

反编译工具的使用

实验概述

反编译就是逆过程，也称为计算机软件还原工程。用户通过将一个应用程序进行反编译，将会得到源代码。然后分析源代码，可以从中找出存在的各种漏洞。当用户找到某个软件的漏洞后，即可进行相应的渗透测试。

实验目的

- 1、学会如何使用apktool反编译apk
- 2、学会dex2jar和jd-gui结合使用查看apk源代码

实验原理

一个Android应用是一个数据的归档文件，并且源文件是在开发该程序是创建的。Android应用的扩展名为.apk。如果用户想查看该包中包括的文件，就必须解压缩该程序。

在应用程序包中，通常包括以下文件和文件夹。

Classes.dex（文件）：编译成apk之后的所有的类文件，包括所有的资源文件。

AndroidManifest.xml（文件）：开发人员在开发过程中，必须将应用程序中出现的组件，在此文件中申明

META-INF：（文件夹）：存储的签名的信息

resources.arsc：资源的索引表，里面维护着资源ID、Name、Path或者Value的对应关系

res：存放的可编译的资源文件

assets：存放的原生资源文件

apktool工具

apktool是一个apk反编译工具。该工具可是对.apk文件反编译，并且可以回编译。使用该工具对一个android应用程序进行反编译后，将会把.dex文件转化为.smali文件。反编译后生成所有的smali文件都被保存在smali文件夹中。该反编译方式的缺点是不能查看到apk的源代码，只能看到smali代码。

apktool反编译工具的语法格式：apktool d <file.apk>

apktool回编译工具的语法格式：apktool b <dir>

参数详解：

b：回编译

d：以调节模式解码

file.apk：需要反编译的apk

dir: 反编译之后的apk目录

✚ dex2jar、jd-gui工具

dex2jar可以将android的.dex文件转换成java的.class文件，用此工具时，需要将apk文件解压缩获得其dex文件。在linux下，把apk后缀改成zip后缀，使用命令“unzip file.zip -d <输出目录>”可以获得classes.dex文件。jd-gui是一图像界面的工具，它可以把.class文件反编译成java源代码。因此两个工具结合可以查看apk的源代码。

实验环境

虚拟机: kali

工具: apktool、dex2jar、jd-gui

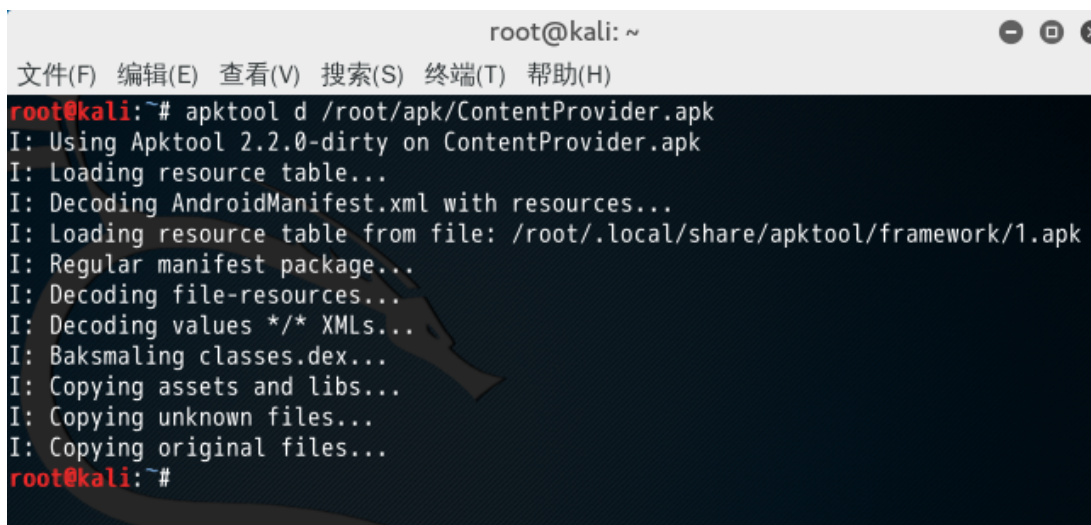
apk: ContentProvider.apk

模拟器: android 4.0

实验步骤

1、对实验apk进行反编译

使用命令: “apktool d /root/apk/ContentPorvider.apk”如图 1



```
root@kali: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
root@kali:~# apktool d /root/apk/ContentProvider.apk  
I: Using Apktool 2.2.0-dirty on ContentProvider.apk  
I: Loading resource table...  
I: Decoding AndroidManifest.xml with resources...  
I: Loading resource table from file: /root/.local/share/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...  
root@kali:~#
```

图 1反编译apk

2、编译完成后会在当前路径生成Content文件夹，查看apk的res资源文件

实验命令: “cd Content/res”>“ls”

```

root@kali:~# ls
android-sdk-linux  ContentProvider  公共  图片  音乐
apk                jd-gui_1.4.0-0_all.deb  模板  文档  桌面
bluedon.png       tools           视频  下载
root@kali:~# cd ContentProvider/
root@kali:~/ContentProvider# ls
AndroidManifest.xml  apktool.yml  original  res  smali
root@kali:~/ContentProvider# cd res
root@kali:~/ContentProvider/res# ls
drawable-hdpi-v4  drawable-xxhdpi-v4  menu-v11  values-v11
drawable-mdpi-v4  layout              values    values-v14
drawable-xhdpi-v4  menu                values-sw720dp-land-v13
root@kali:~/ContentProvider/res#

```

图 2查看资源

3、查看AndroidManifest.xml文件

使用命令：“cd /root/Content”>“cat Androidmanifest.xml”如图 3

```

root@kali:~/ContentProvider/res# cd ..
root@kali:~/ContentProvider# cat AndroidManifest.xml
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.isi.contentprovider" platformBuildVersionCode="19" platformBuildVersionName="4.4.2-1456859">
    <application android:allowBackup="true" android:debuggable="true" android:icon="@drawable/ic_launcher" android:label="@string/app_name" android:theme="@style/AppTheme">
        <activity android:label="@string/app_name" android:name="com.isi.contentprovider.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <provider android:authorities="com.isi.contentprovider.MyProvider" android:exported="true" android:name=".MyProvider"/>
    </application>
</manifest>
root@kali:~/ContentProvider#

```

图 3查看AndroidManifest.xml

4、把ContentProvider.apk后缀改成zip

使用命令：“cd /root/apk”>“ls”>“mv ContentProvider.apk ContentProvider.zip”>“ls”

```

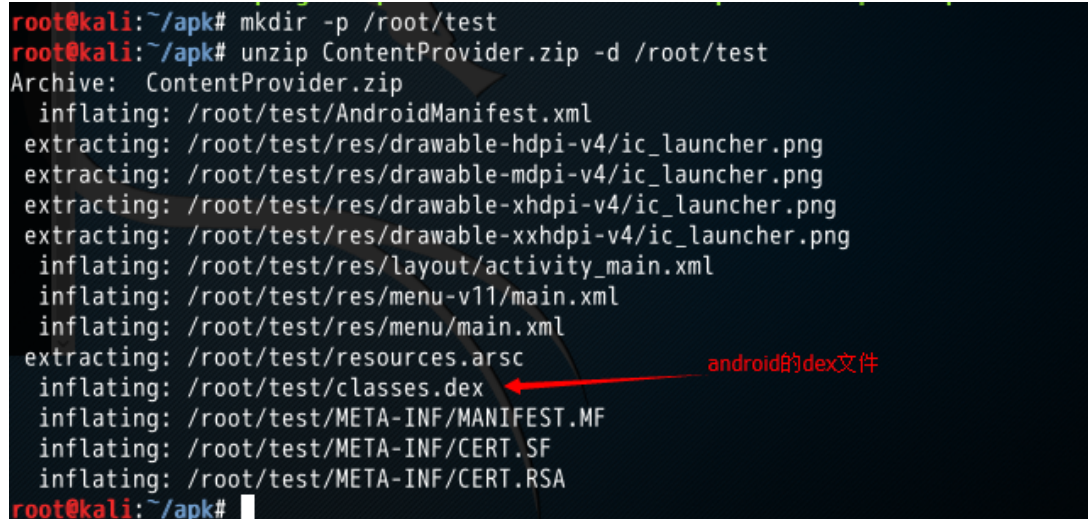
root@kali:~# cd /root/apk/
root@kali:~/apk# ls
AdobeReader.apk  drozer-agent-2.3.4.apk  QQ_410.apk  test.apk
ContentProvider.apk  game.apk  sieve.apk  vulnsqlite.apk
root@kali:~/apk# mv ContentProvider.apk ContentProvider.zip
root@kali:~/apk# ls
AdobeReader.apk  drozer-agent-2.3.4.apk  QQ_410.apk  test.apk
ContentProvider.zip  game.apk  sieve.apk  vulnsqlite.apk
root@kali:~/apk#

```


图 4改后缀

5、新建文件夹test，把ContentProvider.zip解压到test

使用命令：“mkdir -p /root/test”>“unzip ContentProvider.zip -d /root/test”如图 5



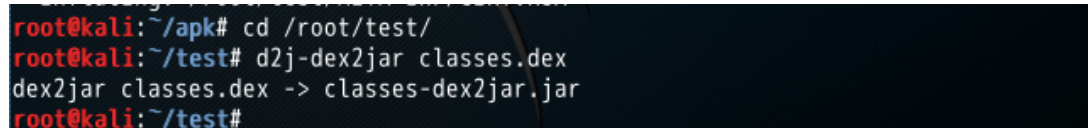
```

root@kali:~/apk# mkdir -p /root/test
root@kali:~/apk# unzip ContentProvider.zip -d /root/test
Archive: ContentProvider.zip
  inflating: /root/test/AndroidManifest.xml
  extracting: /root/test/res/drawable-hdpi-v4/ic_launcher.png
  extracting: /root/test/res/drawable-mdpi-v4/ic_launcher.png
  extracting: /root/test/res/drawable-xhdpi-v4/ic_launcher.png
  extracting: /root/test/res/drawable-xxhdpi-v4/ic_launcher.png
  inflating: /root/test/res/layout/activity_main.xml
  inflating: /root/test/res/menu-v11/main.xml
  inflating: /root/test/res/menu/main.xml
  extracting: /root/test/resources.arsc
  inflating: /root/test/classes.dex
  inflating: /root/test/META-INF/MANIFEST.MF
  inflating: /root/test/META-INF/CERT.SF
  inflating: /root/test/META-INF/CERT.RSA
root@kali:~/apk#
  
```

图 5解压apk

6、进入到test目录，将dex文件转换成jar文件

使用命令：“cd /root/test”>“d2j-dex2jar classes.dex”如图 6



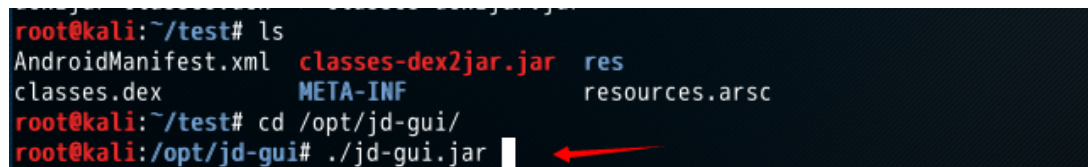
```

root@kali:~/apk# cd /root/test/
root@kali:~/test# d2j-dex2jar classes.dex
dex2jar classes.dex -> classes-dex2jar.jar
root@kali:~/test#
  
```

图 6转换为jar文件

7、启动jd-gui

使用命令：“cd /opt/jd-gui”>“./jd-gui.jar”如图 7



```

root@kali:~/test# ls
AndroidManifest.xml  classes-dex2jar.jar  res
classes.dex         META-INF             resources.arsc
root@kali:~/test# cd /opt/jd-gui/
root@kali:/opt/jd-gui# ./jd-gui.jar
  
```

图 7启动jd-gui

8、打开转换后的classes-dex2jar.jar

操作：“单击file”>“单击Open File”如图 8

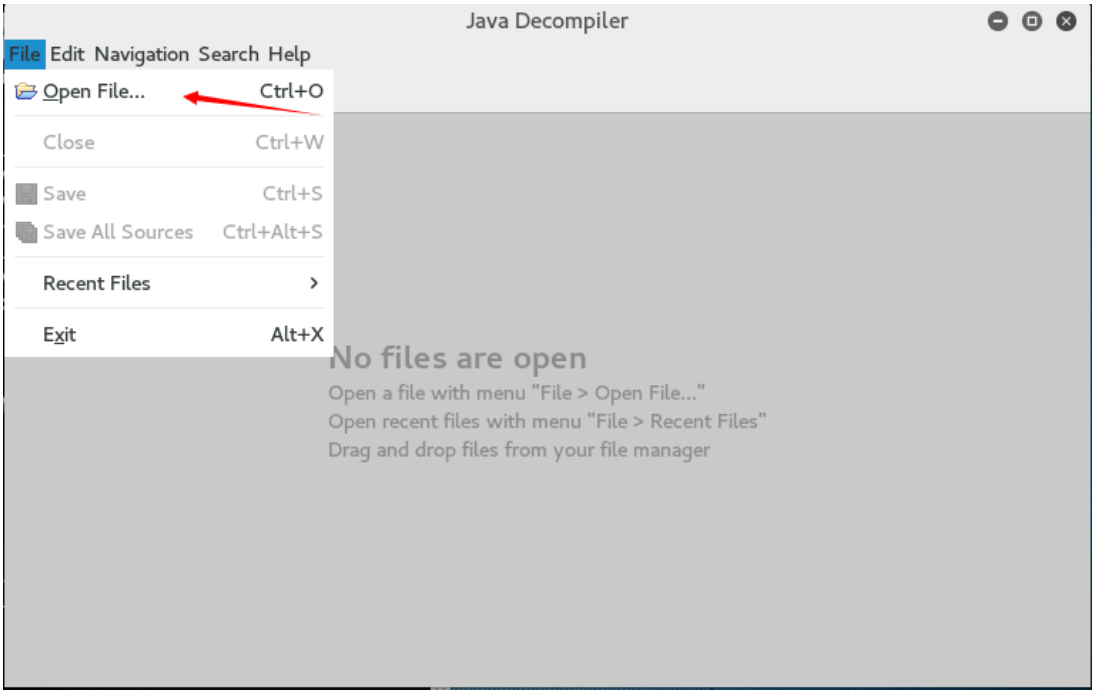


图 8打开jar文件

9、打开转换后的classes-dex2jar.jar

操作：“选定内容的文本框输入classes-dex2jar.jar文件的路径”>“单击确定”如图 9



图 9选中文件

10、成功看到apk源码，如图 10

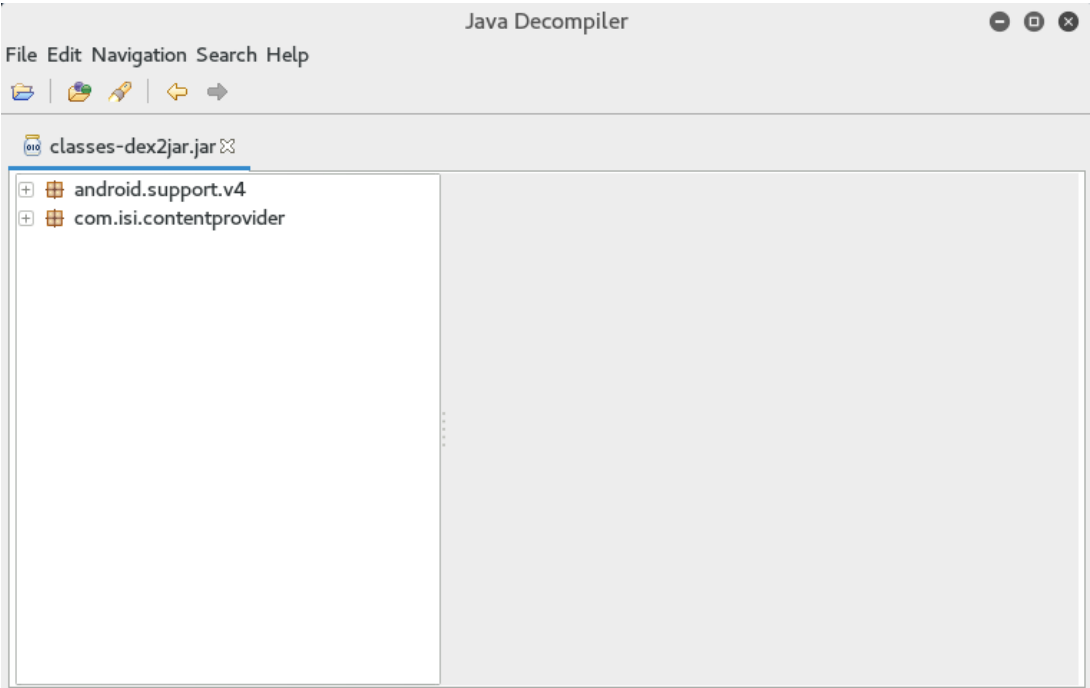


图 10 apk的java代码

11、查看activity和contentprovider组件源码。如图 11

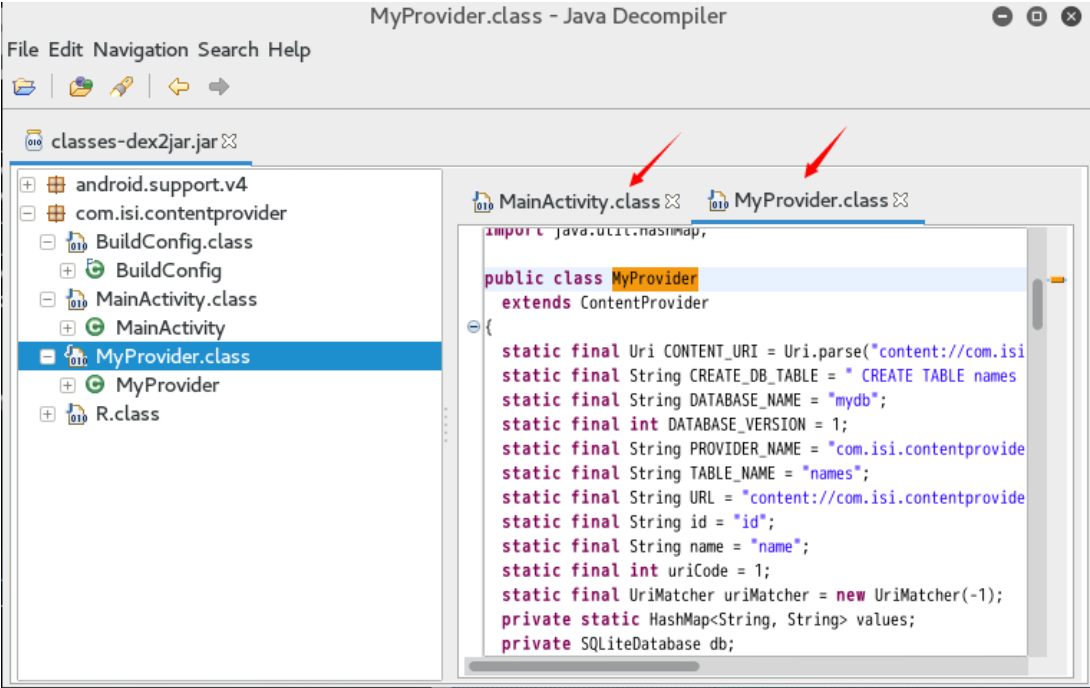


图 11 apk的java代码

思考总结

本实验介绍了两种反编译工具的使用方法，其中**apktool**反编译出来的是**smali**代码和相关资源，**dex2jar**、**jd-gui**反编译出来的是**apk**的源码。

- 1、除了实验介绍的工具，还有什么其它的反编译工具？
- 2、反编译之后，可以干那些事情？
- 3、**apk**的文件架构是怎样的？