【转自看雪】反编译apk+eclipse中动态调试smali

转载

TreeExplore 于 2015-07-31 19:54:06 发布 654 收藏
分类专栏： JAVA


JAVA 专栏收录该内容
8 篇文章 0 订阅
订阅专栏

初涉移动端，请各位前辈多多指教！

本文参考http://www.kaifazhe.com/android_school/380973.html
在此，对作者表示感谢！

跟踪apk一般的做法是在反编译的smali代码中插入log输出，然后重新编译运行看输出日志，这种方法费时费力，如果能够动态调试就最好了。下面就给大家介绍apk+eclipse来调试smali。

前期准备：
eclipse。
Jdk或jre，并设置好环境变量，我的是7.0版本。
Apktool2.0
签名文件
目标apk：light.apk。

1. 将LIGHT.APK复制到APKTOOL目录下，如图：

利用apktool反编译light.apk。
命令：
D:\apktool2.0>java -jar apktool.jar d -d light.apk -o out(这里必须使用-d参数，这样反编译出来的代码后缀均是java，因为只有java文件才能被eclipse识别调试),如图：

在当前目录生成了反编译后的out文件目录：

2. 设置调试标记和寻找主类
在输出的out文件夹中，用文本编辑工具打开AndroidManifest.xml，在application节点中设置属性android:debuggable=”true”。

继续在AndroidManifest.xml中，搜索以下关键字

```
<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
```

找到含有以上信息的activity节点，记录其android:name属性的值，该值则为其应用的主类。本例主类为com.devuni.flashlight.MainActivity。

3. 在主类的ONCREATE事件中添加调试等待。
用文本编辑工具打开主类文件，找到onCreate方法，在第一句前插入invoke-static {}, Landroid/os/Debug;->waitForDebugger()V，记得添加a=0;//的前缀保持上下一致，结果如下：

4. 保存文件，用APKTOOL重新编译打包为DEBUG.APK
命令：
D:\apktool2.0>java -jar apktool.jar b -d out -o debug.apk
如图：

生成debug.apk文件：

5. 对DEBUG.APK签名(需要下载签名工具)，我把签名工具放在了SIGNAPK文件夹下，生成DEBUG.SIGN.APK
命令：

```
D:\APKTOOL2.0>JAVA -JAR .\SIGN\SIGNAPK.JAR .\SIGN\TESTKEY.X509.PEM .\SIGN\TESTKE
Y.PK8 DEBUG.APK DEBUG.SIG.APK
```

6. 上传DEBUG.SIGN.APK至手机或模拟器，然后安装并运行。这时你会看到程序运行后停留在白屏界面，这时不要动设备和退出程序，因为程序现在是运行到刚才添加的WAITFORDEBUGGER代码这里，这行代码的意思是一直挂起中，等待调试器。

下面开始设置实时调试的环境。

7. 进入第1步产生的OUT文件夹，把里面的BUILD和DIST文件夹删除，这是APKTOOL编译APK时产生的。
8. 启动ECLIPSE，构建JAVA项目
1) File -> New -> Project -> Java Project -> Next
2) Project Name随便起，Use default location选项去掉，Location选择out文件夹，然后Next
3) 把smali文件夹设为Source Folder，然后Finish
9. 在ECLIPSE中，打开第2步找到的主类，并找到ONCREATE方法，在WAITFORDEBUGGER后面的第一个方法开始添加断点。如下图

10. 打开DDMS，如果在第6步中运行了修改后的程序，在DDMS的设备列表中会显示可以调试的程序。

对应程序最后一栏为8602/8700，其中8602即为调试该程序的端口。

11. 现在要做的就是把代码与调试程序关联即可。 回到ECLIPSE，配置远程调试
1) 菜单Run -> Debug -> Debug Configurations
2) 双击Remote Java Application，Host处默认localhost就行，Port填第10步得到的8602，然后Apply -> Debug。

12. 这时ECLIPSE自动切换至DEBUG视图，并看到程序已经运行并中断在下一行可执行的代码了，相关的变量可以直接查看了。

已经可以用eclipse调试smali了，上面的例子是从程序开头的地方开始调试，但要调试到自己所关心地方的代码处确实麻烦。建议先用jd-gui等软件直接查看反编译的java代码，确定要调试的位置后，再进入smali定位断点并实时调试，就可以事半功倍。