**免ROOT框架TweakMe介绍**

TweakMe是一个轻量级的逆向开发框架，目前已经升级到4.0版本，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12.0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 QQ群：**325509091**

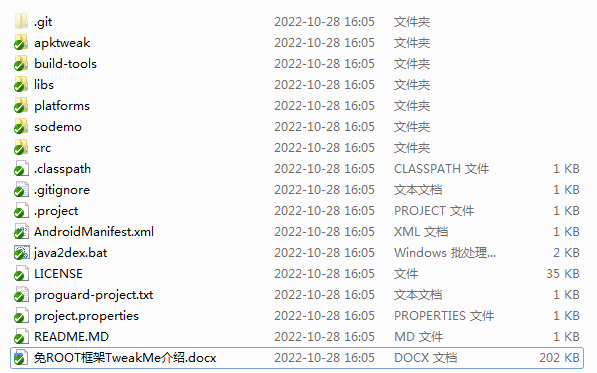
TweakMe 地址：**<https://github.com/liaoguobao/TweakMe>**

强烈建议在非ROOT的正常手机上使用本框架。

一、使用前准备：

1. 一台没有ROOT过的正常Android手机，系统版本12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。

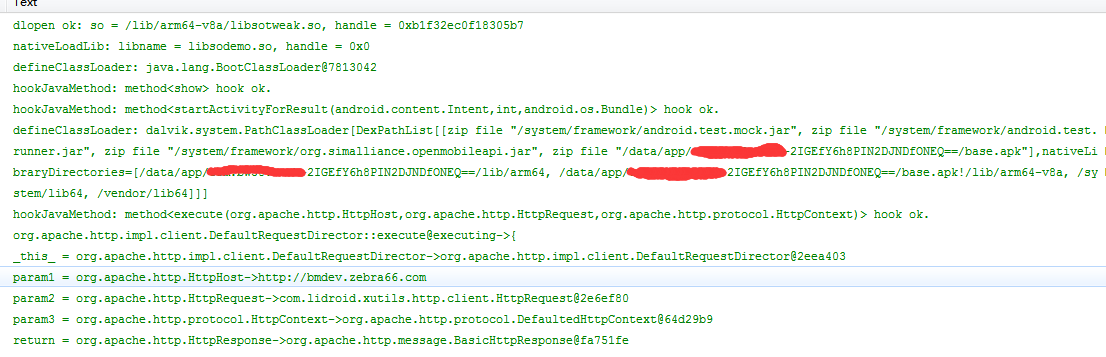
一键内嵌打包命令（梆梆、爱加密、360）：

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

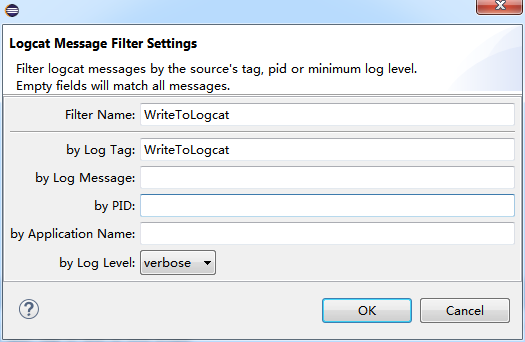
**apktweak.exe --embed --apk xxx.apk --target assets/ijm\_lib/armeabi/libexec.so**

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

如果一切正常的话，会在apk所在目录生成一个全新的xxx-embed.apk。如果你是TweakMe的首次使用者，或者想一睹最新4.0框架的内嵌打包功能在非root手机上、在加固的app中的使用效果。那么，现在你可以在你的手机上直接安装xxx-embed.apk了，并且可以直接运行（希望如此），在logcat中会有类似如下的日志输出（demo插件的输出日志）。



为了更方便的查看TewakMe框架日志，记得加上如下的日志过滤器。



至此，如果你对本框架产生了兴趣，想要进一步系统的了解框架的高级使用技巧，开发出自己的实用插件，甚至希望在自己的项目中使用它，那么请继续往下看。

如果你的app很不幸运的没有运行起来而是闪退了，那么希望你提交你的样本给我，我会在后续的升级中尝试修复它（不保证一定能修复哦）！

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/$ABI/libsotweak.so。**

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

--target参数指定app中的目标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。**

**--embed参数为可选参数，如果设置此参数，代表将整个TweakMe框架内嵌到apk中。**

**重打包后的apk可以直接安装到手机，无需授予任何权限，直接运行app后即可看到默认生效的demo插件打印出的hook日志。如果不带此参数，则是框架外置模式。**

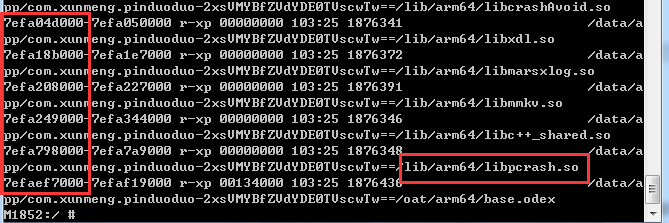
**内嵌模式下会自动编译javatweak.dex，而外置模式需要手动编译。**

**内嵌和外置两种模式各有优势。一般来说，外置方式文件更新更为方便灵活，文件更新后无需再次打包和安装，app重启即生效，适合插件开发阶段使用。而内嵌模式则依赖更小，操作更简单，且无需任何附加权限，不管是安装还是分发都更方便，适合插件固化后的最终成品打包发布。**

**内嵌和外置两种模式同时存在的情况下，外置模式优先。接下来讲的几步操作都是外置模式下的操作流程和注意事项。**

目前大多数的app都有加壳，所以这里的--target其实就是壳的so。每个第三方加固商都有对应的固定so名称，直接百度搜集即可。对于没有加固的app，大家可以查看/proc/$PID/maps文件，基地址最大的so即是第一个被加载的so。

如下图所示（必须要在root过的手机上才可以看maps文件）



libpcrash.so的基地址被映射的最大，可以尝试做为目标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**

**apktweak --apk xxx.apk --target assets/libjiagu\_a64.so --abi arm64-v8a**

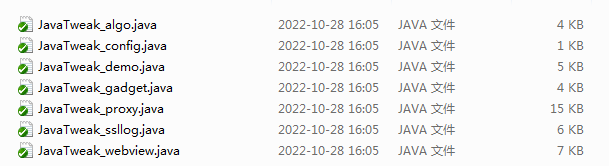
如果打包成功，则会在apk当前目录中生成一个重签名的新apk包，如下图所示

apk名称带sign的为重签名apk，另一个为原始apk



1. 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。源码中附带有数个简单的插件，默认会编译demo插件。



编译dex插件命令：**java2dex.bat**

如果想编译java插件，必须要安装java环境，建议安装java1.8。如果想编译sodemo，必须要安装NDK。如果想查看运行日志，必须要安装ddms，这个工具在安装了AndroidStudio或EclipseForAndroid之后会自动安装。

编译环境准备好后，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



五、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

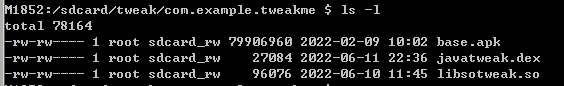
1. 在/sdcard目录下新建tweak目录（目录名称固定）
2. 在tweak目录下新建包名目录（不同的app包名不同，目录名称也不同），我这里就是

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：

**注意原始apk名称必须重命名为base.apk**

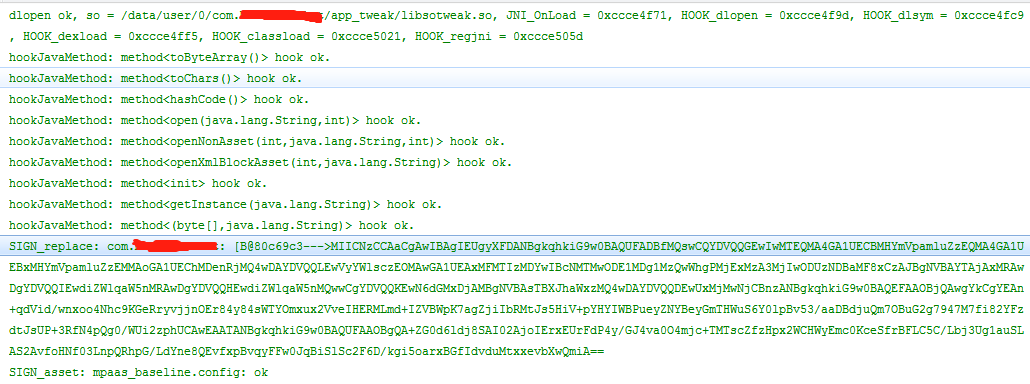


六、赋予app权限，允许读写/sdcard目录

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



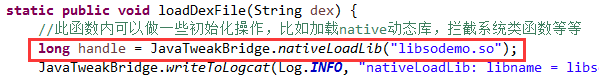
如果读写权限设置OK，会有如下类似日志打印



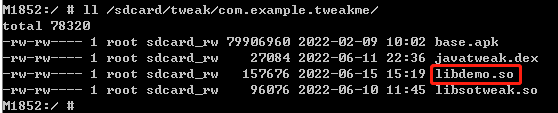
七、功能扩展：如何在java层调用自己编写的so

如果只开发dex插件的话，上面几步已经足够了。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）。

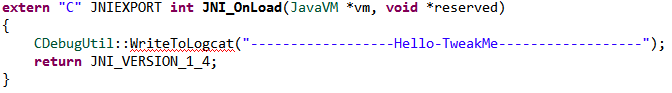
1. 在java层加载so，如下图



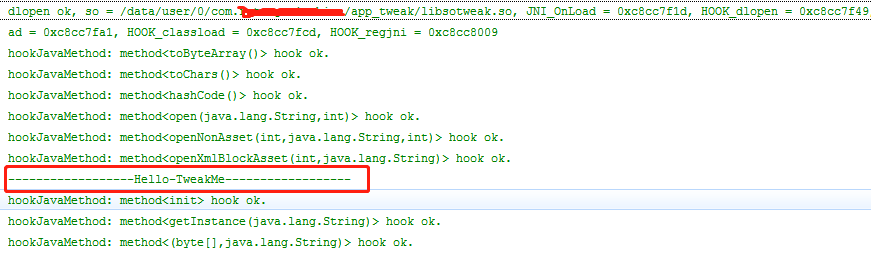
1. 将so放入libsotweak.so所在的目录中，如下图



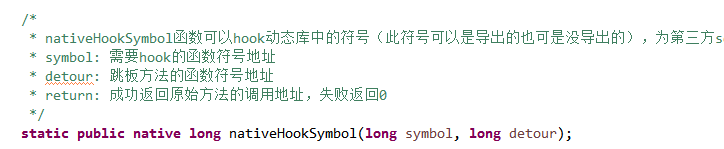
1. 我的libdemo.so在JNI\_OnLoad中只打印了一条简单的日志



日志输出如下



4、为了支持native层符号hook，在JavaTweakBridge类中有如下方法：



需要注意的是，此方法并不是用来拦截java层的native方法的，而是用来拦截native层的各种符号函数或者过程的（IDA中以sub\_xxxx这种形式显示）。具体使用请看函数说明，以及sodemo中的使用例子。

八、重签名apk

如果想手工调整apk包的内容（比如可以在WINRAR中增删文件），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**

九、添加额外文件到apk

通过常用的Zip可视化工具添加文件到apk中时，文件都是会被压缩再放入的。如果想不压缩放入文件，可以通过apktweak工具完成

Apk添加文件命令：**apktweak --apk xxx.apk --add file1[;file2;file3;...]**

下面是常用的添加单个文件的命令例子（--add参数中不能有空格和中文）

1. apktweak --apk xxx.apk --add c:\libdemo1.so
2. apktweak --apk xxx.apk --add c:\libdemo2.so@@**STORE**
3. apktweak --apk xxx.apk --add c:\libdemo3.so@lib/armeabi@**DEFLATE**
4. apktweak --apk xxx.apk --add c:\libdemo4.so@lib/armeabi@**STORE**

第一个例子表示将libdemo1.so**压缩**添加到apk的根目录

第二个例子表示将libdemo2.so**不压缩**添加到apk的根目录

第三个例子表示将libdemo3.so**压缩**添加到apk的lib/armeabi目录

第四个例子表示将libdemo4.so**不压缩**添加到apk的lib/armeabi目录

如果想要同时添加多个文件，那么文件与文件之间用**英文分号隔开**。

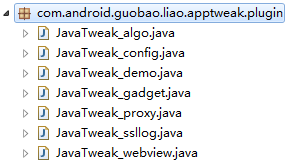
1. apktweak --apk xxx.apk

--add c:\libdemo1.so**;**c:\libdemo2.so@@STORE**;**c:\libdemo3.so@lib/armeabi

十、开发自己的java插件类

1、必须要放在com.android.guobao.liao.apptweak.plugin包中

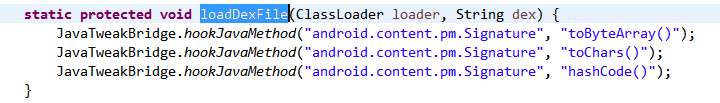
2、类名必须要以JavaTweak\_开头，如下图所示



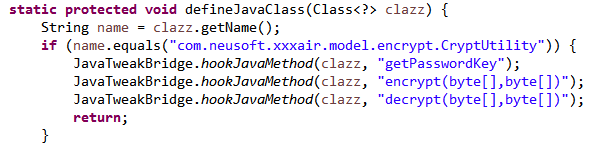
3、loadDexFile、defineClassLoader、defineJavaClass必须要至少实现其中一个，如下图demo插件



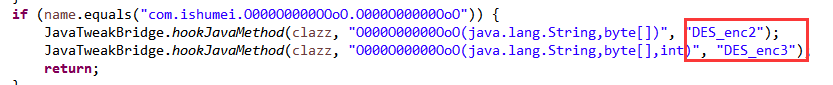
4、对于android.jar中实现的系统方法，建议在loadDexFile函数中hook



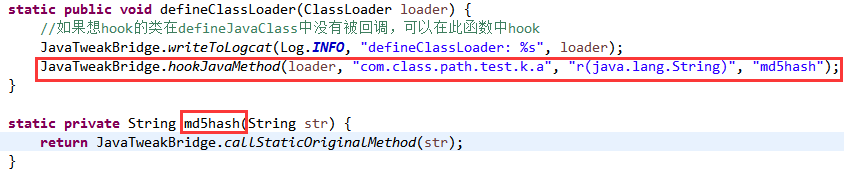
对于具体应用层的java方法，比如数据加解密相关的、请求包相关的方法，应该在defineJavaClass回调中hook，采用如下形式



如果需要hook类中的重载方法，一定要写明参数列表，对于混淆的方法，可以指定一个友好的方法名称。如下图



如果想要hook的类没有在defineJavaClass中回调，此种情况下，可以在defineClassLoader回调函数中hook目标类。写法如下



以上就是对TweakMe4.0框架的简单介绍，如果对TweakMe感兴趣的朋友在使用过程中遇到了问题，可以加入QQ群（**325509091**）提问，我会及时回复解答各位的问题。如果在使用框架重打包后出现app闪退的现象，可以在群里上传你的app样本，我也会及时修复框架问题。如果对框架有什么使用建议或改进意见，可以在群中@我，我会在下一个版本升级中考虑加入。如果觉得框架解决了你的某些问题，记得在github中为我点赞加星哦。