Mobile Phone Security

Dr. Digvijaysinh Rathod
Associate Professor
(Cyber Security and Digital Forensics)
Institute of Forensic Science
National Forensic Sciences University

digvijay.rathod@gfsu.edu.in

De-Assembling and Reverse Engineering using APKTools and JDAX

Dr. Digvijaysinh Rathod

Dalvik & Smali

✓Most Android applications are written in Java. Kotlin is also supported and interoperable with Java.

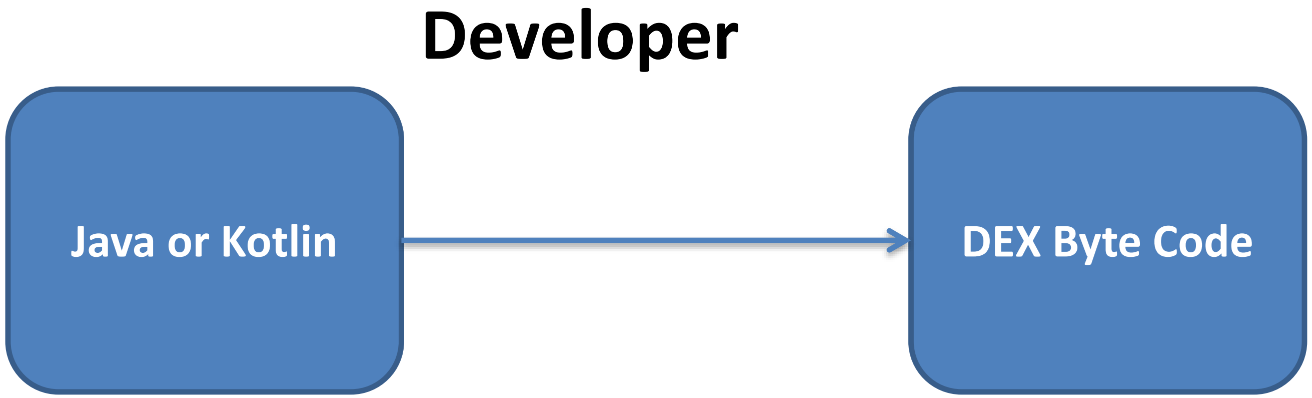
✓ Instead of the Java code being run in Java Virtual Machine (JVM) like desktop applications, in Android, the Java is compiled to the Dalvik Executable (DEX) bytecode format.

✓For earlier versions of Android, the bytecode was translated by the Dalvik virtual machine.

Dalvik & Smali

✓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.



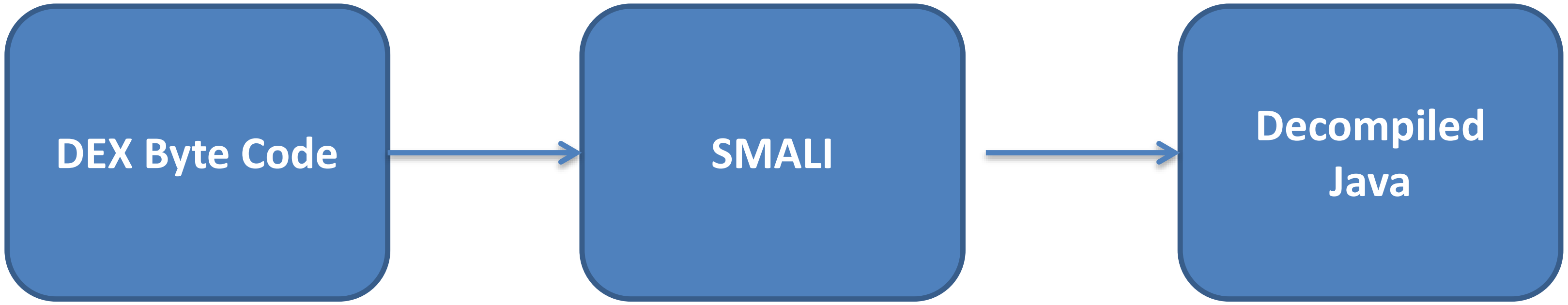
https://ragingrock.com/AndroidAppRE/app_fundamentals.html

Dalvik & 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

Reverse Engineer

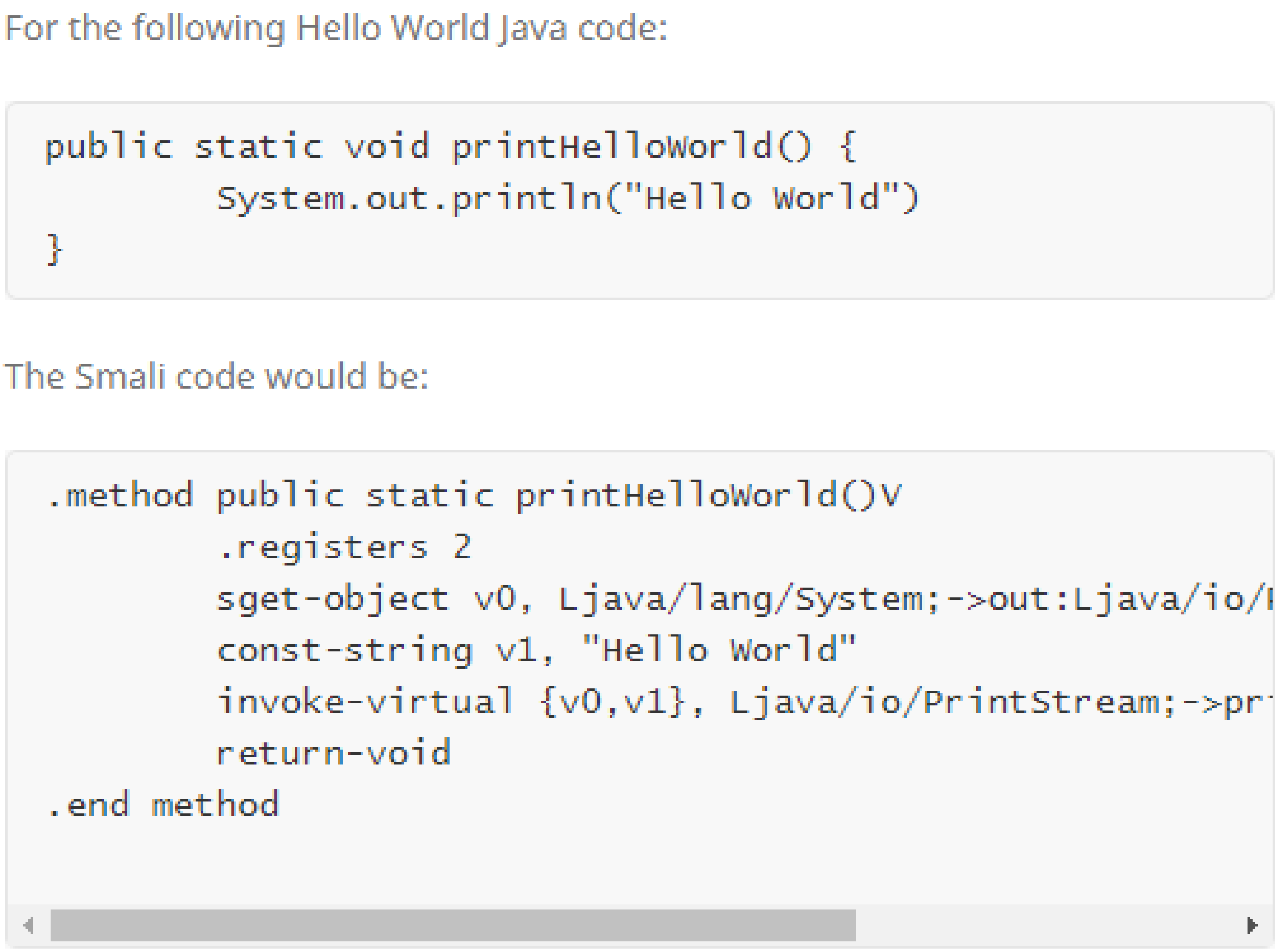


- ✓ Smali is the human readable version of Dalvik bytecode.
- ✓ Technically, Smali and baksmali are the name of the tools (assembler and disassembler, respectively), but in Android, we often use the term “Smali” to refer to instructions.

Dalvik & Smali

- ✓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

Dalvik & Smali



For the following Hello World Java code:

```
public static void printHelloWorld() {
        System.out.println("Hello World")
}
```

The Smali code would be:

```
.method public static printHelloWorld()V
        .registers 2
        sget-object v0, Ljava/lang/System;->out:Ljava/io/
        const-string v1, "Hello World"
        invoke-virtual {v0,v1}, Ljava/io/PrintStream;->pr
        return-void
.end method
```

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.

Ref: https://ibotpeaches.github.io/Apktool/

APKTool

✓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.

APKTool - Features

✓Disassembling resources to nearly original form (including resources.arsc, classes.dex, 9.png. 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

APKTool - Features

- ✓Requirements
  - ✓Java 8 (JRE 1.8)
  - ✓Basic knowledge of Android SDK, AAPT and smali
- ✓Authors
  - ✓Connor Tumbleson - Current Maintainer
  - ✓Ryszard Wiśniewski - Original Creator

Decompile with APKTool

✓apktool d <APK filename>

✓apktool **d** facebook_lite_v118.0.0.9.94.apk

✓Apktool will create a new folder with the same name as the APK file and place all the App data inside it.

Compile APK from a Modified Source

- ✓Compiling a modified source with apktool is as simple as decompiling.
  - ✓apktool b <app_source_path>
  - ✓apktool b facebook_lite_v118.0.0.9.94

Mobile Phone Security

Dr. Digvijaysinh Rathod
Associate Professor
(Cyber Security and Digital Forensics)
Institute of Forensic Science
Gujarat Forensic Sciences University

digvijay.rathod@gfsu.edu.in