**免ROOT框架TweakMe介绍**

TweakMe是一个轻量级的逆向开发框架，它与传统的xposed、frida、magisk等框架相比的最大优势是无需root环境，对手机系统的要求很低，只需要一个正常手机即可对app进行逆向分析。简单的几步操作就可以完成对指定app的进程注入，可以绕过大多数加固(360、梆梆、爱加密等)的签名证书校验。完成注入后通过Java编写hook代码，一键脚本编译为dex插件。目前TweakMe框架在5.0到11.0的android手机上测试通过。由于篇幅问题，本文档并不介绍TweakMe的实现原理，仅仅介绍如何快速使用它，以及使用过程中的注意事项。

TweakMe地址：**<https://git.17usoft.com/TweakMe/TweakMe>**

一、使用前准备：

1. 一台没有ROOT过的正常Android手机，系统版本11及以下都可。
2. 开发电脑需要安装AndroidStudio，Java版本最好是1.7或1.8。
3. 一个待测试的apk【demo.apk】，假设包名为【com.example.tweakme】。

二、将TweakMe拉取到本地，目录结构类似如下图



Apktweak目中附带了一个apktweak.exe的命令行工具，后续步骤会经常用到。

1. 重打包apk

**重打包的目的是让APP在启动时主动加载我们的TweakMe/libs/armeabi/libsotweak.so。**

所以我们要在app中放入我们自己的so，并微调某个so，让它把我们的so加载起来。

重打包命令：**apktweak --apk xxx.apk --target xxx.so**

--target参数指定app中待微调的so，不同app设置可能会不一样，但总体原则是：

**指定为app在启动时第一个加载的应用层so。**

目前大多数的app都有加壳，所以这里的--target其实就是壳的so。

对于梆梆加固是libDexHelper.so，对于爱加密是libexec.so，对于360加固是libjiagu.so。

需要注意的是，对于爱加密、360加固等壳的so并不在通常的lib/abi目录下，而在assets目录中，所以--target 参数必须要指定绝对路径，只有在通常的lib/abi目录下时才可以只写so名称。可供参考的命令如下：

**apktweak --apk xxx.apk --target libDexHelper.so**

**apktweak --apk xxx.apk --target assets/ijm\_lib/armeabi/libexec.so**

**apktweak --apk xxx.apk --target assets/libjiagu.so**

如果打包成功，则会在apk当前目录中生成一个重签名的新apk包，如下图所示

apk名称带sign的为重签名apk，另一个为原始apk

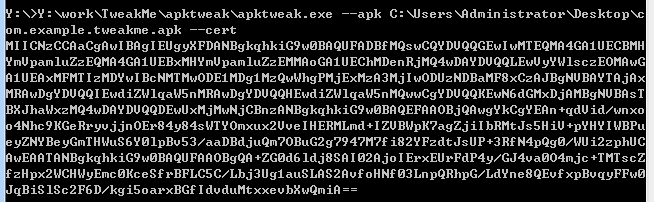


1. 获取原apk证书签名数据

重签名的包，最大的改变是签名证书和原来不一样了，加固壳在启动过程中肯定会检查apk包是否被篡改，那时签名证书校验就会报错直接闪退。所以，我们需要获取到原apk证书签名数据，在dex插件中适当时机再重新设置回去。

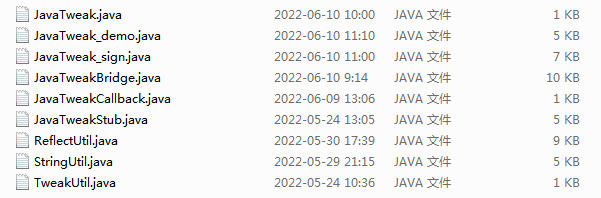
打印apk签名证书命令：**apktweak --apk xxx.apk --cert**

命令执行后会在控制台打印出签名证书数据（Base64编码），如下图



1. 编译插件

TweakMe/src目录是插件源码目录，大家可自行开发JavaTweak\_xxx.java插件类。源码中附带有一个demo插件和一个过证书签名校验的sign插件。



在编译插件之前，必须要设置正确demo插件中的apk签名证书数据。

调用如下方法设置：

JavaTweak\_sign.*setAppCert*



自行开发的tweak插件，也必须要有这一步。

设置好正确的值后就可以编译打包插件了。

编译dex插件命令：**java2dex.bat**

直接在控制台中执行上述bat脚本即可自动将java代码打包为dex插件。编译成功后会在脚本目录生成一个javatweak.dex。如下图



六、安装apk和插件

apk的安装不再赘述，直接用adb install 命令即可，只要注意这里安装的是重签名后的apk。

下面着重介绍下插件的安装。

由于没有root权限，我们唯一能读写的目录是/sdcard

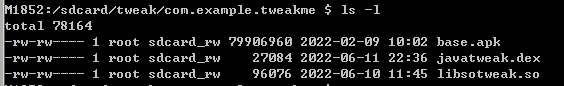
1. 在/sdcard目录下新建tweak目录（目录名称固定）
2. 在tweak目录下新建包名目录（不同的app包名不同，目录名称也不同），我这里就是

com.example.tweakme目录。

1. 将javatweak.dex、libsotweak.so、原始demo.apk通过adb push到刚创建的

/sdcard/tweak/com.example.tweakme目录下，如下图所示：

**注意所有的原始apk名称都必须重命名为base.apk**

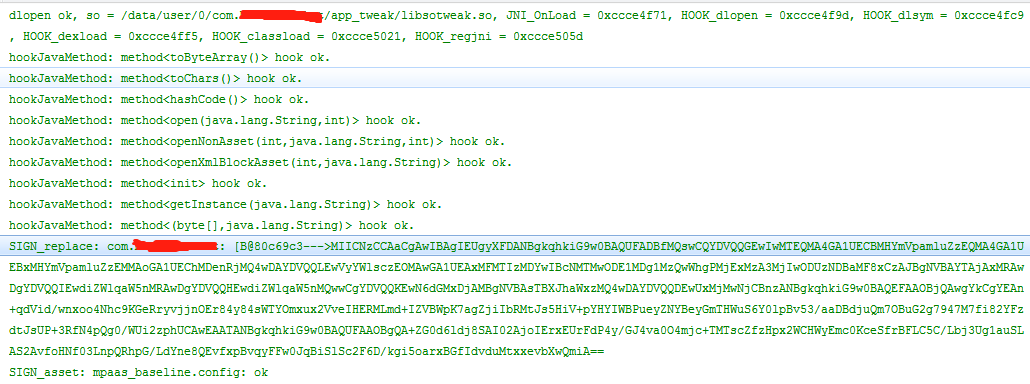


七、设置app权限，允许读写/sdcard目录

安装的app默认是没有读写/sdcard目录权限的，我们在运行app之前要为它设置权限。如果没有设置权限就直接运行app会报如下图所示的错误，然后app会闪退。



如果读写权限设置OK，会有如下类似日志打印



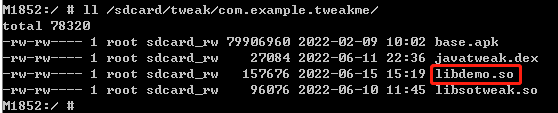
八、功能扩展：如何在java层调用自己编写的so

如果只开发dex插件的话，上面7步已经足够了。如果还想在java层调用自己编写的so，那么可以通过如下步骤完成（假设你已经写好了一个名为libdemo.so的动态库）。

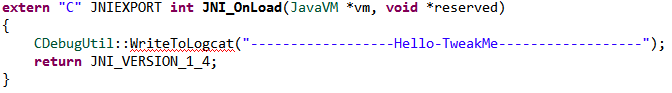
1. 在java层加载so，如下图



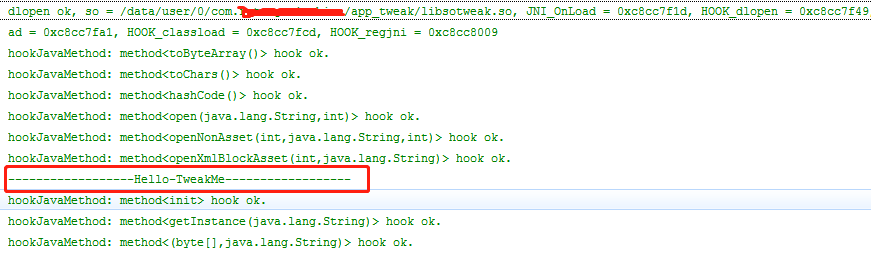
1. 将so放入libsotweak.so所在的目录中，如下图



1. 我的libdemo.so在JNI\_OnLoad中只打印了一条简单的日志



日志输出如下



1. 重签名apk

如果想手工调整apk包的内容，调整完成后，可以用如下命令重签名一下

Apk重签名命令：**apktweak --apk xxx.apk --sign**