Android逆向系列之动态调试5–代码注入(JDB调试)

转载

独坐一隅 于 2017-09-15 14:39:51 发布 2883 收藏 2
分类专栏： Android 文章标签： Android逆向 android动态调试 android代码注入JDB调试


Android 专栏收录该内容
21 篇文章 1 订阅
订阅专栏

一、环境

本次实验需要的环境有JDK、模拟器、adb

破解思路:通过jdb动态调试找到正确的变量，然后进行代码注入，修改Try again 为Hacked!!

二、准备

老操作，反编译APK，然后查看AndroidManifest.xml中是否有 android:debuggable=”true” 没有则在相应位置添加

启动模拟器或连接真机，然后安装apk –> adb install debug.apk

运行结果如图:

破解思路:通过jdb动态调试找到正确的变量，然后进行代码注入，修改Try again 为Hacked!!

三、获取进程PID

方法1: 可以使用DDMS，直接打开DDMS，如下图箭头所指:

方法2:adb jdwp，然后再打开应用(如果之前打开过apk，注意这里需要杀死进程)，再执行一次命令 adb jdwp，比较两者多出来的数字即是PID

四、JDB调试

这里注意，如果你的机器是使用共享内存(shared memory),那么请使用方法1,否则会报错，如果不是，可以使用方法1或者方法2，均可

方法1:

1.打开DDMS，选择需要调试的进程，然后再模拟器中启动apk

2.在终端输入: jdb -sourcepath .\src -connect com.sun.jdi.SocketAttach:hostname=localhost,port=8700

成功如图:

方法2:
1.使用adb转发端口: adb forward tcp:54321 jdwp:1234（注解:这里的54321可任意，但尽量避免端口冲突，这里的1234为第三步获取的PID）
2.jdb连接: jdb -attach localhost:54321（注解:这里的端口注意与上面对应）
成功也类似上图.

五、远程代码注入
1.查看类
使用classes命令查看所有的类，这里会打印出相当多的类名，可以结合DDMS找到需要类名
当然直接反编译后可查看相当多的类名，具体可使用class 类名或ID 进行查看

```
id
```

2.查看方法
命令: methods com.example.debug.MainActivity$1

```
方法
```

3.设置断点
选择我们自己感兴趣的地方，即关键的地方进行断点设置。
命令: stop in com.example.debug.MainActivity$1.onClick(android.view.View)

```
断点
```

4.触发断点
点击apk的按钮，触发断点，视具体应用而不同，这里仅需点击

```
命中断点
```

5.查看局部变量
命令: locals

6.执行下一条命令
命令: next

7.进入方法
命令: step

8.文本设置方法
命令: set 变量名=”xxxxx”

以上调试步骤结果如下:

```
调试步骤
```

注意这里，触发断点后，如果执行一次next命令没断下来或者使用locals没查看到变量，可以尝试多执行几下next命令或者step命令

9.运行程序
命令: run

10.代码注入结果

tasfa hack

参考资料: 点我传送

本文工具下载: 360安全播报提供 http://yunpan.cn/cf3RVN5fRRC73 （提取码：8734）
转载自http://www.tasfa.cn/index.php/2016/05/31/code-injection-jdb/