 今天国庆的第六天，西安淅淅沥沥小雨下个不停，索性卧在床上不如品一杯西湖龙井，更新一篇博客，一来帮助需要之人；二来加深自己的理解。

**问题篇：**

在上一篇中讲到了关于Android so的动态调试，没看的可以点这里：[点击打开链接](http://blog.csdn.net/feibabeibei_beibei/article/details/52740212)；

我自认为写的还是挺全的，在上文中我们说到关于最后一步jdb附加调试时，很多时候都会出现附加不上的问题，使人很闹心。。。于是这一篇就是专门关于这个问题进行展开的。解决这个问题方法有很多，我是按照自己认为的优良答案顺序展开的，都是借鉴网上的各路大神而总结的。

**解决篇：**

根据android的官方文档，如果调试一个APK，必须满足以下两个条件中的任何一个：

1.APK的AndroidManifest.xml文件中的Application标签包含android:debuggable="true";

2./default.prop中的ro.debuggable的值为1；

***方法一：***

在已经root的手机安装Xposed框架和xinstaller插件

目的：就是利用Xposed的HOOK插件xinstaller开启系统中所有应用的调试功能。

使用方法：

第一步：下载Xposed框架，并激活，再下载xinstaller插件安装；

第二步：开启模块，点击xinstall插件设置专家模式，进入其他设置，开启调试应用，最后在xposed中激活重启，OK！！

关于框架和xinstaller插件会放在附件中。

***方法二：***

修改android:debuggable="true"

用AK反编译以后在AndroidManifest.xml文件中的Application标签中加入android:debuggable="true";然后回编译。

这个方法虽然简单，但是问题多，比如：

第一：有的反编译，签名验证等等。

第二：如果说软件已经爱加密或加壳了，修改XML几乎是不可能的，因为改了也不能回包

于是引入了方法三.

***方法三：***

如果我们在真机，则可以修改根目录下的default.prop文件，将里面的ro.debuggable =1。

第一：  从Google官方网站下载到boot.img；  
第二：  使用工具（abootimg，gunzip, cpio）把boot.img完全解开，获取到default.prop；  
第三：  修改default.prop；  
第四： 把修改后的文件重新打包成boot\_new.img；  
第五： 使用fastboot工具把boot\_new.img刷入设备（fastboot flash boot boot\_new.img）；

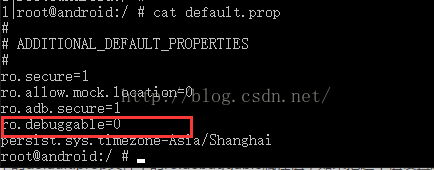
优点是可以永远调试，不再担心此问题；

缺点是我没有Nexus 5手机骂人，这个问题很严肃；于是引出了方法四。

***方法四：***

我们没有谷歌的亲儿子，但是我们有神器的小工具。

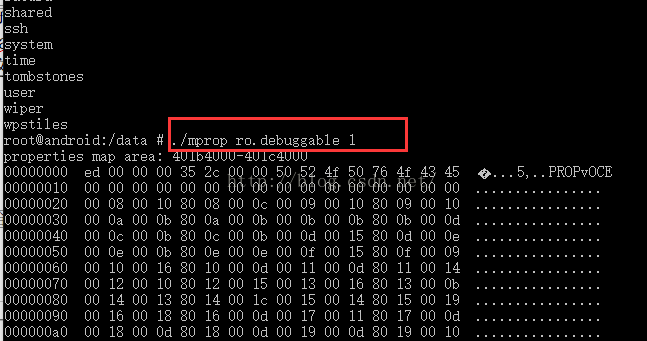
首先我们看到如图所示ro.debuggable=0；



init进程会解析这个default.prop文件，然后把这些属性信息解析到内存中，给所有app进行访问使用，所以在init进程的内存块中是存在这些属性值的，那么这时候我们可以利用进程注入技术，我们可以使用ptrace注入到init进程，然后修改内存中的这些属性值，只要init进程不重启的话，那么这些属性值就会起效。当然这个工具已经写好，我会放在后main附件中。

解决方法：

第一步：拷贝mprop 到/data/目录下；  
第二步：./mprop ro.debuggable 1；  
第三步：getprop ro.debuggable;（查看此时ro.debuggable在内存中的值）  
第四步：stop;start(重启adbd进程)；



注意：

我们在上面的第三步的时候查看好像没有改过来，分析原因为：

该工具是通过ptrace修改init进程中的内存，然而4.X系统强制开启了selinux;因此这个时候我们需要设置Selinux的状态.需要个APK，会放在后面的附件中。

**BY：雪一梦**

**2016.10.06**