ANDROID LAB EXERCISE

Material Produced in the framework of:

DECAMP PROJECT

University of Padova
November 10, 2015
Contents

1 Activity 1: Creating an Android Virtual Machine .......................... 1
  1.1 Install VirtualBox ............................................. 1
  1.2 Download an Android-x86 ISO .................................. 1
  1.3 Creating an Android Virtual Machine .......................... 1

2 Activity 2: Installing Android Studio ........................................ 15
  2.1 Purpose ......................................................... 15
  2.2 Missing Java .................................................. 15
  2.3 Downloading Android Studio ................................... 15
  2.4 Linux Installation Instructions ............................... 15
  2.5 Using the Emulator ............................................ 19
  2.6 Starting your App .............................................. 21
  2.7 Source/Reference .............................................. 23

3 Activity 3: Connecting Android Virtual Machine to adb .................. 24
  3.1 Configure Network Settings .................................. 24
  3.2 Finding Android-x86 VM IP address .......................... 25
  3.3 Connect Android-x86 to adb - Commands ..................... 26

4 Activity 4: Installing Genymotion Virtual Machine ....................... 28
  4.1 Purpose ......................................................... 28
  4.2 Dowloanding Genymotion ...................................... 28
  4.3 Installing Genymotion ........................................ 28
  4.4 Configuring Genymotion Settings ............................ 29
  4.5 Starting the Genymotion Device ............................. 30
  4.6 Installing Play Store ......................................... 32

5 Activity 5: Android Security Auditing with Genymotion Virtual Machine and BurpSuite Proxy ............................ 41
  5.1 Purpose ......................................................... 41
  5.2 Finding your Host Computer’s IP Address ...................... 41
  5.3 Installing Burp Proxy ......................................... 41
  5.4 Configuring Burp Proxy ....................................... 42
  5.5 Adjusting Android Networking ................................ 47
  5.6 Testing the Proxy .............................................. 49
  5.7 Opening a Secure Page ....................................... 49
  5.8 Exporting the PortSwigger CA Certificate from Burp ........... 51
  5.9 Installing the PortSwigger CA Certificate into Android ...... 52
  5.10 Intercepting HTTPS Traffic (Passive MITM attack) ........... 55
  5.11 Viewing HTTPS Requests in Burp ............................. 56

6 Activity 6: Capturing and Monitoring Android Network Traffic ........... 57
  6.1 Purpose ......................................................... 57
  6.2 Install Wireshark .............................................. 57
  6.3 Install tcpdump on real smart-phone .......................... 57
  6.4 Quick Wireshark Guide ....................................... 57
  6.4.1 Start Wireshark .......................................... 58
1 Activity 1: Creating an Android Virtual Machine

What you need for this experiment

Before you can get started, you will need both VirtualBox and an Android-x86 ISO to install inside VirtualBox.

1.1 Install VirtualBox

Download and install VirtualBox if you do not already have it installed by opening up the website: https://www.virtualbox.org/.
In Ubuntu machine alternatively you can open up a terminal (ctrl+alt+T) and then type in: sudo apt-get install virtualbox

1.2 Download an Android-x86 ISO

You can find the latest Androidx86 ISO files at: http://www.android-discretionary{-}{}{}x86.org/download
Download this ISO file: android-x86-4.4-r1.iso (360 Mb).

1.3 Creating an Android Virtual Machine

-Power on your new virtual machine by opening an Ubuntu terminal (ctrl+alt+T) and then type in: virtualbox
-Point VirtualBox at the Android-x86 ISO file you downloaded and create a new virtual machine by clicking the “New” button. Go through the wizard and configure your virtual machine with the following settings:
-Operating System: Linux 2.6/ 3X

Figure 1: Name and OS
- Memory Size: 512 MB

Figure 2: Memory size

- Hard Drive Size: 3 GB or More

Figure 3: Hard drive
Figure 4: Hard drive file type

Figure 5: Storage on physical hard drive
-In the next step, the installer will boot up inside your virtual machine. Select the Installation option and press Enter to continue.
-Select the “Create/Modify partitions” option and press Enter to create a partition for the Android system.

-Select the New option with **Arrow Keys** and press Enter to create a new partition.
Choose the Primary option and press Enter, then press Enter again, the partition takes up the entire size of the virtual disk you created.

Highlight the Bootable option and press Enter to make the partition bootable.
Select the Write option and press Enter to write your changes to the virtual disk. Then, type yes and press Enter to confirm writing the changes.

- Activate the Quit option to go back to the Choose Partition screen.
-Select the partition you created and press Enter to install Android to it.

-Select the ext3 file system, and then select “Yes” to format the partition.
Choose file system

Choose Yes to install the GRUB boot loader

Select Yes to install the /system directory as read-write.
Once you are done, remove the ISO file and reboot your virtual machine, according to below picture:
After rebooting Android-x86, you can run it without installation as follows:

**Figure 23: Running Android-x86**

*Note:* go through the set up process and optionally provide Google account details. Skip the WiFi set up screen. If your mouse cursor does not work, click the Machine menu and select the Disable Mouse Integration option. Click inside the virtual machine to see the mouse cursor. Press “Right Ctrl” to release the mouse cursor from the virtual machine.
Figure 24: Android-x86 Welcome Screen

Figure 25: Android-x86 Home Screen
If your Android home screen becomes idle and then dark close your virtual machine, a window pops up and then select “send shut down signal” instead of “power off the machine”.

Note: if your mouse cursor does not work, click the Machine menu and select the Disable Mouse Integration option. Click inside the virtual machine and you will see the mouse cursor. Press the host key displayed at the bottom-right corner of the virtual machine window (default key: Right Ctrl) to release the mouse cursor from the virtual machine.
Figure 28: Android-x86 Home Screen

Figure 29: Disable Mouse Integration
2 Activity 2: Installing Android Studio

What You Need for This experiment

Any computer, running any OS. (we use Ubuntu 14.04), a high-speed Internet connection.

2.1 Purpose

Android Studio is Google’s software development environment for Android. It is essential for creating, debugging, and testing Android apps.

2.2 Missing Java

First make sure you have the latest Java JDK installed - at least JDK 6. If you don’t have Java installed visit: http://www.oracle.com/technetwork/java/javase/downloads/index.html
Or, install JDK as bellows:

```
sudo apt-get install openjdk-7-jdk
```

Environment Variable:

```
sudo nano /etc/environment
```

Adding the following line:

```
JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-i386
```

and Reboot.

To check: `java -version`

2.3 Downloading Android Studio

Follow the instructions on the Web page to download and install Android Studio for your OS. Go to http://developer.android.com/sdk/index.html. The Linux version is 350 MB in size.

2.4 Linux Installation Instructions

1. Unpack the Android Studio distribution archive that you downloaded to where you wish to install the program. We will refer to this destination location as your installation home.
2. Open a console and cd into “/home/user/Downloads/android-studio/bin” and type: ./studio.sh
Accept any defaults that the set up program offers you. It installs not only Android Studio, but the SDK, the virtual device and if you are using an Intel based machine a virtual device accelerator.
Figure 32: pica

Figure 33: Android-studio verify settings
Open up Android-studio by opening a console and cd into “/home/user/Downloads/android-studio/bin” and type:

```
./studio.sh
```

as shown below:

Figure 34: Android-studio downloading components

Open up a web browser and go to https://github.com/securitycompass/AndroidLabs and then download Zip file named “AndroidLabs-master.zip”. Unzip the file in somewhere in your host system.

Figure 35: Android-studio welcome page
Click Import Project (Eclipse ADT, Gradle ...) and navigate to AndroidLabs-master directory then click OK. Click on the project (needs API-21)

![Figure 36: Importing project](image1.png)

Figure 36: Importing project

![Figure 37: Android-studio Side Menu](image2.png)

Figure 37: Android-studio Side Menu

2.5 Using the Emulator

The next step is to run your app on an emulated Android device. From the Android Studio menu bar, click Tools, Android, “AVD Manager”, as shown below:
Figure 38: Creating a virtual device

Chose a device definition and click Next twice and finish. Let’s AVD be saved.

Figure 39: Selecting Hardware
2.6 Starting your App

From the Android Studio menu bar, click Run, “Run app”.

A box pops up, asking you to "Choose a running device", as shown below:
Select your virtual device and click OK. Wait until virtual device to be opened. As shown below:
2.7 Source/Reference

https://samsclass.info/128/proj/p3-astud.htm
### 3.1 Configure Network Settings

To connect Android virtual machine to adb, run the VirtualBox, click on Android virtual machine (Android-4.4.1).

![Android-x86 Network setting (1)](image)

Click on Setting menu
Go to Network tab.
Select Adapter 1 sub-tub and configure it as shown below:
Attach adaptor 1 to Bridged Adaptor and then change the name section to "wlan0" or "eth0" according to your Internet connection. (e.g., if you are connected to a LAN select eth0).

3.2 Finding Android-x86 VM IP address

While the cursor is inside the Android Virtual Machine home screen push Alt+F1 and see your IP address.
In command prompt type: `netcfg`
Take a note of the IP address. To exit push Alt+F7.
3.3 Connect Android-x86 to adb - Commands

Open up an Ubuntu terminal and type:

```
adb connect YOUR-IP-ADDRESS:5555
```

In this case:
```
adb connect 192.168.1.105:5555
```

Then, you will be connected to Android Virtual Machine. To check type:

```
adb devices
```

You should see below message:

```
hossein@hossein:~
```
```
adb devices
List of devices attached
192.168.1.105:5555 device
```

Now you can Install, Uninstall Android application, Push and Pull files using adb commands.
**Note:** if adb is not running or responding you can do following:

```bash
adb kill-server
adb start-server
```

**Source/Reference**

4 Activity 4: Installing Genymotion Virtual Machine

What you need for this experiment

A computer, Mac, Ubuntu or Windows OS with Java, Android-Studio, and VirtualBox installed. (In this activity, we use Ubuntu 14.04), Genymotion VM, Internet Connection.

4.1 Purpose

To get Genymotion working, install Android apps. There are many reasons to use Genymotion. First of all, it is lightning fast and not as slow as the android emulators. Secondly, it is a rooted emulator unlike the android emulators. This means you have much more freedom of installing your own custom applications that can be used for auditing other Android apps.

4.2 Downloading Genymotion

-Open up a web browser, go to: https://www.genymotion.com/#!
-Click “Get Genymotion”. Scroll to the bottom of the “Free” column and click Download.
-Select your OS and then Download Genymotion according to your Operating System. A “Sign in” box pops up.
-Click the “Create Account” button. Build a username and password, and enter a valid email account.
-Accept the terms and click “Create account”. Click the link in your email. Once your account is working, click Continue to download genymotion.

4.3 Installing Genymotion

-Open up a terminal in Ubuntu (ctrl+alt+T)
-Change the directory to the place you download Genymotion (e.g., cd /home/user/Downloads)
-Type in terminal: chmod u+x genymotion-2.5.2_x64.bin
You will see this message:

| Installing to folder [/home/user/Downloads/genymotion] |
| Are you sure [y/n] ? |
| Then type y. |

-Change the directory to genymotion folder (cd /genymotion)
-Type in: ./genymotion
-It is better to install Genymotion with the default options. After it starts, Genymotion asks whether you want to add a new virtual device. Click “Yes”.
-Sign in with your Genymotion account to see available virtual devices.
Select a devices (e.g., Google Galaxy Nexus 5 -5.0.0 - API 21 - 1080x1920).
-Click Next twice. The virtual device starts downloading.
-Click Finish.

Note: you can download as many virtual devices as you want.
4.4 Configuring Genymotion Settings

In the Genymotion window, click Settings. In Settings, on the General tab, insert your Genymotion username and password, as shown below:

![Figure 49: Configuring Genymotion Settings - Account](image)

On the ADB tab, fill in the SDK Path for your host system, as shown below:

![Figure 50: Configuring Genymotion Settings - ADB](image)

**Note:** To find your SDK path, open Android Studio and click Tools, Android, “SDK Manager”. Click OK. The path should be like this: /home/user/Android/Sdk/platform-tools
- Adding sdk path to your environment

Open up an Ubuntu terminal and type in:

```
PATH=/home/hossein/Android/Sdk/platform-tools:$PATH
export PATH
```

**Note:** the Network tab here is used to configure an interface that is used only for Genymotion updates, not for the Internet traffic to the Android device. Just ignore it.

### 4.5 Starting the Genymotion Device

Open up the Genymotion emulator and in the Genymotion window, click Start. A window pops up and you see initializing and starting of virtual device.

![Genymotion virtual devices](image)

Figure 51: Genymotion virtual devices

Also, you can add several virtual devices by pushing “add” button.
Figure 52: Select a new virtual device

After you start the virtual device, it will be initializing as the following pictures:

Figure 53: Genymotion Home Screen
4.6 Installing Play Store

By default, Genymotion does not provide Google Apps and to get Play Store on the Genymotion device, you need to install two files.

1. On your host computer, in a Web browser, Download below zip file: - ARM Translation Installer v1.1 available at: [http://filetrip.net/dl?4SUOrdcMRv](http://filetrip.net/dl?4SUOrdcMRv) (Or from repository provided for you). After the download completes, drag the file and drop it onto the Genymotion Android home screen. A dialogue will appear and show a file transfer in progress.
Then, another dialogue will appear and ask if you want to flash it on the emulator.
Click OK.
A box will appear telling you to reboot the device. Rebooting the Android Device with ADB on your host machine as follows:
Open a Terminal (e.g., cd /home/user/Android/Sdk/platform-tools) -type in: ./adb devices -l
Or simply you can close your virtual device and restart it.

2. On your host computer, in a Web browser, download this file: Google Apps for your Android version: 2.3.7 - 4.4.4 or 5.0 available at: http://wiki.cyanogenmod.org/w/Google_Apps Or https://www.androidfilehost.com/?fid=957848910016145590. After the download completes, repeat instructions the same steps as above. If the above procedures failed re-do it again.

Once you rebooted the Genymotion emulator, the Google Apps (Google+) will be in the emulator.

Figure 57: Installing Play Store (3)
At this point Google Apps Services will crash frequently with the following message google play services has stopped working.
Figure 60: Installing Play Store (6)

Figure 61: Installing Play Store (7)
Close the Genymotion emulator and restart it again. Open Google+ and after providing your account details, update Google+ then update Google Play. Now you can launch the Play Store app to download and install apps on your virtual device.

![Add your account](image)

Figure 62: Adding Google Account
Figure 63: Updating Google+

Figure 64: Updating Google Play services
Note: if you want install local Android app on emulator, To allow install third-party apps, in Genymotion emulator go: Settings then Security and check mark Unknown resources.

Figure 65: Activating third-party app installation

In Ubuntu terminal (ctrl+alt+T) type in:
adb install android.apk

Figure 66: Installing app via adb (1)
Enjoy the speed of Genymotion and thanks to CPU and GPU hardware acceleration! Start your apps two to three times faster than Android-x86 Virtual Machine.
5 Activity 5: Android Security Auditing with Genymotion Virtual Machine and BurpSuite Proxy

5.1 Purpose
To get Genymotion and Burp working, install Android apps, and audit their security.

5.2 Finding your Host Computer’s IP Address
- In your host system, open a Terminal or Command Prompt.
- In Ubuntu or Mac execute the “ifconfig” command.
- In Windows execute the “ifconfig” command.
Find the IP address of the interface that goes to the Internet and make a note of it.

5.3 Installing Burp Proxy
Burp is a proxy software, enabling you to sniff and alter network traffic. In a web browser, go to http://portswigger.net/burp/download.html. At the bottom of the “Free Edition” column, click “Download now”. A “burpsuite_free_v1.6.jar” file is downloaded.
In Ubuntu open up a terminal and change the directory where the downloaded file is.
Type in: cd /home/user/Downloads
java -jar burpsuite_free_v1.6.jar
burp opens as shown below:
In Burp, click the Proxy tab. Click the Intercept sub-tab.
Click the "Intercept is on" button, so it shows "Intercept is off", as shown below:
Burp opens, as shown below. Click the Target tab.

Click the Options sub-tab. If any listeners are running, clear their check-boxes to stop them, as shown below.
In Proxy Listeners, click the “Add” button. In the “Add a new listener” box, in the “Bind to port” field, enter 8080. In the “Specific address” list box, select your host computer’s IP address that you determined earlier, as shown below.
Click OK.
Burp should now show a listener on your IP address and port 8080, as shown below.
Click the Target tab. Click the Scope sub-tab. In the "Include in scope" section, clear all the check boxes if exist, Click the Add button. In the Add in “include in scope” box, click OK. Burp now shows a scope of “any”, as shown below:
5.5 Adjusting Android Networking

Open up the Genymotion emulator and click Settings. In Settings, click “Wi-Fi”.

Click and hold WiredSSID (a few second) until a box pops up.
Click on “Modify network”.

Check the “Show advanced options” box and select Manual from the Proxy Settings menu. Enter your host machine’s IP address in the “Proxy hostname” field, and 8080 in the “proxy port” field, as shown below. Then click “Save”.
5.6 Testing the Proxy

In the Android device, open Browser and go to one website (e.g., http://krpano.com/examples/). In Burp, click the Proxy tab.
Click the “HTTP history” sub-tab.
Find the line that shows the http://krpano.com page loading.

5.7 Opening a Secure Page

In the Android device, open Browser and go to https://google.com A “Security warning” box pops up, as shown below:
Click “View certificate”. This certificate is from “PortSwigger”, the makers of Burp, as shown below:
5.8 Exporting the PortSwigger CA Certificate from Burp

This is HTTPS working as it should, warning you that you do not have a secure connection to the end site. Burp is intercepting the traffic. We want to add PortSwigger as a trusted certificate authority to get rid of these messages.

In Burp, click the Proxy tab. Click the Options sub-tab. Click the “CA certificate...” button.

![Figure 81: Exporting CA certificate (1)](image)

In the “CA Certificate” box, click the ”Certificate in DER format” button, as shown below:

![Figure 82: Exporting CA certificate (2)](image)

Click Next. Save the certificate somewhere you can find it, such as on your Desktop. Give it a name of portswigger.cer, as shown below:
Click Save. Click Next. Click Close.

5.9 Installing the PortSwigger CA Certificate into Android

Drag the portswigger.cer file and drop it on the Android home page. A message appears, saying “Files successfully copied to sdcard/Download”.

Click OK. From the Android home screen, Click Settings. In Settings, scroll down and click Security. In Security, scroll down and click “Install from SD card”, as shown below:
Figure 84: Installing CA certificate on phone (1)

- Select internal storage then Download.
A box pops up titled “Name the certificate”.

53
Click OK.
If you have a PIN configured, you will have to enter it.
A box pops up, saying "Portswigger is installed".

### 5.10 Intercepting HTTPS Traffic (Passive MITM attack)

In Android, launch the browser. Go to a secure page like: https://gmail.com
The page should load without any error messages, as shown below:
5.11 Viewing HTTPS Requests in Burp

In Burp, click the Proxy tab. Click the “HTTP history” sub-tab. Find the line that shows the https://gmail.com page loading, as shown below:

![Figure 89: Viewing HTTPS traffic](image_url)

Source/Reference

What you need for this experiment

One PC with any OS (In our experiment Ubuntu 14.04), Genymotion Virtual Machine installed, A real android-powered smart-phone (rooted device), Wireshark.

6.1 Purpose

Capturing and analyzing Android Network traffic.

6.2 Install Wireshark

If you the Wireshark has not been installed on your host machine, you can do following:
Open up an Ubuntu terminal and type in:
sudo add-apt-repository ppa:n-muench/programs-ppa
sudo apt-get update
sudo apt-get install wireshark
when your are done type “wireshark”

-To check installation type:
whereis wireshark

6.3 Install tcpdump on real smart-phone

If your device has no tcpdump binary file, install it following instructions below:
-Download a tcpdump precompiled binary for ARM (Tcpdump-4.3.1-arm.tar.gz) from:
http://omappedia.org/wiki/USB_Sniffing_with_tcpdump
-Decompress to obtain the tcpdump binary.
Connect Android device to computer via USB cable, change the directory using “cd” and go to where you downloaded tcpdump.
Open a command terminal and execute the following commands:
tcpdump
adb push tcpdump /sdcard/tcpdump
adb shell
su
mount -t ext3 -o rw,remount /dev/block/mmcblk0p25/system
mount -o remount,ro /system
cat /sdcard/tcpdump /system/bin/tcpdump
chown root:shell /system/bin/tcpdump
chmod 754 /system/bin/tcpdump
mount -o remount,ro /system
rm /sdcard/tcpdump

6.4 Quick Wireshark Guide

Wireshark is a very powerful and popular network analyzer for Windows, Mac and Linux. It is a tool that is used to inspect data passing through a network interface which could be your
Ethernet, LAN and WiFi.

### 6.4.1 Start Wireshark

To open wireshark in Linux open it in a terminal with:

```bash
sudo wireshark
```

![Figure 90: Running Wireshark on Ubuntu](image)

This will open the Wireshark GUI.

**Note:** you need to make sure that Wireshark has sufficient rights to use a network card to capture data. You need to run Wireshark on an account with sufficient privileges to capture (Super-user).

### 6.4.2 Wireshark GUI

Once the Wireshark GUI has opened, you will see that the dashboard has a left column box called “Interface List”. This list lets you know which devices and capture cards you can use.

![Figure 91: Wireshark GUI](image)
At the top of the application there is an option called “Capture Options” which is exactly that, it allows you to modify and tweak how you would like to capture the packets of data that are being transmitted over your network.

Figure 92: Wireshark GUI, Capture Options

6.4.3 Capture Interface Options

If you have a look at your interface list you will see that one of your devices is actually sending and receiving packets.

Figure 93: Wireshark Interface

This screen shot shows the Wireshark capture interfaces, in other words, it shows what pro-
cesses and platforms are receiving and sending data on your machine. If you have a wireless card, then it will show it, etc.

### 6.4.4 The Main Packets Panel

Once you are happy with the interface you would like to use, go ahead and click “start” and Wireshark will show all the packets that are being transmitted over your network. If you open a web browser or for example watch a video on YouTube, you will notice a sudden surge of packet data.

The whole point here is to find patterns or anything that looks suspicious. Taking look at the columns at the top of the Wireshark interface from left to right:
- The first number is the “packet number”.
- The second column shows how many seconds it has been since the start of the capture.
- The third column is the source IP Address.
- The fourth column shows the destination IP Address.
- The fifth column is the protocol that sent the packet, i.e. it could be DNS, TCP (Transmission Control Protocol) or even HTTP.

Filtering the packets is key when using Wireshark—done by using the search bar within the interface (top left). If you right click on a packet of interest you can “follow TCP stream” and you get a ton of raw information.

![Wireshark Main Panel](image)

Figure 94: Wireshark Main Panel

60
6.5 Capturing Network Traffic - Real device

Connect the real device to PC via USB cable. Do not forget to check mark “USB Debugging”. You must go to developer options. In order to activate developer options on your phone, you can do following:
- Go to Settings, select “about phone”, then “software information”.
- Tap 5 times on “Build number”. Now you are a developer!
- Go back to Settings, select developer options and check mark “USB debugging”.

Open up an Ubuntu terminal then type:
adb devices (to see if your smart-phone is connected or not)
adb shell
su
tcpdump -n -s 0 -w -l nc -l -p 11233

Open up another terminal: find the place you already installed Wireshark using ‘whereis wire- shark’ command type in:
adb forward tcp:11233 tcp:11233 && nc 127.0.0.1 11233 l /usr/bin/wireshark -k -S -i -

Figure 95: Capture traffic from real device (1)
6.6 Capturing Network Traffic - Genymotion virtual machine

Launch Genymotion emulator. Use “cd” to change the directory to Genymotion home directory.
Type in: 

```
adb shell
tcpdump -p -vv -s 0 -w /sdcard/file.pcap
```

Figure 96: Capture traffic from real device (2)
The “file.pcap” is stored at sdcard folder of Genymotion (To exit type ctrl+C). When it is finished type “exit”, and now you are in Ubuntu terminal and type in:

```adb pull /sdcard/file.pcap /home/user/```

In another terminal type “wireshark”. Open the file.pcap with wireshark as shown below:
6.7 Source/Reference

http://omappedia.org/wiki/USB_Sniffing_with_tcpdump
https://josetrochecoder.wordpress.com/2013/11/04/installing-tcpdump-for-android/
Appendix

Some Useful Commands

The **adb install** and **adb uninstall** commands allows you to install an application APK package and uninstall it on the connected emulator or device.

- The most commonly used “adb” command is the “adb install” command that allows you to install an application stored in the specified APK file to the connect Android emulator or device:
  
  \[
  \text{adb install \ "path_to_apk"}
  \]

- You can also uninstall a package from the connected emulator or device by using the “adb uninstall” command:
  
  First, we must find the package name using below command:
  
  \[
  \text{adb shell pm list packages | grep \"apk_name\"}
  \]
  
  Then:
  
  \[
  \text{adb uninstall \ \"package_name\"}
  \]

  The “adb” tool also offers us commands to copy files into and from the connected Android emulator or devices.

- To copy a file or folder from the local system to the remote emulator or device. Type in an Ubuntu terminal:
  
  \[
  \text{adb push \ "path_to_file_local" \ "path_to_file_remote"}
  \]

  For instance, \texttt{adb push /home/Pictures/UNIPD.jpg /storage/sdcard/Picture}

  This command copies UNIPD.jpg file from the local system to emulator or device.

- To copy a file or folder from the emulator or device to local system. Type in an Ubuntu terminal:
  
  \[
  \text{adb pull \ "path_to_file_remote" \ "path_to_file_local"}
  \]

  For instance, \texttt{adb pull /data/app /home/app}

  This command copies file app from emulator or device to local system.

  The file will be pulled and saved in the directory pointed and you can see the transfer rate in the command prompt.

\textbf{Note:} pulling any file from the device is easy but pushing files to the device needs permissions. You can push any file to the internal or external SD card without any issue, but when you are pushing it to the System partitions you might get this error:

\[
\text{adb push /home/hossein/FaceLock.apk /system}
\]

failed to copy “FaceLock.apk” to “/system/FaceLock.apk”: Permission denied.