



DOP: / /2023

DOS: / /2023

Experiment No:8

Title: Reversing Android applications (APKs) APKTOOL, dex2jar and JD-GUI.

Theory:

Reverse engineering

Reverse engineering an Android application typically involves using specialized tools to decompile the applications compiled code and resources into a human-readable form. As we go through this blog post, we will discuss the various available tools and examples of how they can be used to find hardcoded data and potentially find static application vulnerabilities.

- Understand how a particular UI in an App is constructed
- reading AndroidManifest.xml - permissions, activities, intents etc in the App
- native libraries and images used in that App
- obfuscated code (android SDK, by default, uses ProGuard tool which shrinks, optimizes, and obfuscates your code by removing unused code and renaming classes, fields, and methods with semantically obscure names.

Required Tools:

Download the followings first.

- Dex2jar from <http://code.google.com/p/dex2jar/>
- JD-GUI from <http://java.decompiler.free.fr/?q=jdgui>
- ApkTool from <http://code.google.com/p/android-apktool/>

Using ApkTool

- to extract AndroidManifest.xml and everything in res folder(layout xml files, images, htmls used on webview etc..)

Run the following command :

```
>apktool.bat d sampleApp.apk
```

It also extracts the .smali file of all .class files, but which is difficult to read.

You can achieve this by using zip utility like 7-zip.

Using dex2jar

- to generate .jar file from .apk file, we need JD-GUI to view the source code from this .jar.

Run the following command :

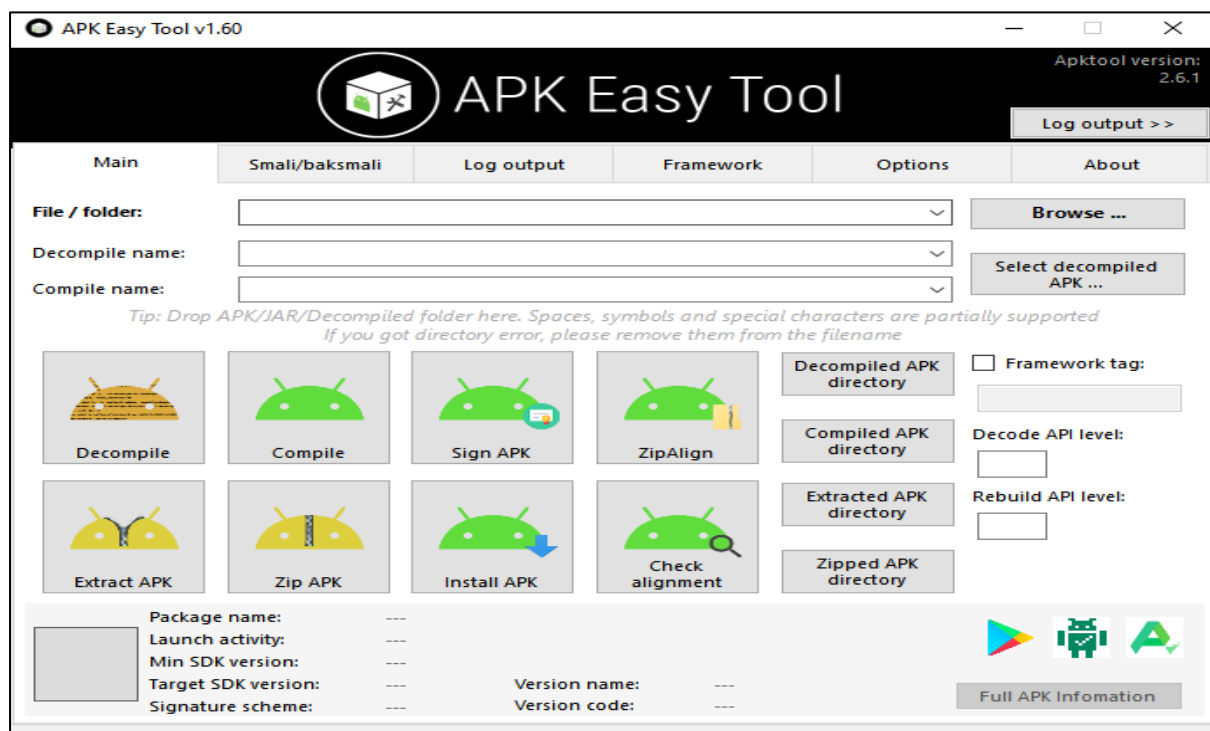
```
>dex2jar sampleApp.apk
```

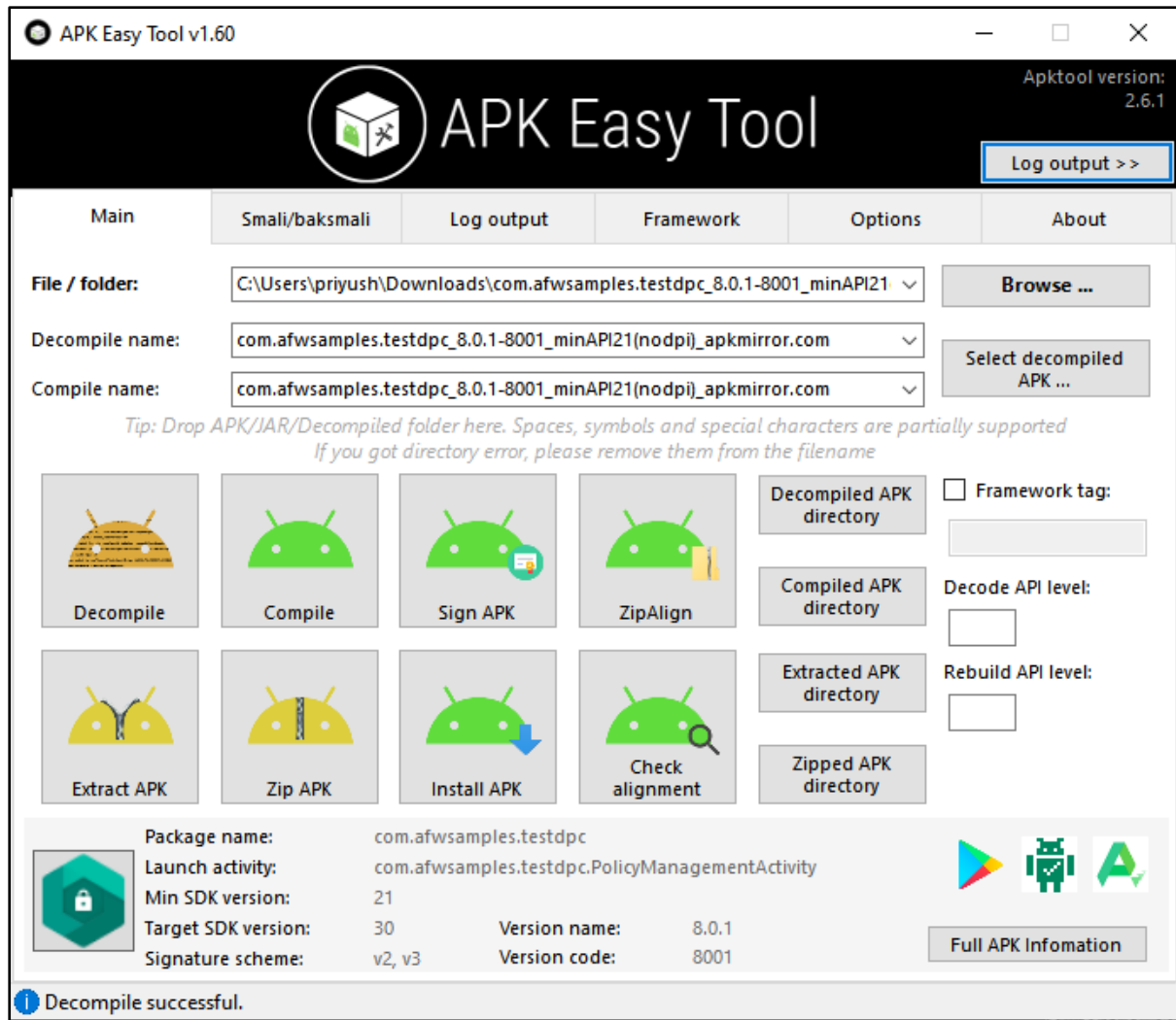
Decompiling .jar JD-GUI

it decompiles the .class files (obfuscated- in case of android app, but readable original code is obtained in case of other .jar file). i.e., we get .java back from the application.

Just Run the jd-gui.exe and File->Open to view java code from .jar or .class file.

Using ApkTool





Today (17)		
1-Decompiled APKs	06-04-2023 20:26	File folder
2-Recompiled APKs	06-04-2023 20:12	File folder
3-Extracted APKs	06-04-2023 20:12	File folder
4-Zipped APKs	06-04-2023 20:12	File folder
5-Baksmali	06-04-2023 20:12	File folder
6-Smali	06-04-2023 20:12	File folder
angular-1.8.2	06-04-2023 10:41	File folder
com	06-04-2023 17:58	File folder
layout	06-04-2023 17:57	File folder

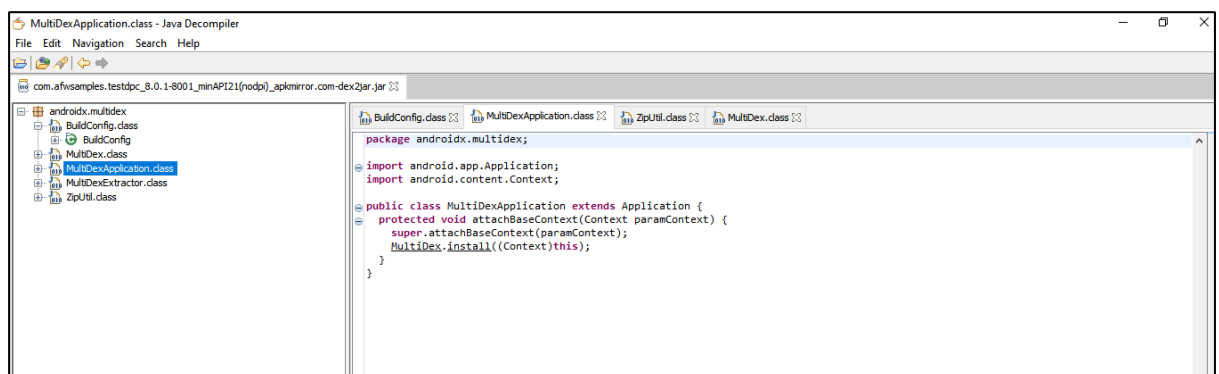
Using dex2jar

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\priyush\Downloads\ex07\dex2jar-2.0>d2j-dex2jar -h
d2j-dex2jar -- convert dex to jar
usage: d2j-dex2jar [options] <file0> [file1 ... fileN]
options:
  -d,--debug-info           translate debug info
  -e,--exception-file <file> detail exception file, default is $current_dir/[fi
                             le-name]-error.zip
  -f,--force                force overwrite
  -h,--help                 Print this help message
  -n,--not-handle-exception not handle any exception thrown by dex2jar
  -nc,--no-code             
  -o,--output <out-jar-file> output .jar file, default is $current_dir/[fi
                             le-name]-dex2jar.jar
  -os,--optimize-synchronized
  -p,--print-ir             print ir to System.out
  -r,--reuse-reg            reuse register while generate java .class file
  -s                        same with --topological-sort/-ts
  -ts,--topological-sort    sort block by topological, that will generate more
                             readable code, default enabled
version: reader-2.0, translator-2.0, ir-2.0

C:\Users\priyush\Downloads\ex07\dex2jar-2.0>d2j-dex2jar "com.afwsamples.testdpc_8.0.1-8001_minAPI21(nodpi)_apkmirror.com.apk"
dex2jar com.afwsamples.testdpc_8.0.1-8001_minAPI21(nodpi)_apkmirror.com.apk -> .\com.afwsamples.testdpc_8.0.1-8001_minAPI21(nodpi)_apkmirror.com-dex2jar.jar
C:\Users\priyush\Downloads\ex07\dex2jar-2.0>
```

Decompiling .jar JD-GUI



```
package androidx.multidex;

import android.app.Application;
import android.content.Context;

public class MultiDexApplication extends Application {
    protected void attachBaseContext(Context paramContext) {
        super.attachBaseContext(paramContext);
        MultiDex.install((Context)this);
    }
}
```

Conclusion: - Hence successfully performed Reversing Android applications (APKs) APKTOOL, dex2jar and JD-GUI.