录

什么是uid

设置自己的应用uid为系统进程的uid

声明uid

签名打包

已有apk

重新生成签名

利用mk文件在源码环境下编译

 MYBOYER

关注



https://blog.csdn.net/MYBOYER/article/details/104902159?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-... 6/6