android reverse engineering

Irit inbar 313332025

I worked with Android studio, apktool, dex2jar and jd-gui.
First I decompiled the original magicdate.apk with apktool. I read over the files to understand the actions happening in the app. To better understand, I used dex2jar and jd-gui to get the estimated java files from the smali files I got from decompiling.
While planning I thought of inserting the code under the random function, but the easier and more efficient way was to add a new file with my code, and change the existing files to work together.
I opened a new project in android studio to work on the malicious code and compiled it to apk.
The libraries that was used:
From android library:
App, bluetooth, content,
os.build, os.process, os.environment, os.usermanager,
provider.setting, util.log.
And java.io library for writing to file.
Using these libraries the app has no need for extra permissions only for notifications, and it collects information about the device hardware using the build and settings library, about media and bluetooth from process and environment libraries.

Now with the compiled apk i could use the smali code to add and change the files in magicdate.apk. Added the whole myaction.smali file and changed the package name to match magicdate. Also added 3 lines of code in magicdate.smali after the call for getRandom to call my function so both work together.

After checking that the app does not crash and the output file in its directory i recompiled the apk using apktool and signed with keytool.jar and jarsigned.

Files in github-
Apk-java2smali is the dummy apk used to get the smali code and java files are my source code. myaction.smali and magicdate.smali are the changed files, information.txt that contains the information pulled by the app.
On the recording the loading takes time because it after wiping the device data to start in a clean device.