篡改app名字和图标

实验概述

本实验通过反编译获得apk的资源和源码，修改对应的资源和代码从而修改app的名字和图标，再重签名得到一个修改后的apk

实验目的

1、了解apk反编译过程

2、认知反编译后的文件结构

3、了解签名的制作过程

实验原理

ApkTool是跨平台工具，可以在linux和windows下直接使用

反编译命令为：apktool d[ecode] [OPTS] <file.apk> [<dir>]

回编译命令为：apktool b[uid] [OPTS] <app_path> [<out_file>]

反编译apk文件成功后，会在当前目录下生成一系列目录与文件。AndroidManifest.xml文件存放了程序的基本信息，如版本、程序包名、主题、权限等。smali目录存放了程序所有反汇编代码，res目录存放着所有的图片文件以及xml文件，也就是在eclipse工程目录中res下面的所有资源，这些目录的子目录和文件与开发时的源码目录组织结构是一致的。

/res文件夹下的几个目录：

anim：已编译的动画文件

drawable：存放着apk所有的图片

layout：UI/视图定义

values：数组、颜色、尺寸、字符串和样式

xml：已编译的任意xml文件

raw：未编译的原始文件

在修改完apk且回编译成功之后，apk并不能成功安装，在Android 系统中，所有安装到系统的应用程序都必有一个数字证书，此数字证书用于标识应用程序的作者和在应用程序之间建立信任关系,如果一个 permission的protectionLevel为signature，那么就只有那些跟该permission所在的程序拥有同一个数字证书的应用程序才能取得该权限。Android使用Java的数字证书相关的机制来给apk加盖数字证书。Android系统要求每一个安装进系统的应用程序都是经过数字证书签名的，数字证书的私钥则保存在程序开发者的手中。Android将数字证书用来标识应用程序的作者和在应用程序之间建立信任关系，不是用来决定最终用户可以安装哪些应用程序。

keytool是kali下自带的签名工具

命令： keytool -genkey -v -keystore android.keystore -alias android -keyalg RSA -validity 20000

该命令中，-keystore andriod.keystore 表示生成的证书，可以加上路径（默认在用户主目录下）；-alias android 表示证书的别名是android；-keyalg RSA 表示采用的RSA算法；-validity 20000表示证书的有效期是20000天。

实验环境

实验环境：kali linux

实验工具：apktool、keytool、android sdk、game.apk

模拟器：android 4.0

实验步骤

1、“打开终端”>“cd android-sdk-linux/tools/”>“./android”来启动android sdk如图 1

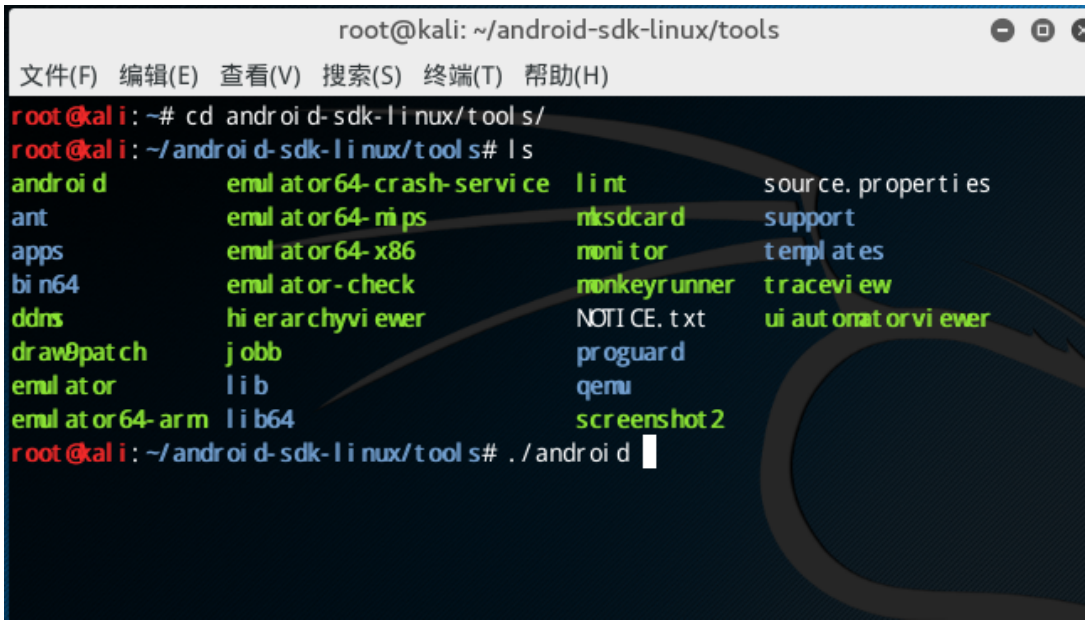

图 1开启android sdk

2、“单击tools”>“选择Manage AVD”打开虚拟机控制台，如图 2

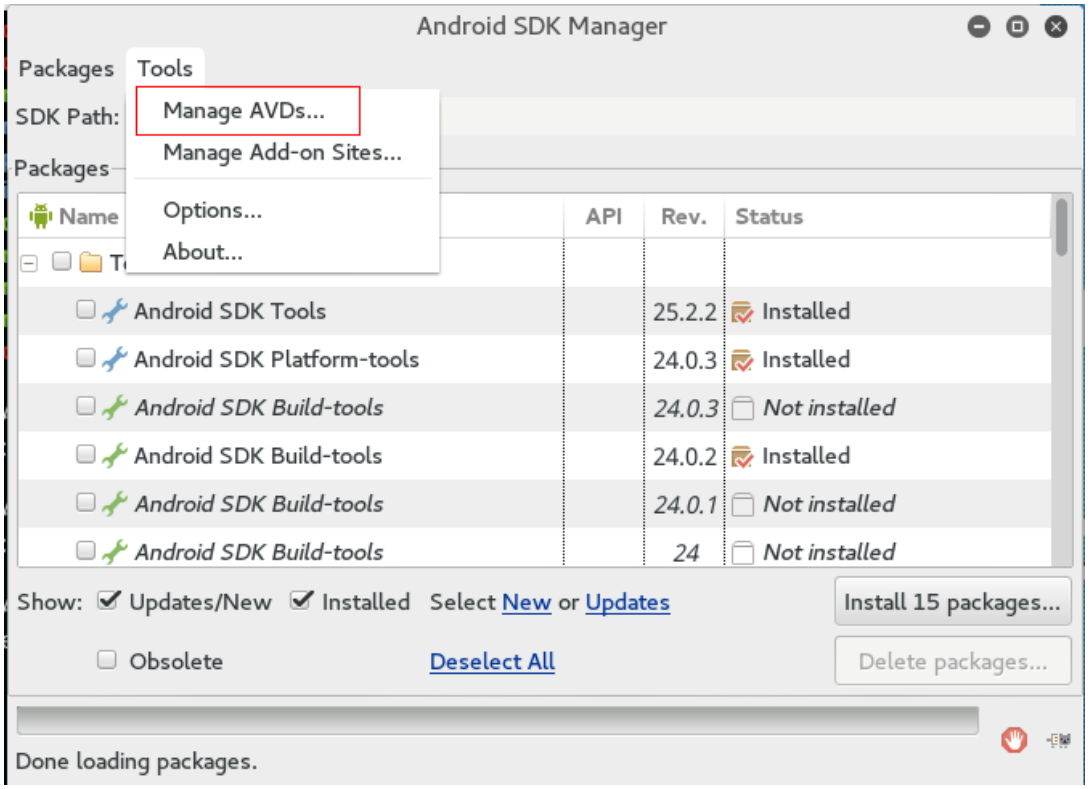

图 2开启模拟器控制台

3、选择创建好的Android虚拟机单击start来开启虚拟机，如图 3

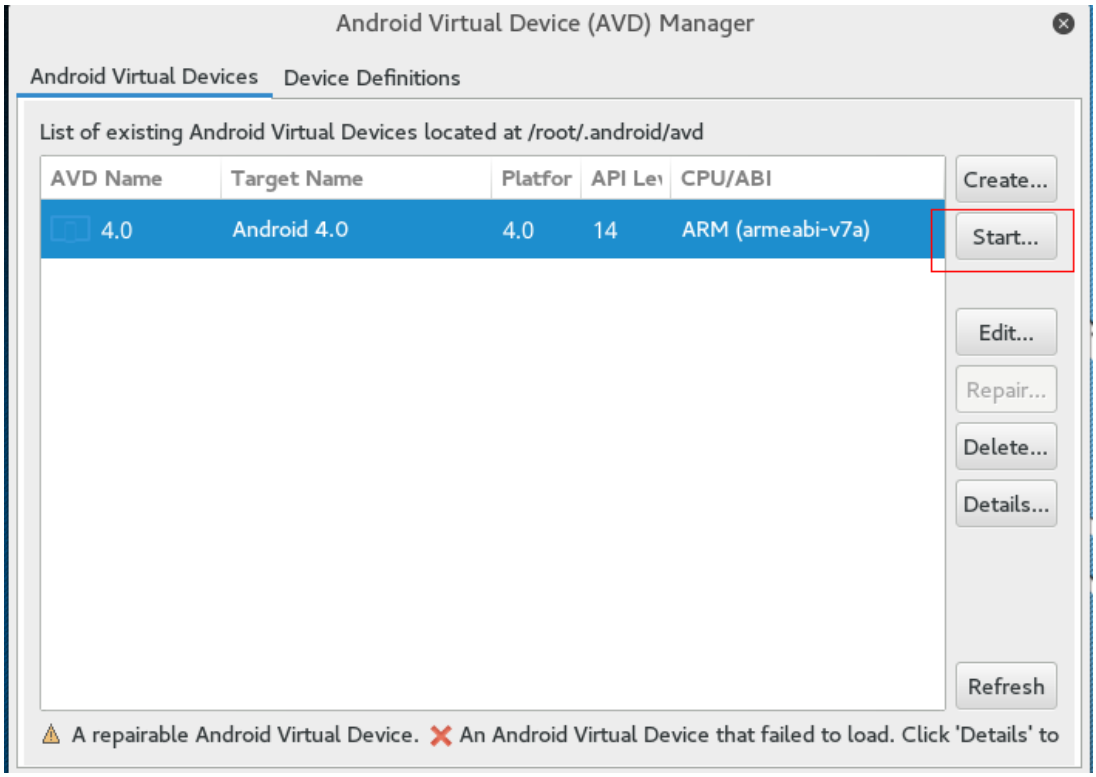

图 3开启模拟器

4、此处可设置屏幕的尺寸，使用默认值，单击launch，如图 4

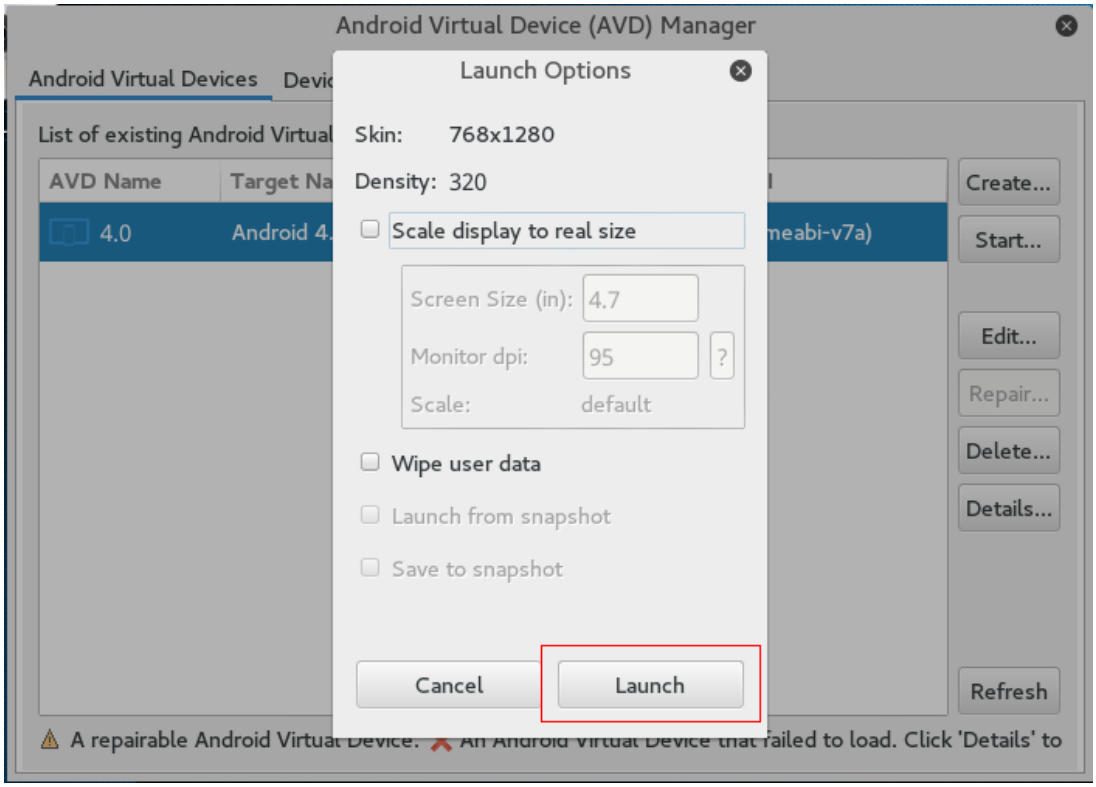

图 4开启模拟器

5、成功开启android虚拟机，桌面如下图 5

图 5模拟器桌面

6、进入apk目录，给android模拟器安装实验app。“打开终端”>“cd apk/”>“adb install game.apk”如图 6

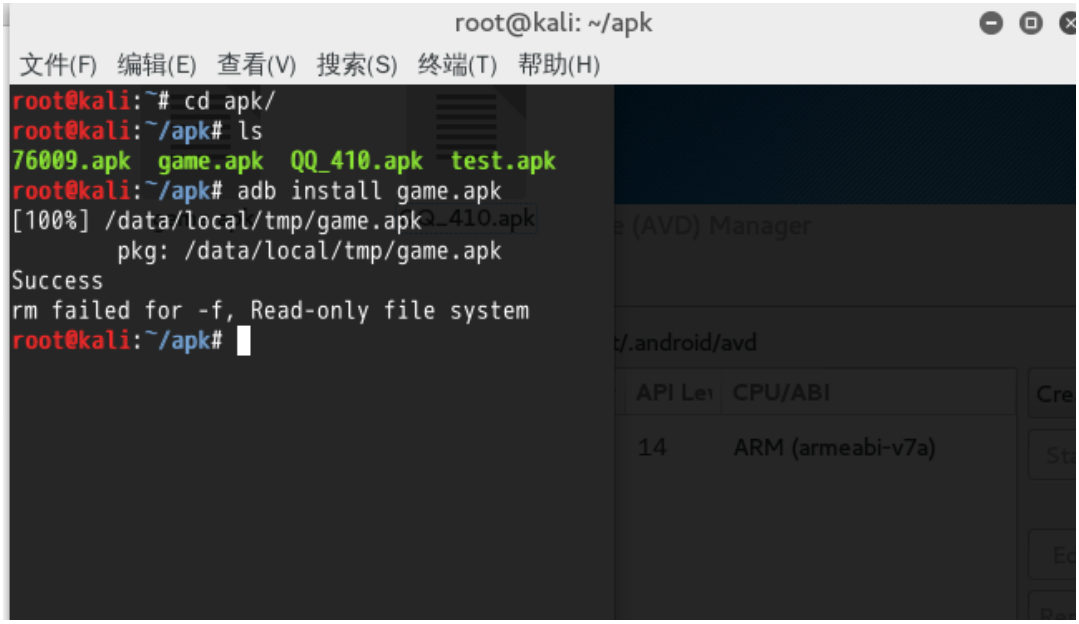

图 6安装实验app

7、打开成功安装的游戏，如图 7

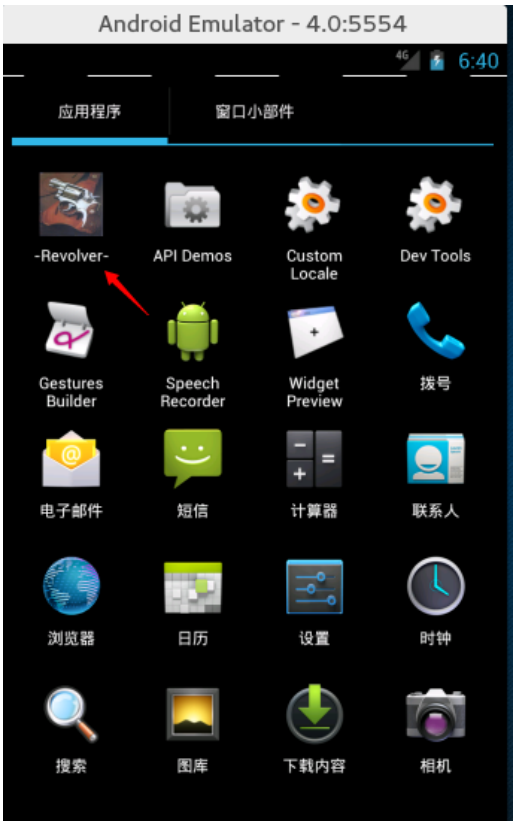

图 7游戏app界面

8、对实验apk进行反编译，使用命令：“apktool d game.apk”，如图 8

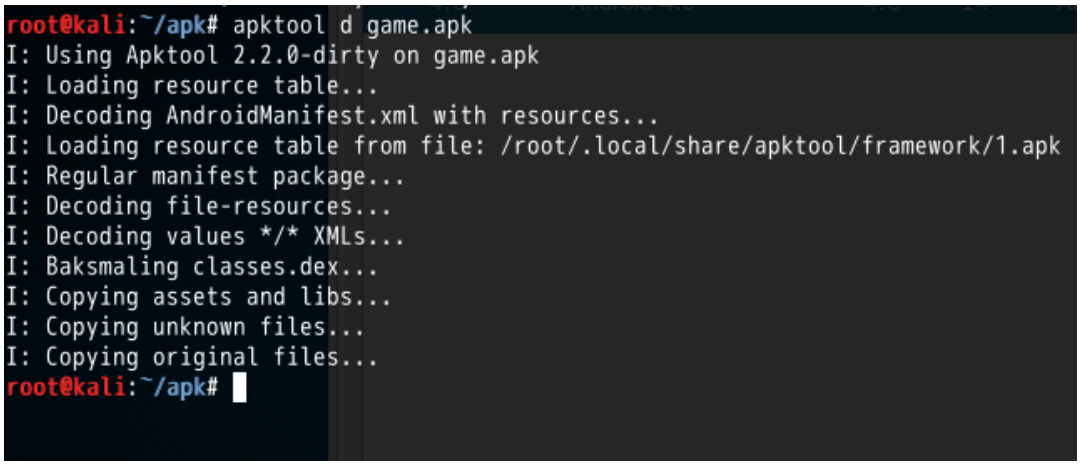

图 8反编译

9、编译成功后，会在当前目录生成game文件夹，进入存放图片的drawable文件夹中，查看app图标的图片。“cd game”>“cd res”>“cd drawable”>“ls” 如图 9

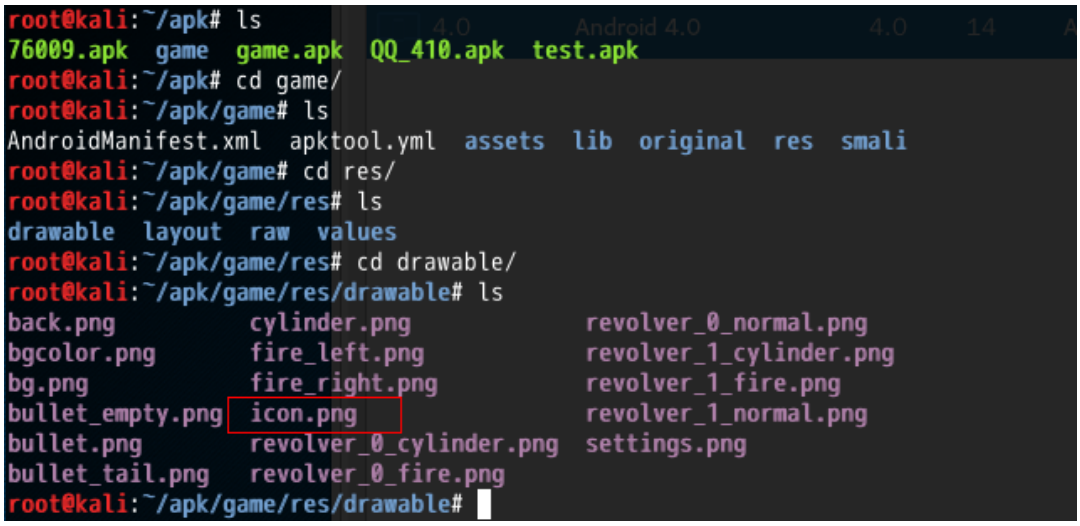

图 9图标的图片

10、替换app图标的图片，“mv /root/bluedon.png ./”>“mv bluedon.png ./icon.png”如图 10

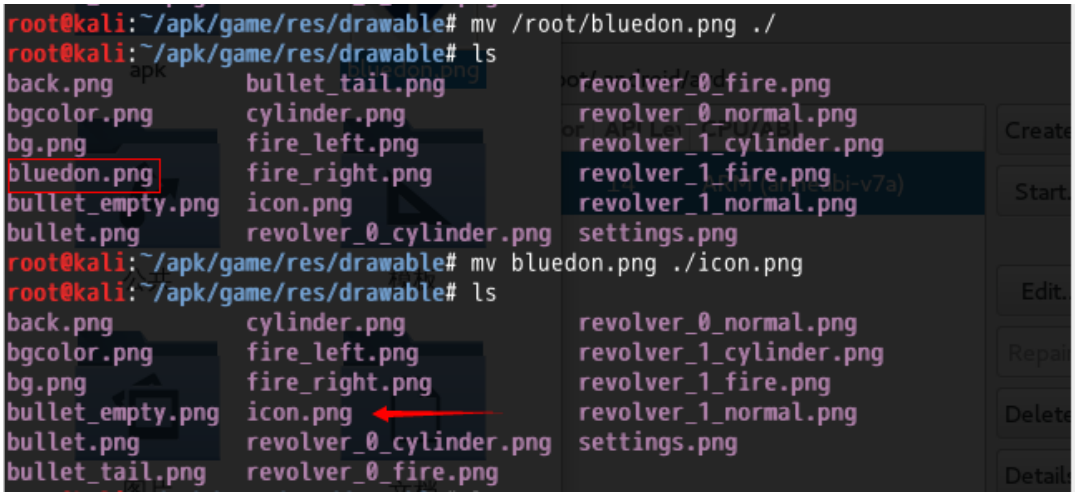

图 10替换图标

11、进入到res的values目录找到strings.xml文件，如图 11

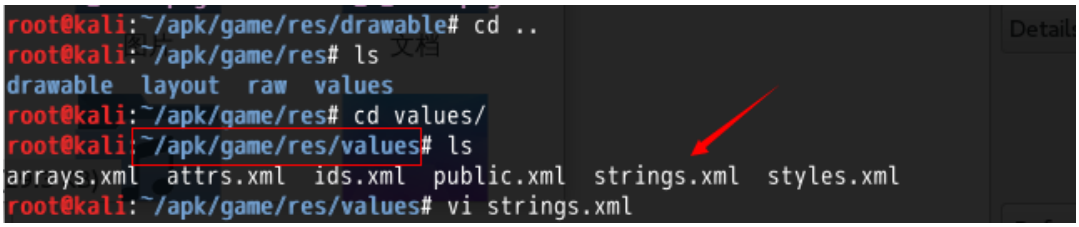

图 11打开strings.xml

12、编辑strings.xml文件的app_name，如图 12图 13

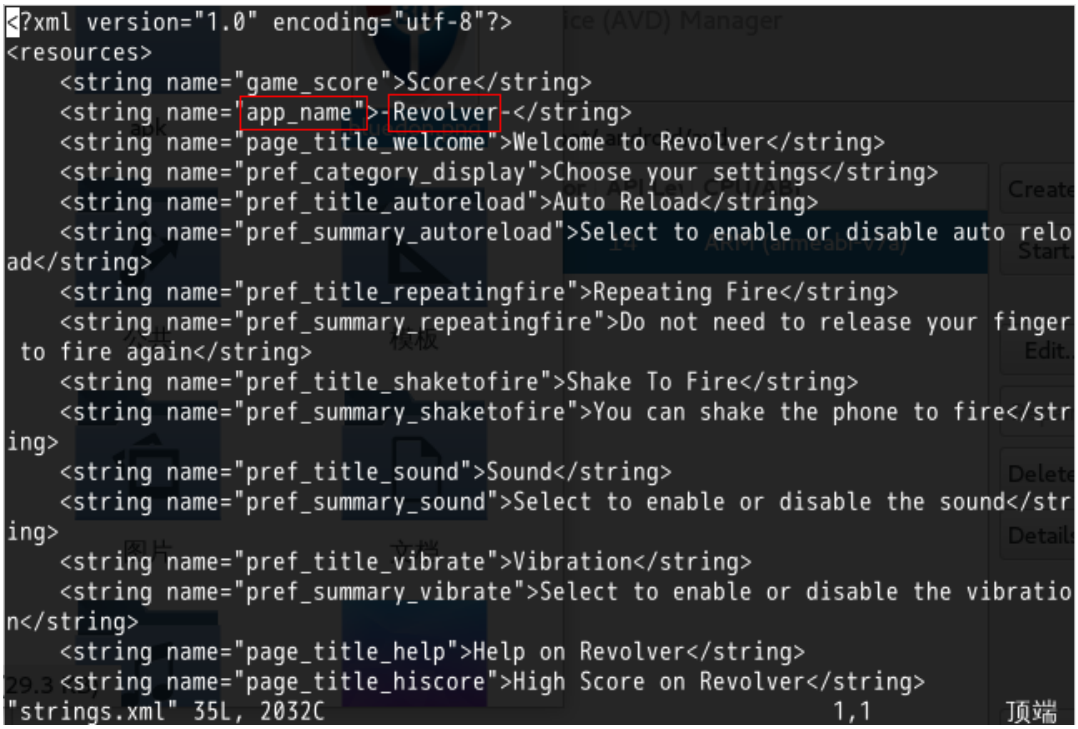

图 12修改前

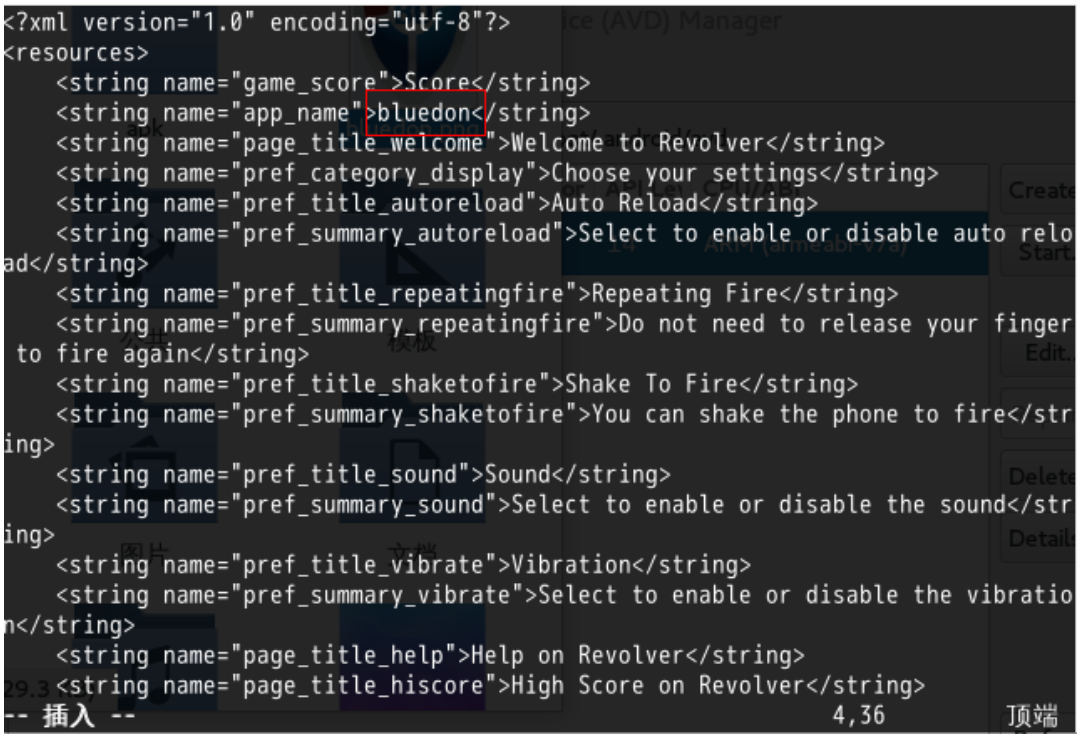

图 13修改后

13、进入apk目录对game进行重打包。“cd apk/”>“apktool b game”，如图 14

图 14回编译apk

14、打包成功后，apk包会生成在game的dist文件夹中，如图 15

图 15未签名的apk

15、复制game.apk到签名工具中，“cd /root/apk/game/dist”>“cp game.apk /root/tools/qianming”>“ls” 如图 16

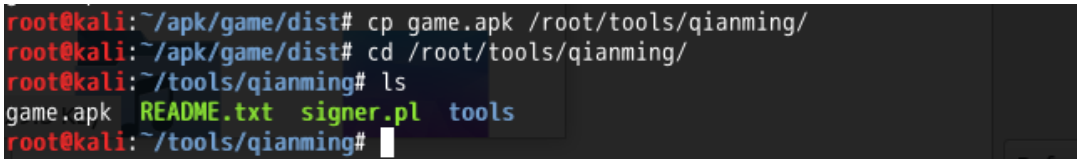

图 16复制到签名工具中

16、创建签名，“cd /root/tools”>“keytool -genkey -v -keystore bluedon.keystire -alias bluedon -keyalg RSA -validity 365”，填写相关签名资料，如图 17

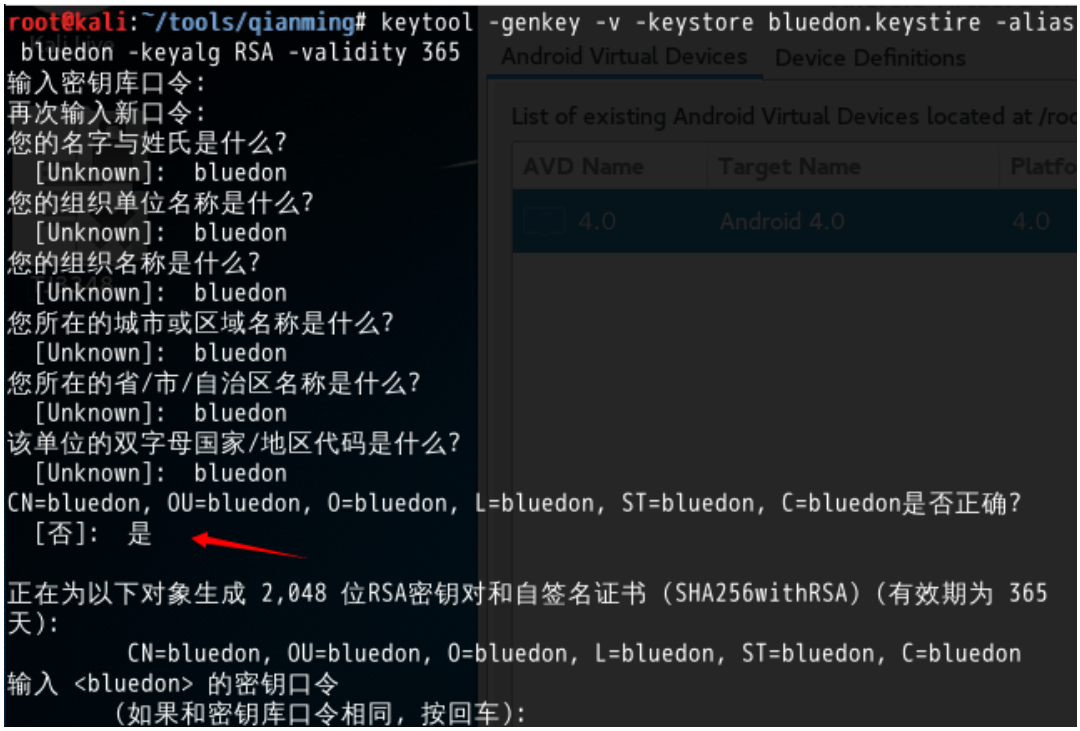

图 17设置签名信息

17、成功生成签名文件，如图 18

图 18签名文件

18、对重打包的game.apk进行签名，使用命令：“perl signer.pl -k bluedon.keystire -p1 123456 -a bluedon -p2 123456 -s ./game.apk -d ./ ”如图 19

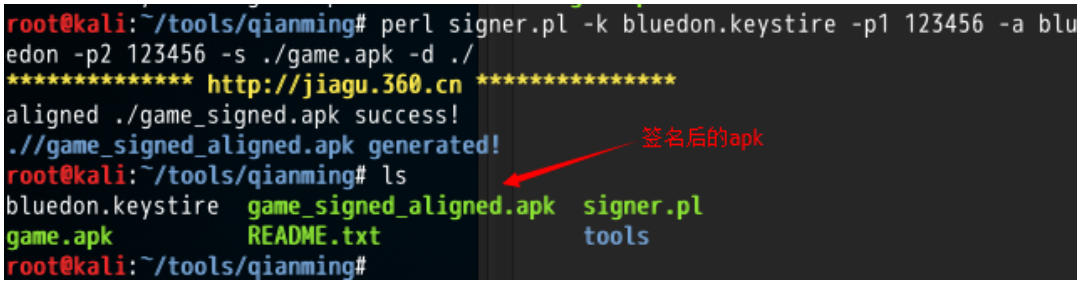

图 19签名

19、在android模拟器对初始apk进行卸载，如图 20

图 20模拟器卸载app

20、安装签名成功后的apk，使用命令：“adb install game_signed_aligned.apk”，如图 21

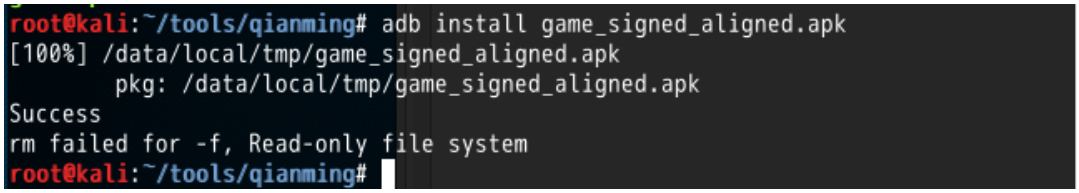

图 21安装修改后的apk

21、在android模拟器中，查看是否修改成功。如图 22

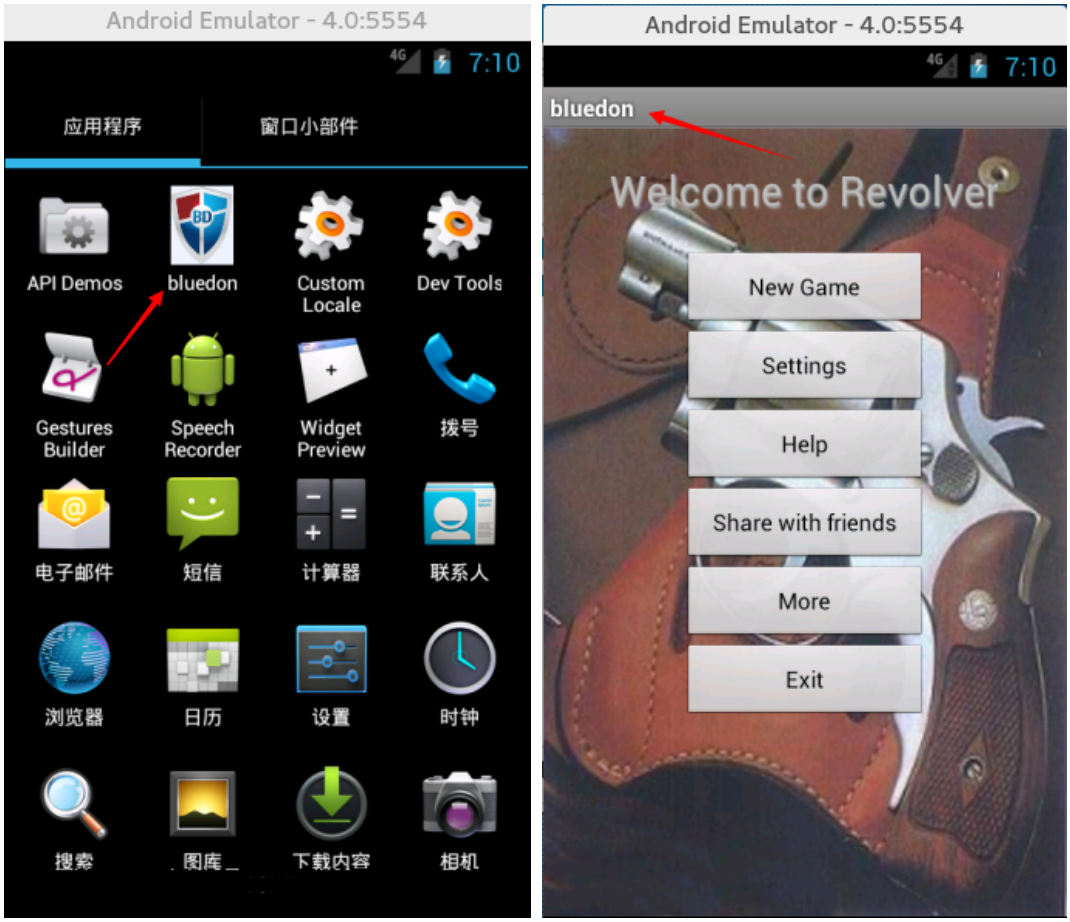

图 22修改成功

思考总结

本实验通过使用apktool对apk进行反编译，修改其中的资源和代码，从而对app的名字和图标进行修改，再使用keytool对apk进行重签名