Experiment Report

Name	张三	Student ID	XXX
Exp. Title	SecureCoding-Permission && ReverseEngineering	Exp. Date	2014/12/5

一、 Basic Principles (原理简述)

SecureCoding-Permission:

In order to protect device resources, Android enforces a resource permission system such that every Android app needs to declare its own permission requirement in a AndroidManifest.xml file using <permission> tags. By default, an Android app has no permissions associated with it, meaning it cannot do much thing and it can't access data on the device.

There are many Android built-in permissions such as contact-list, GPS, camera, SMS, Internet, Etc.

ReverseEngineering:

Reverse engineering is a method to take apart an object to see how it works in order to duplicate or enhance the object, it is a practise taken from older industries, and is now frequently used on computer hardware and software.

Software reverse engineering involves reversing a program's machine code (string 0's and 1's that are sent to the logic processor) back into the source code that it was written in, using program language statements.

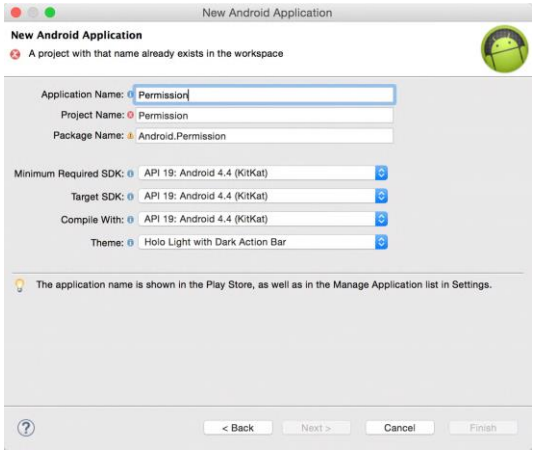
Steps as following:

0. Decompile APK to JavaCode
1. Decompile apkfile to get AndroidManifest.xml
2. Analyze the AndroidManifest.xml file with a set of rules

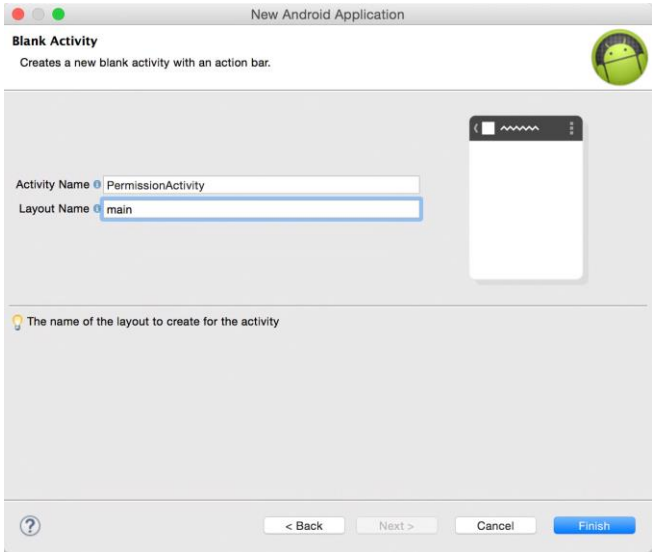
二、Step-by-Step Procedure (实验步骤)

SecureCoding-Permission:

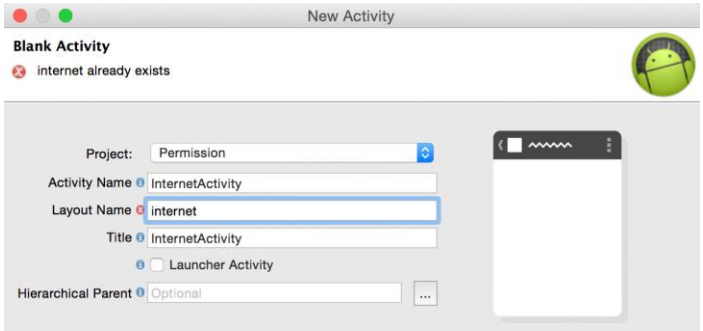
1. 新建工程，命名为 Permission，同时 Package name 修改为 Android.Permission，设备要求均选 API19



然后点击 next，直到 Blank Activity wizard，修改 Activity Name 为 PermissionActivity，Layout name 为 main，然后点击 Finish.



然后建立两个空白 activity，对应的 Activity Name 和 Layout Name 分别为 InternetActivity , internet 和 SMSActivity , sms：



（sms 在这里就不截图了）

2. 键入相关代码，其中包括 main.xml, internet.xml, sms.xml, PermissionActivity.java, InternetActivity.java, SMSActivity.java，具体代码可以参见工程文件，这里面不予展示。不过有两点需要说明的的是：

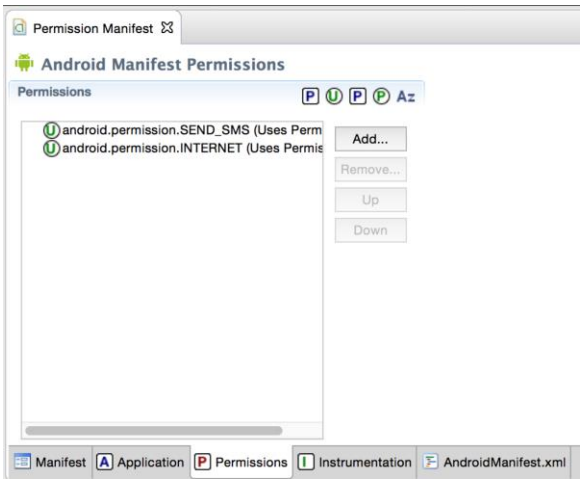
   i. sms.xml 代码有误，应该删除

   </LinearLayout> <WebView android:layout_width="match_parent" android:layout_height="match_parent" android:id="@+id/webView"></WebView>

   ii. 最后要修改 AndroidManifest.xml，加入权限代码

```
<uses-permission
android:name="android.permission.SEND_SMS"></uses-permission>
<uses-permission android:name="android.permission.INTERNET" />
```

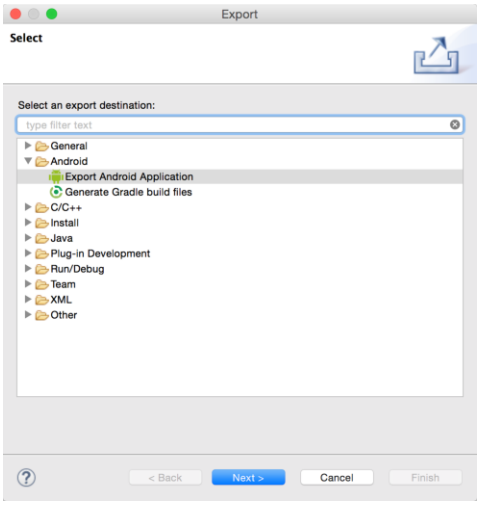
   如果添加成功，可以在 Permissions 栏看到对应的两个权限：

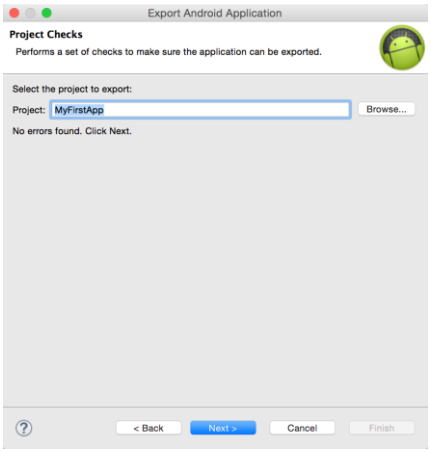


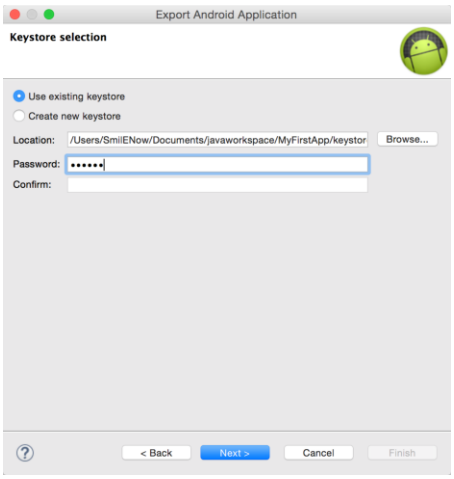
3. 运行工程，然后测试浏览网页和发送短信的功能。
4. 把步骤二中的两行权限代码删除，然后重新运行工程，观察浏览网页和发送短信的功能是否能成功执行。

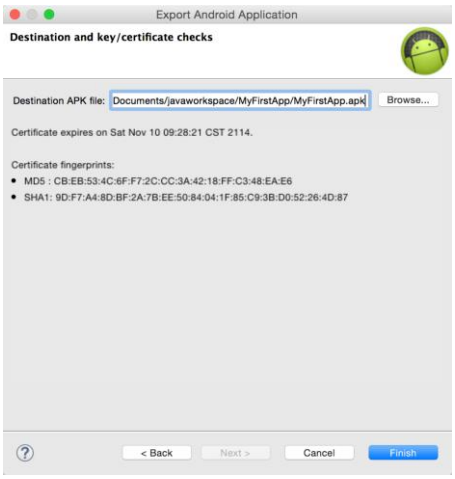
ReverseEngineering:

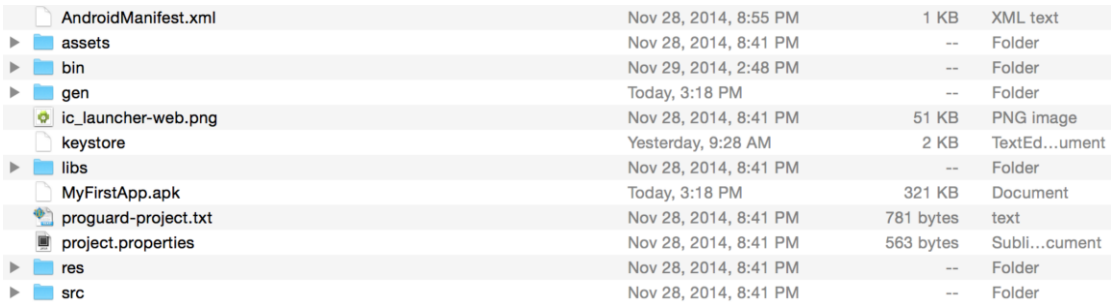
1. 打开 Eclipse，找到 Lab1 的 MyFirstApp 工程，导出 apk 文件：
   在左侧 Package Explorer 的 MyFirstApp 工程文件右键，Export->Export Android Application, 如果之前没有 keystore 就 create，如果有就选中当前的 keystore，然后输入密码，生成对应的 APK 文件。











2. 在 Macintosh 下不需要把后缀名由 apk 修改为 zip 然后解压，对 classes.dex 进行 Decompile，而是可以直接 decompile apk

（具体可参见：https://code.google.com/p/dex2jar/wiki/UserGuide ）

1. Download dex2jar from

   http://code.google.com/p/dex2jar/downloads/list

2. Extract dex2jar-version.zip to a folder. for example /home/panxiaobo/, C:\

```
unzip -x dex2jar-version.zip -d /home/panxiaobo
```

3. use dex2jar to generate .jar file. dex2jar will generate a file named someApk-dex2jar.jar in the working folder.

```
linux sh /home/panxiaobo/dex2jar-version/d2j-dex2jar.sh /home/panxiaobo/someApk.apk
windows C:\dex2jar-version\d2j-dex2jar.bat someApk.apk
```

4. use a decompiler to view the source. jd-gui JAD

that's it

From: https://code.google.com/p/dex2jar/wiki/UserGuide

```
TroySmilENow:~ SmilENow$ Documents/javaworkspace/dex2jar-0.0.9.15/d2j-dex2jar.sh Desktop/MFA/MyFirstApp.apk
dex2jar Desktop/MFA/MyFirstApp.apk -> MyFirstApp-dex2jar.jar
```

然后把生成出来的 MyFirstApp-dex2jar.jar 用 JD-GUI 打开

3. 使用 apkfile 来获得 AndroidManifest.xml 文件：

Macintosh 下安装 apkfile 需要用 root 权限修改/usr/local/bin，具体可以参见官网：

https://code.google.com/p/android-apktool/wiki/Install

这里面就略过安装部分，不予截图展示了。

具体使用可以参见：

https://code.google.com/p/android-apktool/wiki/ApktoolOptions

然后打开 terminal，找到 apk 文件对应的文件夹，之后使用 apktool：

$ sudo apktool d -f MyFirstApp.apk -o ./tmp/

```
TroySmilENow:MFA SmilENow$ sudo apktool d -f MyFirstApp.apk -o ./tmp/
I: Using Apktool 2.0.0-RC3 on MyFirstApp.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /var/root/Library/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
TroySmilENow:MFA SmilENow$
```

然后在 tmp/文件夹下可以找到 AndroidManifest.xml

4. 编写 Java 程序来分析 AndroidManifest.xml，输出结果为警告级别。

具体警告级别定义如下：

warning level: 0 是最低，2 是最高

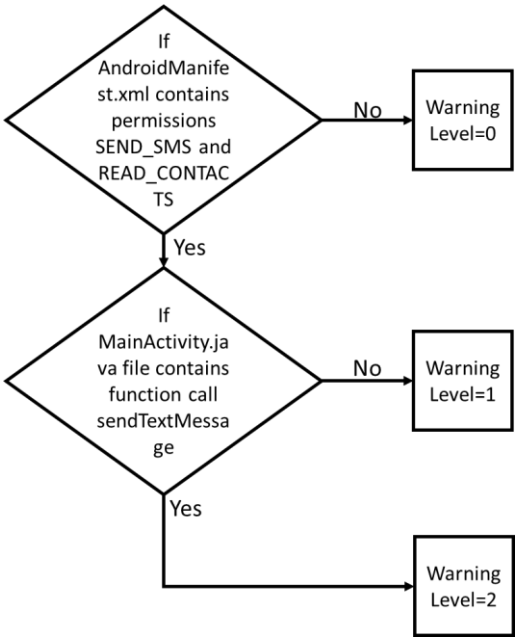
warning level 等于 2 时，说明 AndroidManifest.xml 里面具有 SEND_SMS,READ_CONTACTS 的权限，并且 MainActivity.java 中含有 sendTextMessage 函数

warning level 等于 1 时，说明 AndroidManifest.xml 里面具有 SEND_SMS,READ_CONTACTS 的权限，但 MainActivity.java 不含有 sendTextMessage 函数

warning level 等于 0 时，说明 AndroidManifest.xml 里面不同时具有

SEND_SMS,READ_CONTACTS 的权限。

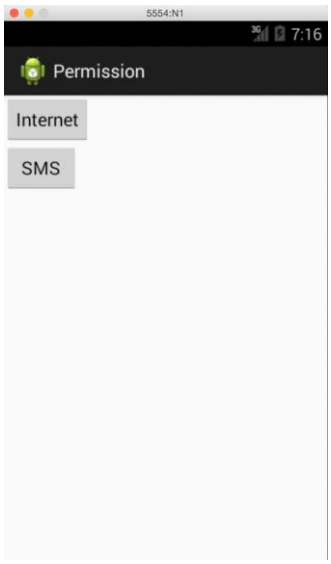
代码可以参见 auto_reverse 工程，具体分析和测试看第三部分。



三、Results and Analysis (结果与分析)

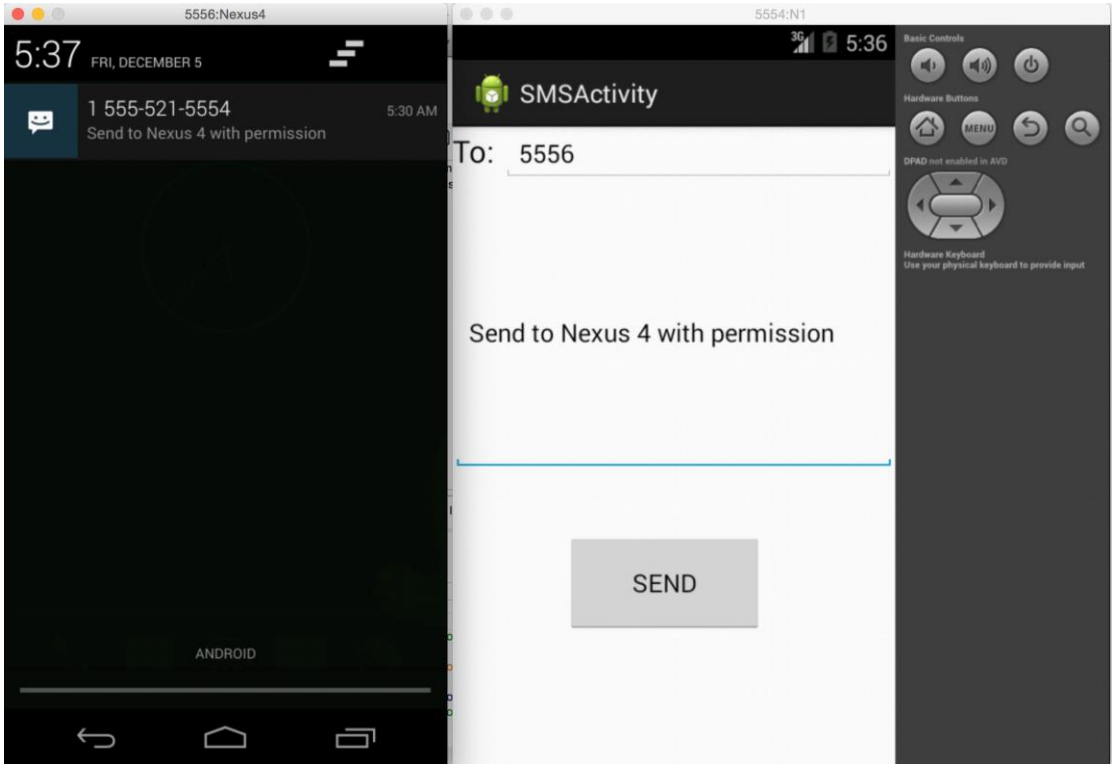
SecureCoding-Permission:

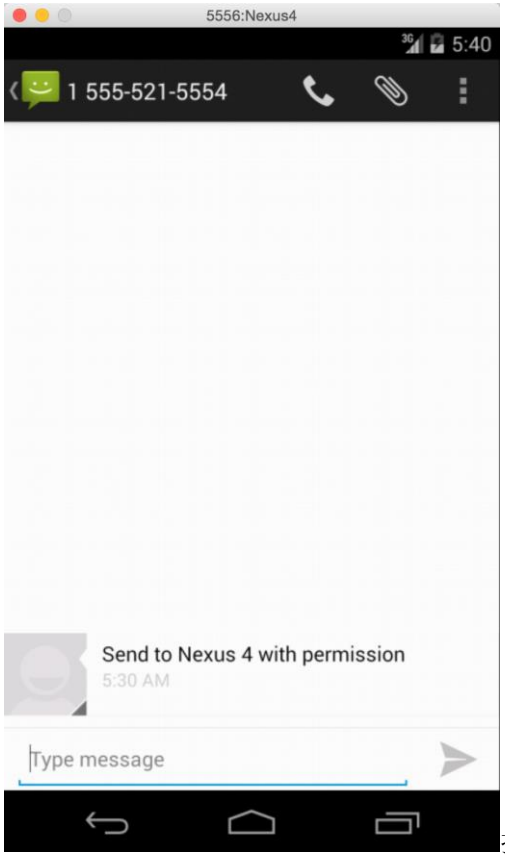
由于 Genymotion 不支持 SMS，所以在 Eclipse 上下载了 Intel Atom(x86) AVD 来运行程序：



添加权限之后，点击 Internet 按钮，浏览网页：

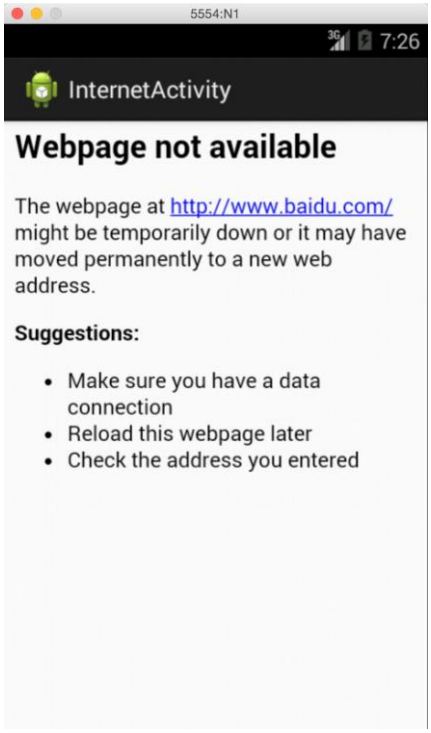
发送短信：



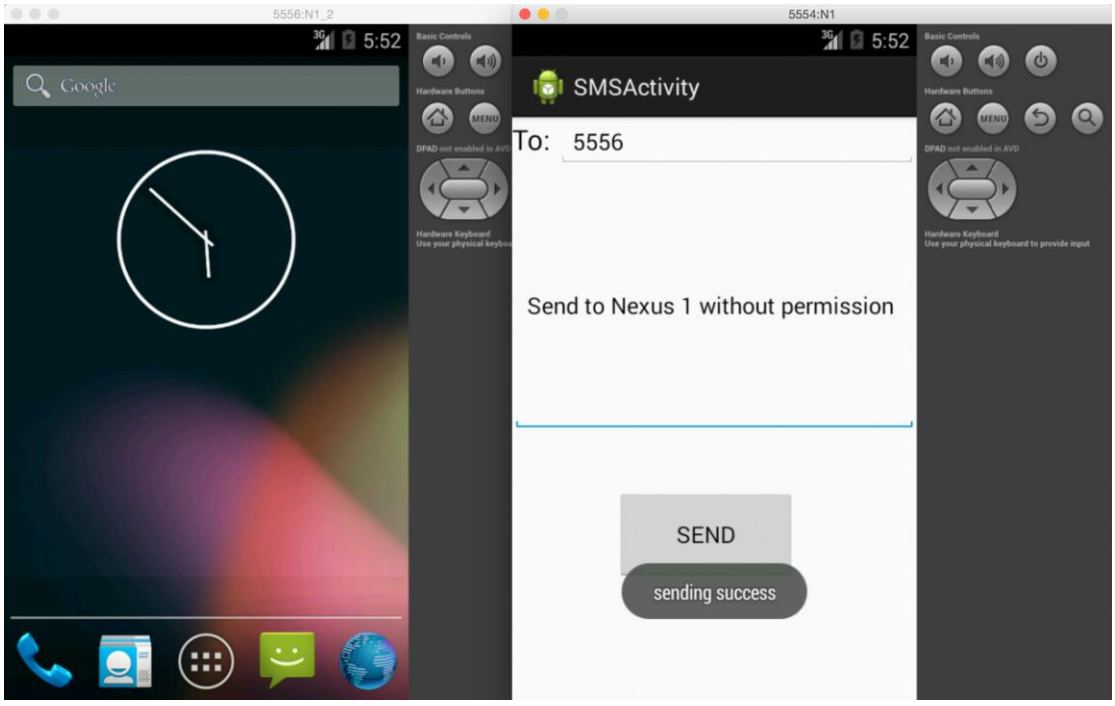


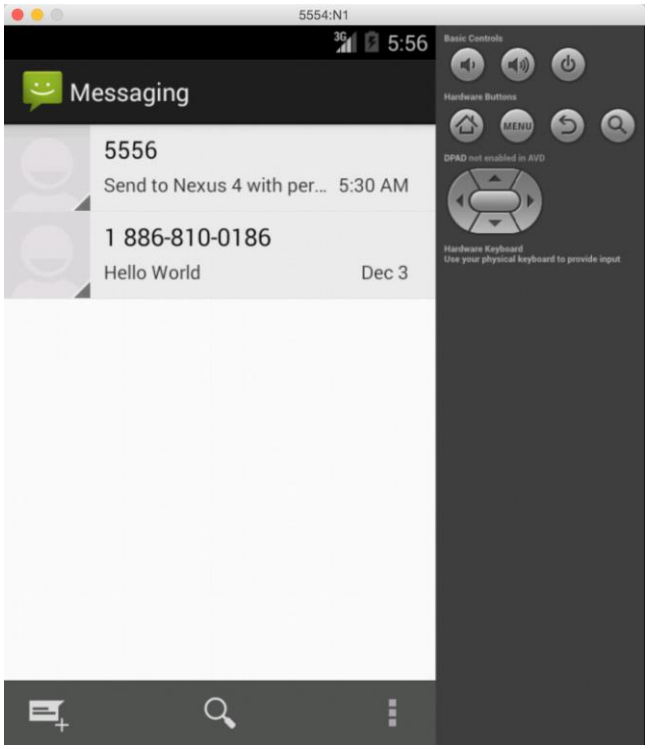
接收方的短信收件箱能收到这条短信

然后修改代码，删除相关的权限之后，浏览网页：



发送短信：





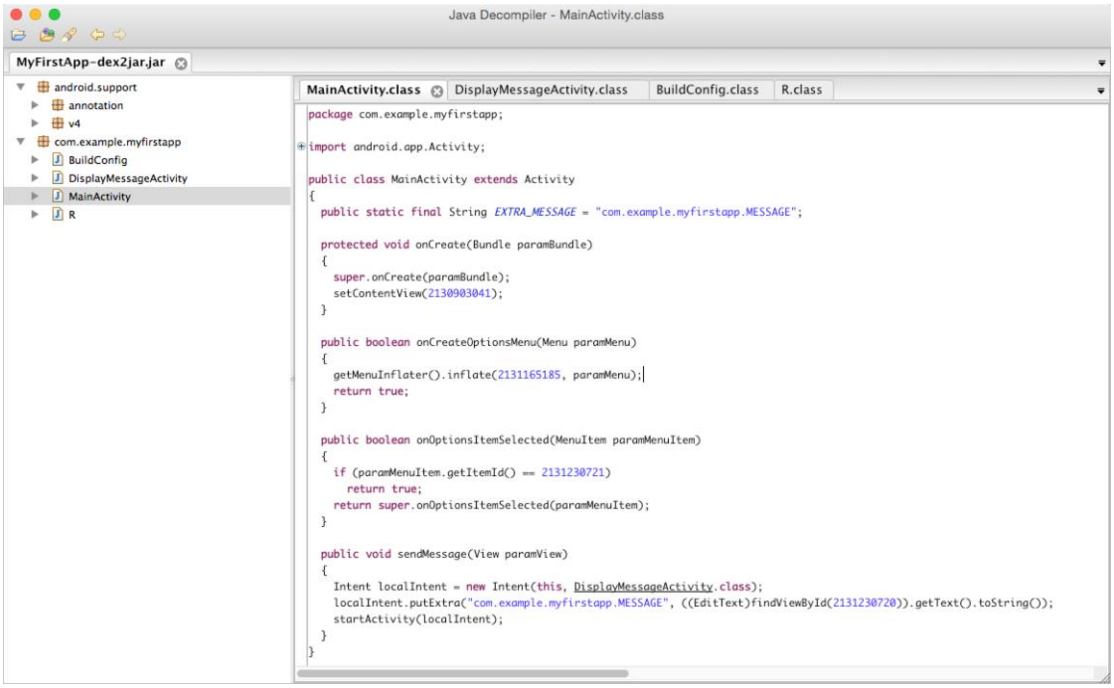
没有 without permission 的短信记录

实验分析：在加入权限之后，能够顺利浏览网页、发送短信，接受者的收件箱里面也能显示该短信。

而如果不加入权限，则无法访问网页，同时报错；短信在 Permission APP 上的手机显示发送成功，而接受者没有收到短信，同时，在发送者机器上面的短信发件箱也没有这条短信记录，综上，实验成功。

ReverseEngineering:

用 JD-GUI 打开由 dex2jar 逆向生成的 MyFirstApp-dex2jar.jar：



对比一开始我的 MyFirstApp 工程文件可以发现，代码除了那些数字不一样之外，其他的都是一样的，而那些数字可以从工程文件中 R.java 里面找到对应的值，所以结果和原来的是一致的。（变量名的 label 有所改变，但是不影响其功能实现。）

打开 MyFirstApp 的工程文件下的 AndroidManifest.xml （上）和由 apktool 生成的 AndroidManifest.xml（下）进行比较：

AndroidManifest.xml

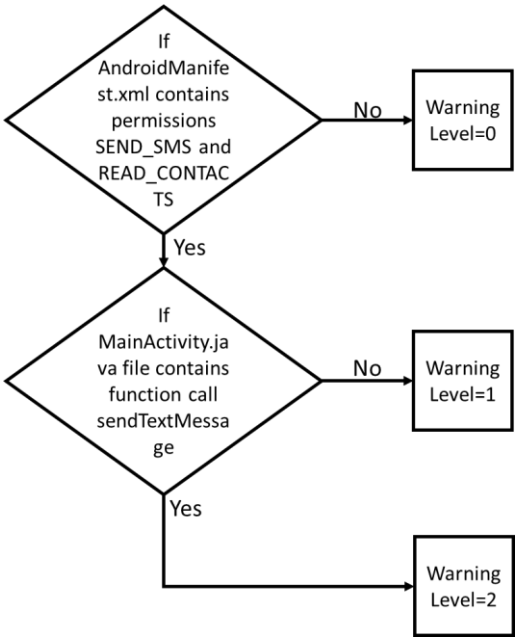
AndroidManifest.xml 〉 No Selection

```xml
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package=
    "com.example.myfirstapp">
    <application android:allowBackup="true" android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" android:theme="@style/AppTheme">
        <activity android:label="@string/app_name" android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:label="@string/title_activity_display_message"
            android:name=".DisplayMessageActivity"
            android:parentActivityName=".MainActivity">
            <meta-data android:name="android.support.PARENT_ACTIVITY" android:value=
                "com.example.myfirstapp.MainActivity"/>
        </activity>
    </application>
</manifest>
```

发现内容是一致的。

使用 auto_reverse 时，由于老师给的代码有问题，所以这里面不再使用老师给的代码，而是自己根据流程图写了一份：具体代码文件看 Auto_Reverse/src/Auto_Reverse.java

流程图：



对代码做几点解释：

0. 先访问 AndroidManifest.xml（命令行输入的第一个参数），判断是否同时含有 SEND_SMS 和 READ_CONTACTS

```
int flag = 0;
          for (int i=0;Line !=null;i++){
              if (Line.contains("android.permission.SEND_SMS")) flag = flag | 1;
              if (Line.contains("android.permission.READ_CONTACTS")) flag = flag | 2;
              Line = br.readLine();
          }
```

在这里使用位运算，同时都含有的时候 flag=3，然后流程图第一步 yes 当且仅当 flag=3

才可能往下执行，不然返回 warninglevel 为 0

1. 如果 flag=3，设置 warninglevel=1，那么就访问 MainActivity.java（命令行输入的第二个参数），判断里面是否含有 sendTextMessage，如果含有，设置 warninglevel=2，break
2. 最后输出 warninglevel，同时设置异常处理机制。

然后建立了测试文件（原型由 Permission 的 apk 逆向得到的）
其中

Androidfest0.xml 只含有 SEND_SMS

Androidfest1.xml 同时含有 SEND_SMS 和 READ_CONTACTS,但是 MainActivity1.java 中没有 sendTextMessage 函数

Androidfest2.xml 同时含有 SEND_SMS 和 READ_CONTACTS,MainActivity2.java 中含有 sendTextMessage 函数

```
Last login: Fri Dec  5 22:00:31 on ttys000
TroySmilENow:~ SmilENow$ cd /Users/SmilENow/学习/ZJU/2014-2015秋冬/courses/信息
系统安全\ 顾宗华/Lab/Lab2/2014-12-5-尹嘉权-3120000419/Auto_Reverse/src
TroySmilENow:src SmilENow$ javac Auto_Reverse.java
TroySmilENow:src SmilENow$ java Auto_Reverse ./AndroidManifest0.xml ./MainActivity0.java
The warning level for this AndroidManifest.xml file is:
0
TroySmilENow:src SmilENow$ java Auto_Reverse ./AndroidManifest1.xml ./MainActivity1.java
The warning level for this AndroidManifest.xml file is:
1
TroySmilENow:src SmilENow$ java Auto_Reverse ./AndroidManifest2.xml ./MainActivity2.java
The warning level for this AndroidManifest.xml file is:
2
TroySmilENow:src SmilENow$
```

符号实验要求！