Reverse Engineering and Secure Source Code Review

Reverse Engineering using APKTool, JADX

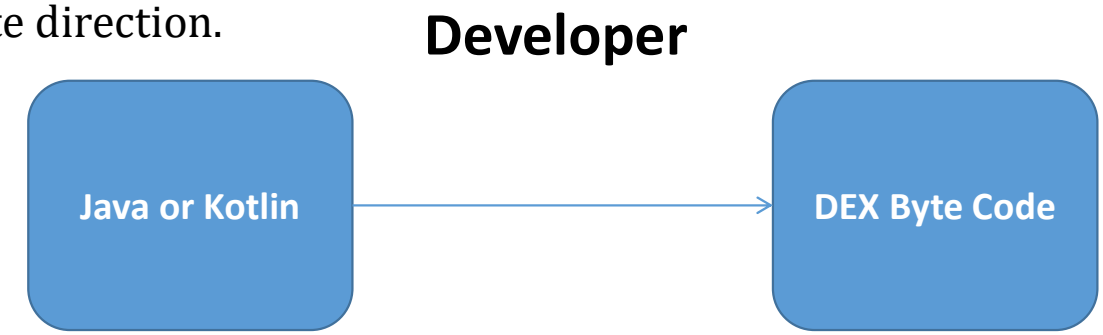
JD-GUI, HEX Dump, Dex Dump

Reversing and Auditing Android Apps

Android Application Teardown and Secure Source Code Review

Dalvik and Smali

- Most Android applications are written in Java. Kotlin is also supported and interoperable with Java.
- Instead of the Java code being run in Java Virtual Machine (JVM) like desktop applications, in Android, the Java is compiled to the Dalvik Executable (DEX) bytecode format.
- For earlier versions of Android, the bytecode was translated by the Dalvik virtual machine.
- For more recent versions of Android, the Android Runtime (ART) is used.
- If developers, write in Java and the code is compiled to DEX bytecode, to reverse engineer, we work the opposite direction.



Dalvik and Smali

- A disassembler is a computer program that translates machine language into assembly language—the inverse operation to that of an assembler.
- Disassembly, the output of a disassembler, is often formatted for human-readability rather than suitability for input to an assembler, making it principally a reverse-engineering tool.

Dalvik Smali

- Smali is the human readable version of Dalvik bytecode.
- Technically, Smali and baksmali are the name of the tools (assembler and disassembler, respectively), but in Android, we often use the term “Smali” to refer to instructions.

Reverse Engineer

- If you’ve done reverse engineering or computer architecture on compiled C/C++ code.
- SMALI is like the assembly language: between the higher level source code and the bytecode.
- The Smali instruction set is available
  - https://source.android.com/devices/tech/dalvik/dalvik-bytecode#instructions

Reverse Engineering using APKTool

- A tool for reverse engineering 3rd party, closed, binary Android apps.
- It can decode resources to nearly original form and rebuild them after making some modifications.
- It also makes working with an app easier because of the project like file structure and automation of some repetitive tasks like building apk, etc.
- It is NOT intended for piracy and other non-legal uses.
- It could be used for localizing, adding some features or support for custom platforms, analyzing applications and much more.

Reverse Engineering using APKTool

- **APK Tool Features**
  - Disassembling resources to nearly original form (including resources.arsc, classes.dex, and XMLs)
  - Rebuilding decoded resources back to binary APK/JAR
  - Organizing and handling APKs that depend on framework resources
  - Smali Debugging
  - Helping with repetitive tasks
- **Requirements**
  - Java 8 (JRE 1.8)
  - Basic knowledge of Android SDK, AAPT and smali
- **Authors**
  - Connor Tumbleson - Current Maintainer
  - Ryszard Wiśniewski - Original Creator

Reverse Engineering using APKTool

- **Steps to Install APK Tool in Windows**
  1. Download Windows apktool.bat
     - https://raw.githubusercontent.com/iBotPeaches/Apktool/master/scripts/windows/apktool.bat
  2. Download apktool
     - https://bitbucket.org/iBotPeaches/apktool/downloads/apktool_2.6.1.jar
  3. Move both files (apktool.jar & apktool.bat) to your Windows directory (Usually C://Windows)
  4. If you do not have access to C://Windows, you may place the two files anywhere then add that directory to your Environment Variables System PATH variable.
  5. Try running apktool via command prompt

- apktool d “D:\\Parag\myapp.apk" -o “D:\PCS\test"

Reverse Engineering using APKTool

- **Decompile with APK Tool**
  - Syntax
    - apktool d apkname.apk –o output_folder_name
  - Example
    - apktool d “D:\Parag\myapp.apk" -o “D:\Parag\test“
  - Apktool will create a new folder with the name test and place all the App data inside it.

Reverse Engineering using APKTool

- **Compiling APK from a Modified Source**
  - Compiling a modified source with apktool is as simple as decompiling.
    - apktool b <app_source_path>
    - apktool b facebook.apk

Disassembling DEX files

- Android's build process compiles Java source code to bytecode (.class file) and later converts it, along with resources, into .dex (Dalvik Executable a.k.a DEX) format to run efficiently on Android devices.
- **MyCode.java → MyCode.class → MyCode.dex**
- This allows you to create the Dalvik Virtual Machine bytecode from a dex file.
- Dexdump tools is used to perform following task
  - **Dalvik Virtual Machine Code ← Dex file**
  - This allows you to create the Dalvik Virtual Machine bytecode from a dex file.
  - To compile the Dalvik Bytecode to Java source code there are no official tools.

Disassembling DEX files – Dexdump

- Extract classes and methods from an APK file
  - dexdump [path/to/file.apk]
- Display header information of DEX files contained in an APK file
  - dexdump -f [path/to/file.apk]
- Display the dis-assembled output of executable sections
  - dexdump -d [path/to/file.apk]
- Output results to a file
  - dexdump -o [path/to/file] [path/to/file.apk]

Hexdump

- Hexdump helps you investigate the contents of binary files.
- Hexdump is a utility that displays the contents of binary files in hexadecimal, decimal, octal, or ASCII.

- It’s a utility for inspection and can be used for data recovery, reverse engineering, and programming.

Hexdump

- The hd or hexdump command in Linux is used to filter and display the specified files, or standard input in a human readable specified format.
- For example, if you want to view an executable code of a program, you can use hexdump to do so.
- Syntax:
  - hd [OPTIONS...] [FILES...]
  - -b : One-byte octal display.
  - -c : One-byte character display
  - -d : Two-byte decimal display
  - n length : Where length is an integer. Interprets only ‘length’ bytes of output.
  - -o: Two-byte octal display.

Dex2Jar

- Dex2Jar is a freely available tool to work with Android ".dex
- As you may aware that ".dex" files are compiled Android application code file.
- Android programs are compiled into ".dex" (Dalvik Executable) files, which are in turn zipped into a single ".apk" file on the device." and Java ".class" files.
- The ".dex" files can be created automatically by Android, by translating the compiled applications written in the Java programming language.
- The core feature of Dex2Jar is to convert the classes.dex file of an APK to classes.jar or vice versa.
- So, it is possible to view the source code of an Android application using any Java decompiler, and it is completely readable.

Dex2Jar

- Here, we get .class files and not the actual Java source code that was written by the application developer.
- it is possible to get ".smali" files directly from the classes.dex file or vice versa.
- That means you can change the source code of an application directly working with this format.
- Syntax
  - d2j-dex2jar –h
  - d2j-dex2jar –d filename.apk

JD GUI

- JD-GUI is a standalone graphical utility that displays Java source codes of ".class" files.
- You can browse the reconstructed source code with the JD-GUI for instant access to methods and fields.
- If you open the ".jar" file with JD-GUI, you can view the source code of the application which is Java classes in a readable format, and it is also very easy to navigate through the code.
- http://java-decompiler.github.io/
- Jd-gui classes_dex2jar.jar