[逆向工程： 1](#_Toc282623981)

[Protobuf 3](#_Toc282623982)

[android-market-api 4](#_Toc282623983)

[strazzere.com » Blog Archive » Downloading market applications without the vending app 4](#_Toc282623984)

# 逆向工程：

本文主要介绍如何逆向一个Android的APK应用程序，本文提供的方法仅供研究学习之用。

本文需要用到的工具有：

jdk 这个用于搭建java运行环境

AXMLPrinter2.jar 这个用于逆向.xml文件

baksmali.jar 这个用于逆向classex.dex文件

由于Android的.apk文件实际上就是一个zip文件 可以直接用winrar打开

用rar打开之后 我们可以看到该文件实际上是一个zip包 里面包含了META-INF文件夹，这个文件夹是用于保存签名文件，确保包的完整性的

res文件夹下就是apk所要用的资源文件，都是原封不动地保存，我们可以直接提取出来，做汉化时就可以直接阅读string文件然后进行修改

AndroidManifest.xml文件则是编译过后的一个配置文件，用于声明程序中所包含的activity,service以及程序所具有的能力，也就是权限。resources.arsc则是编译过后的一个资源说明文件，而我们要关注的主要是classes.dex 。我们编写的Android程序，在源程序里的所有.java的文件，最终都编译到这样1个.dex文件当中，在Android手机上的dalvik虚拟机上执行。

首先，我们介绍如何逆向一个.xml文件

由于apk包里的xml文件我们直接用记事本打开还是有一些乱码，所以需要我们还原才能更好的看出。这里需要用到AXMLPrinter2.jar 工具

具体的则是打开命令行 我们以AndroidManifest.xml为例，输入如下命令

**java -jar AXMLPrinter2.jar AndroidManifest.xml > AndroidManifest.txt**

有兴趣的也可以写成一个.bat的脚本，方便执行

接下来，大家肯定更加关心classes.dex的逆向，这个其实跟之前那个也很相似。采用baksmali.jar这个工具，国外一个对Android研究的很深入的大牛做的。执行代码

**java -jar baksmali.jar -o classout/ classes.dex**

讲classes.dex能逆向成一个文件夹。

Android由于其代码是放在dalvik虚拟机上的托管代码，所以能够很容易的将其反编译为我们可以识别的代码。

之前我写过一篇文章反编译Android的apk包到smali文件 然后再重新编译签名后打包实现篡改apk的功能。

最近又有一种新的方法来实现直接从Android apk包里的classes.dex文件,把dex码反编译到java的.class二进制码，然后从.class二进制码反编译到java源码。

首先我们需要的工具是dex2jar和jd-gui

其中第一个工具dex2jar是用来把classex.dex文件也就是dex二进制码转化为java的标准.class二进制码,然后jd-gui是把标准的.class二进制码再反编译为java源码。

首先从apk包里面提取出classes.dex，放到dex2jar目录下，然后在命令行下执行

**dex2jar.bat classes.dex**

这时执行成功后会生成一个classes.dex.dex2jar.jar文件。这个文件再用jd-gui打开，

## dex2jar

platform$ ./dex2jar.sh Android\_Market\_2.2.7.apk

Exception in thread "main" java.lang.UnsupportedClassVersionError: Bad version number in .class file

at java.lang.ClassLoader.defineClass1(Native Method)

at java.lang.ClassLoader.defineClass(ClassLoader.java:620)

at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:124)

at java.net.URLClassLoader.defineClass(URLClassLoader.java:260)

at java.net.URLClassLoader.access$100(URLClassLoader.java:56)

at java.net.URLClassLoader$1.run(URLClassLoader.java:195)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:188)

at java.lang.ClassLoader.loadClass(ClassLoader.java:306)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:268)

at java.lang.ClassLoader.loadClass(ClassLoader.java:251)

at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:319)

platform$ sudo update-alternatives --config javac

sudo update-alternatives --config java

支持1.6，不支持1.5。

## Assert

AssetTypeData

./sibrary/market/model/EncryptedSubscriberInfo.java:16: $assertionsDisabled = i;

./sibrary/market/model/EncryptedSubscriberInfo.java:55: if (!$assertionsDisabled)

./sibrary/market/model/RiskHeaderInfo.java:15: $assertionsDisabled = i;

./sibrary/market/model/AssetRequest.java:19: $assertionsDisabled = i;

./sibrary/market/model/AssetRequest.java:43: if (!$assertionsDisabled)

./sibrary/market/model/Address.java:16: $assertionsDisabled = i;

./sibrary/market/model/Comment.java:15: $assertionsDisabled = i;

./sibrary/market/model/Comment.java:29: if (!$assertionsDisabled)

./sibrary/market/model/ResponseProperties.java:16: $assertionsDisabled = i;

./sibrary/market/model/ResponseProperties.java:23: if (!$assertionsDisabled)

./sibrary/market/model/Asset.java:1631: $assertionsDisabled = i;

./sibrary/market/model/Asset.java:1665: if ($assertionsDisabled)

./sibrary/market/model/DeviceConfiguration.java:16: $assertionsDisabled = i;

./sibrary/market/AssetInfoActivity.java:2709: $assertionsDisabled = i;

./sibrary/market/AssetTypeData.java:24: $assertionsDisabled = i;

static {

if (!Comment.class.desiredAssertionStatus());

for (int i = 1; ; i = 0) {

$assertionsDisabled = i;

return;

}

}

static

{

if (!AssetTypeData.class.desiredAssertionStatus());

for (int i = 1; ; i = 0)

{

$assertionsDisabled = i;

sAssetTypeCount = 0;

Asset.AssetType localAssetType1 = Asset.AssetType.APPLICATION;

APPLICATION = new AssetTypeData(localAssetType1);

APPLICATION.registerResource(0, 2131099776);

...

}

}

简单的：

./sibrary/market/AssetInfoActivity.java:2722: assert (this.mAttached);

./sibrary/market/AssetInfoActivity.java:2757: assert (this.mAttached);

./sibrary/market/AssetTypeData.java:142: assert ((paramInt >= 0) && (paramInt < 4));

./sibrary/market/AssetTypeData.java:160: assert ((paramInt1 >= 0) && (paramInt1 < 4));

复杂的， 反编译器就不能完全处理（EncryptedSubscriberInfo.java、AssetRequest.java、Comment.java、ResponseProperties.java、Asset.java）：

if (!$assertionsDisabled)

{

ProtoBufType localProtoBufType1 = paramProtoBuf.getType();

ProtoBufType localProtoBufType2 = ApiDefsMessageTypes.ENCRYPTED\_SUBSCRIBER\_INFO\_PROTO;

if (!localProtoBufType1.equals(localProtoBufType2))

throw new AssertionError();

}

DeviceConfiguration、Address.java、RiskHeaderInfo.java比较奇怪， 没有适用到assert。 却有assertionsDisabled 声明。

可以验证的是：

assert (false); =〉

if (!$assertionsDisabled)

throw new AssertionError();

或者

if ($assertionsDisabled)

break;

throw new AssertionError();

assert (true); 就直接优化掉了。

assert ("ok".equals("Ok"));， assert (paramContext == null); 形式的可以正确反编译。

与第一种方法相比较，该方法生成的文件全面，代码整齐，虽然有些中文会出现乱码，id等全部是数字代替，但整体结构很完整，不错。

类似的工具还有Dedexer（http://sourceforge.net/projects/dedexer/files/） 具体的应用代码为（先进入jar包所在路径）：

java -jar ddx.jar -d <directory> <dex file>

可以再其中加入-o,这样可以生成一个介绍文件内部结构的日志文件。

网上还流传着这样一种用smali工具的方法:

反编译命令 java -jar baksmali-1.2.jar <dex文件> -o <新生成的文件名>

重新编译 java -jar smali-1.2.jar <需编译文件夹>

本人亲自实践了一下，确实新版本的baksmali可以直接对.apk文件进行直接的反编译。 但是这种方法生成的文件也不是很好。

说到这儿，对于xml的文件的编译，需要工具AXMLPrinter2.jar（http://code.google.com/p/android4me/downloads/list）。具体的使用方法，与上面的差不多，可以运用命令

java -jar AXMLPrinter2.jar main.xml > main.txt

进行相关的操作。大家可以参照这位仁兄的（http://hi.baidu.com/linrw/blog/item/cc3afd2233ab88a94723e878.html）

Android 逆向apk程序的心得

<http://blog.csdn.net/Zengyangtech/archive/2010/08/12/5807517.aspx>

Android 反编译apk 到java源码的方法

<http://blog.csdn.net/Zengyangtech/archive/2010/10/26/5967263.aspx>

Java Decompiler：<http://java.decompiler.free.fr/>

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

Dedexer：<http://sourceforge.net/projects/dedexer/files/>

smali： <http://code.google.com/p/smali/downloads/list>

AXMLPrinter2.jar：<http://code.google.com/p/android4me/downloads/list>

Android札记【3】——关于Android反编译方面[首届 Google 暑期大学生博客分享大赛]

<http://www.myext.cn/android/9613.html>

$java -jar AXMLPrinter2.jar Android\_Market\_AndroidManifest.xml > AndroidMarket.AndroidManifest.xml

android:configChanges="0x00000080">

./Browser/AndroidManifest.xml: android:configChanges="orientation|keyboardHidden">

/home/leo/android/platform/frameworks/base/api/current.xml

<field name="keyboardHidden"

....

core$ grep keyboardHidden ./res -R

./res/res/values/attrs\_manifest.xml: <flag name="keyboardHidden" value="0x0020" />

core$ pwd

/home/leo/android/platform/frameworks/base/core

# Protobuf

从Google protobuf工程把Java拷贝过来，不过工程的Java源码缺少com.google.protobuf.DescriptorProtos类。

这需要编译protobuf的C++实现，编译后有个protoc，generate\_descriptor\_proto.sh用它来产生DescriptorProtos。

$sudo aptitude install libtool

$./autogen.sh

$./configure

$make

$make check

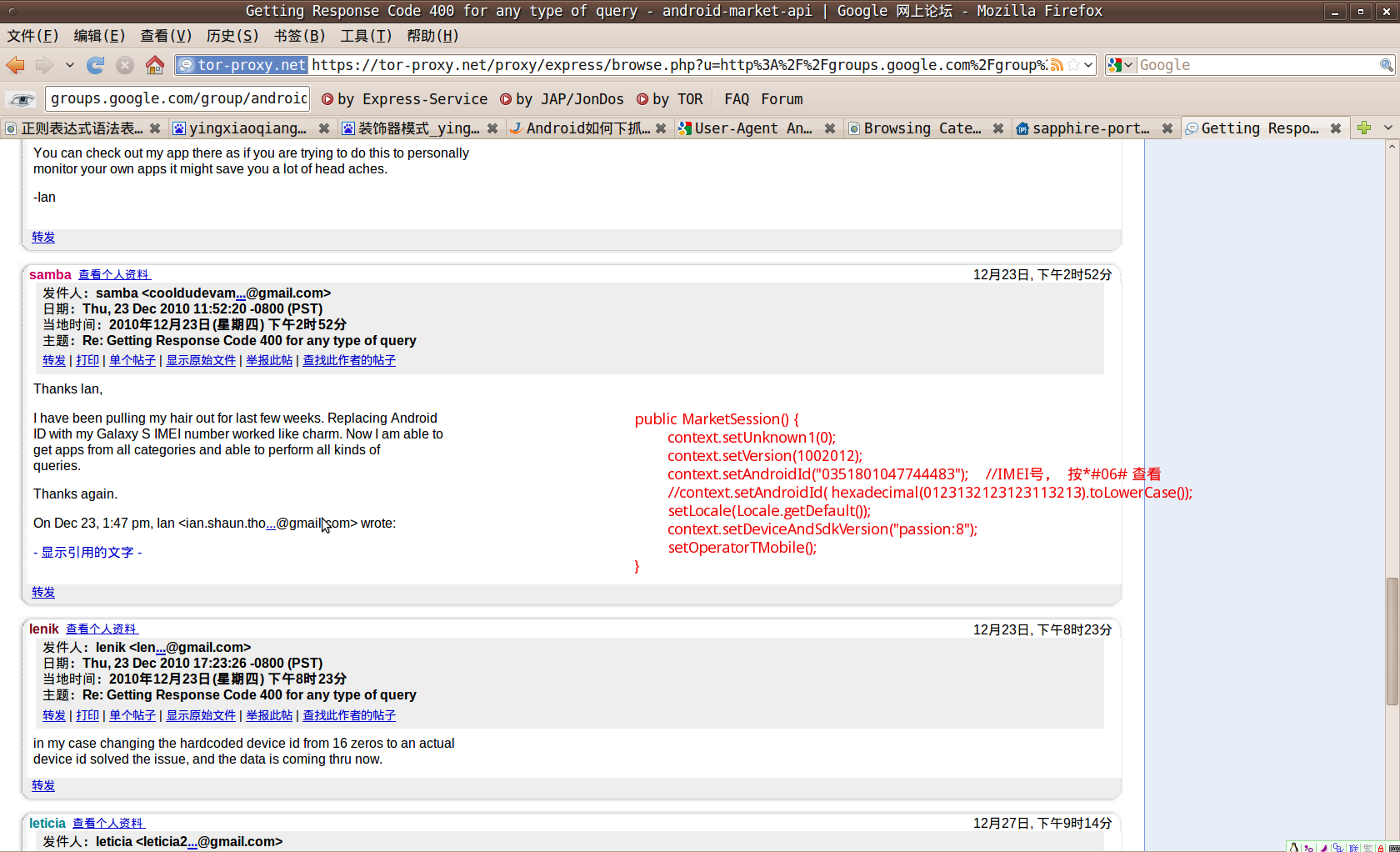
$sudo make install

$sudo ldconfig //update /etc/ld.so.cache

$cd java

$protoc --java\_out=src/main/java -I ../src ../src/google/protobuf/descriptor.proto

# android-market-api



C:\market-proto>protoc.exe market.proto --java\_out=C:\market-proto\out

# strazzere.com » Blog Archive » Downloading market applications without the vending app

MarketDownload

