Lab 4: ADB Commands

Overview

Android Debug Bridge (ADB) is a command-line tool that lets you communicate with an Android device over USB or the wireless network, to interact with the system and the applications for example transfer files from and into the Android System, so now we will learn the most important ADB commands

Steps

1. In the terminal navigate to labs > adb_commands folder
    - you will find APk File which is hack.apk you can use it for this lab or open the Hack project in Android Studio and switch to the adb_commands branch
      `git checkout adb_commands`
      and build a new debug app

2. Install the new version of the hack app
    - `adb install hack.apk`

3. ADB devices command
    - this command will show you the devices that are connected using ADB
    - `adb devices`
      the serial number of the connected device will be displayed

4. Show the device CPU architecture
    - you will need to know the device CPU architecture in some cases for example when you select the Frida server that matches your Android device CPU architecture
      `adb shell getprop ro.product.cpu.abilist`
      you will see a result similar to this: arm64-v8a or x86_64

5. Show the running processes
    - this command will show information about the running processes, it will show a list of the running processes but you can filter the result using findstr or grep
        - **for Windwos**
            - `adb shell ps | findstr hack`
        - **for Linux**
            - `adb shell ps | grep hack`

6. Shell commands

- adb enables us to enter the shell using this command
  `adb shell`
- now we can execute the shell commands directly for example
  `whoami` to print the username of the current user
  `pwd` to print the path of the current working directory
- exit the shell
  `exit`

7. Transfer files to an Android device

- create a new text file, name it test.txt, and push it to the download directory of Android using the push command

  ```
  adb push test.txt /sdcard/download
  ```

- retrieve the file to your machine with the name pull_test.txt using the pull command

  ```
  adb pull /sdcard/download/pull.txt pull_test.txt
  ```

8. List the installed applications

- list packages command will show the names of the packages for the installed applications and you can filter the result for only our hack app
  - **for Windows**
    - `adb shell pm list packages | findstr hack`
  - **for Linux**
    - `adb shell pm list packages | grep hack`

- the result is the package name of our app using the package name we can retrieve the APK path
  `adb shell pm path com.afif.hack`
  now we have the APK path
- using the APK path we can pull the APK
  `adb pull {path_of_APK}`
  the APK will be extracted to your machine with the name base.apk
- now uninstall the app
  `adb uninstall com.afif.hack`
- reinstall the base.apk
  `adb install -t base.apk`

9. SHow Logcat output
    - to show the Logcat output without Android Studio you can use adb logcat command, which takes the name of the tag, in our app AdbTag
      ```
      adb logcat -s AdbTag -v tag
      ```
    - now if you run the app, the log message will be printed