FORENSICS
LAB SERIES

Lab 16: Introduction to Android OS

Material in this Lab Aligns to the Following Certification Domains/Objectives	
Certified Cyber Forensics Professional (CCFP) Objectives	Computer Hacking Forensic Investigator (CHFI) Objectives
4: Digital Forensics	20: Mobile Forensics

Document Version: 2016-08-17

Contents

Introduction

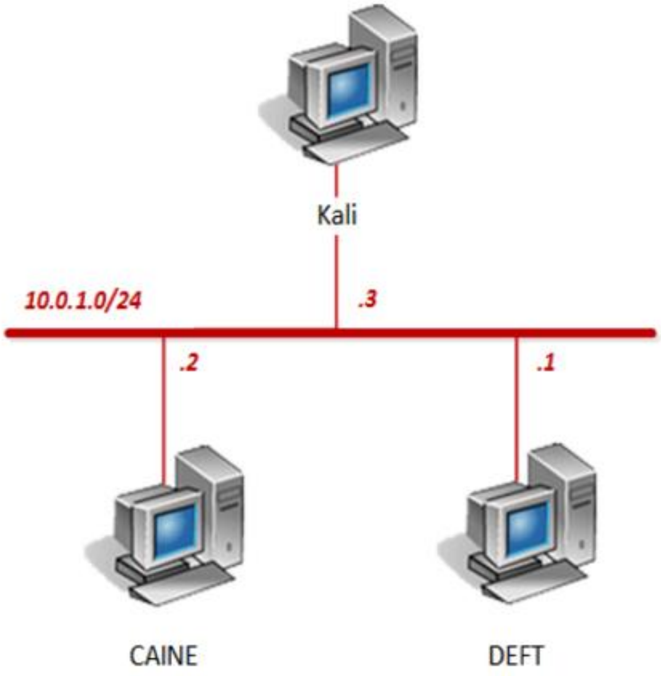
This lab will introduce the Android operating system, which can be found in many mobile devices. Different pieces of the operating system using *Android-SDK* will be examined throughout the lab.

Objective

In this lab, you will be conducting forensic practices using various tools. You will be performing the following tasks:

1. Launching Android SDK
2. Exploring the Android Filesystem

Pod Topology



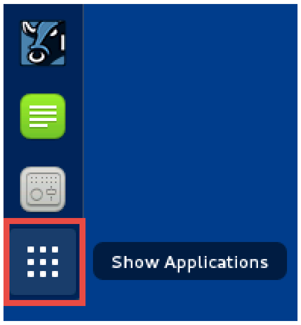
Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

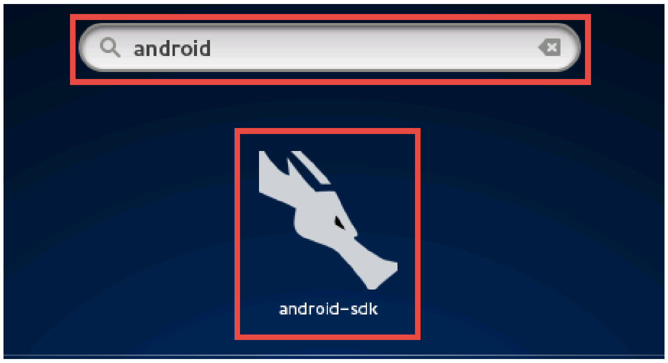
Virtual Machine	IP Address	Account (if needed)	Password (if needed)
DEFT	10.0.1.1	deft	password
CAINE	10.0.1.2	caine	
Kali	10.0.1.3	root	toor

1 Launching Android SDK

1. Click on the **Kali** graphic on the *topology page* to open the VM.
2. Login using `root` as the *username* and `toor` as the *password*.
3. Click on the **Show Applications** icon located in the left pane.

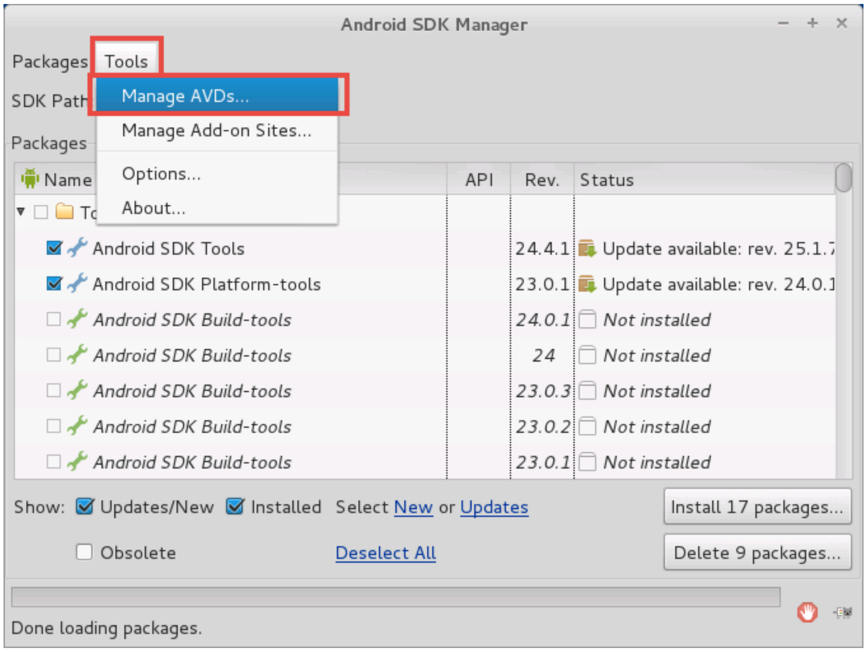


4. Type `android` in the search field located at the top. From the search results, click on the **android-sdk** icon to launch the *Android SDK* application.



When the *Android SDK Manager* is launched, wait 1-2 minutes until the progress bar on the bottom is finished.

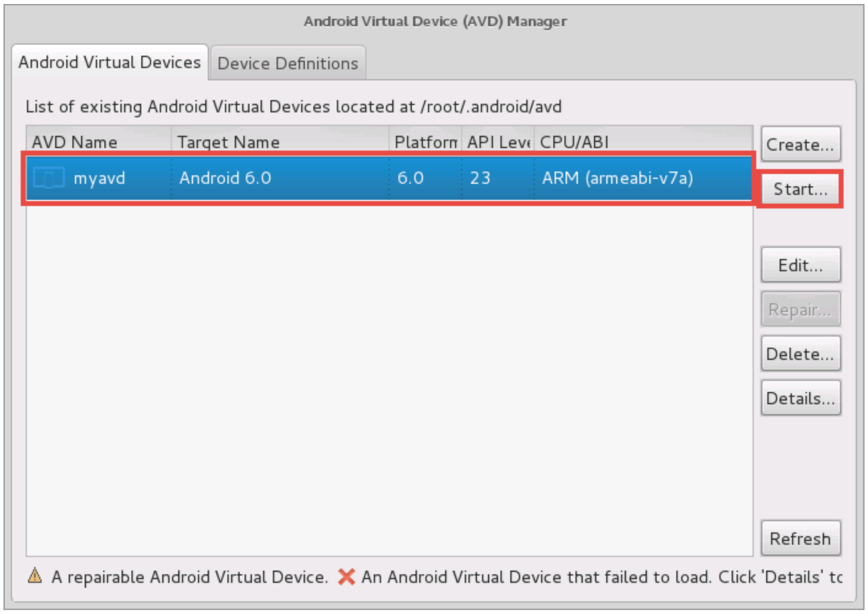
5. Using the *Android SDK Manager*, click on **Tools** and select **Manage AVDs**.



Name	API	Rev.	Status
▾ ☐ To			
☑ Android SDK Tools		24.4.1	Update available: rev. 25.1.7
☑ Android SDK Platform-tools		23.0.1	Update available: rev. 24.0.1
☐ *Android SDK Build-tools*		*24.0.1*	*Not installed*
☐ *Android SDK Build-tools*		*24*	*Not installed*
☐ *Android SDK Build-tools*		*23.0.3*	*Not installed*
☐ *Android SDK Build-tools*		*23.0.2*	*Not installed*
☐ *Android SDK Build-tools*		*23.0.1*	*Not installed*

Done loading packages.

6. Select **myavd** from the middle pane and click **Start**.

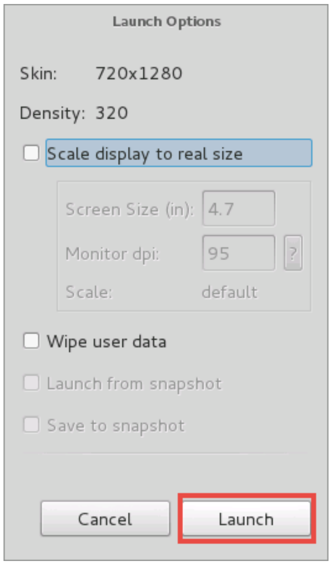


List of existing Android Virtual Devices located at /root/.android/avd

AVD Name	Target Name	Platform	API Leve	CPU/ABI
myavd	Android 6.0	6.0	23	ARM (armeabi-v7a)

⚠ A repairable Android Virtual Device. ✕ An Android Virtual Device that failed to load. Click 'Details' tc

7. In the *Launch Options* dialog window, leave the defaults set and click **Launch**.



Launch Options

Skin: 720x1280

Density: 320

☐ Scale display to real size

Screen Size (in): 4.7

Monitor dpi: 95

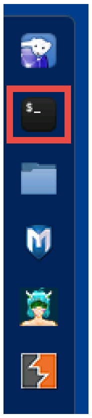
Scale: default

☐ Wipe user data

☐ Launch from snapshot

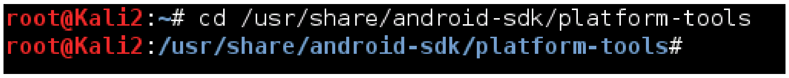
☐ Save to snapshot

8. Open a new terminal window by clicking on the **Terminal** icon.



9. Using the terminal, navigate to the **/usr/share/android-sdk/platform-tools/** directory by typing the command below followed by pressing the **Enter** key.

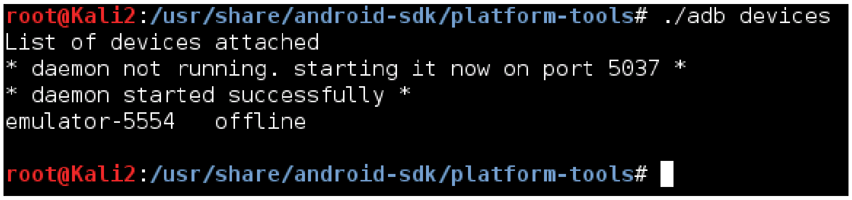
```
cd /usr/share/android-sdk/platform-tools
```



```
root@Kali2:~# cd /usr/share/android-sdk/platform-tools
root@Kali2:/usr/share/android-sdk/platform-tools#
```

10. Enter the command below to connect to an Android emulator device using *Android Debug Bridge* (*adb*).

```
./adb devices
```

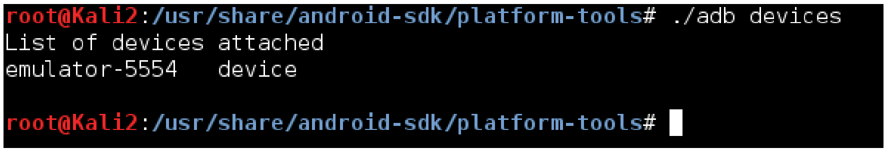


```
root@Kali2:/usr/share/android-sdk/platform-tools# ./adb devices
List of devices attached
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
emulator-5554   offline

root@Kali2:/usr/share/android-sdk/platform-tools# 
```

11. Initiate the same command once more.

```
./adb devices
```

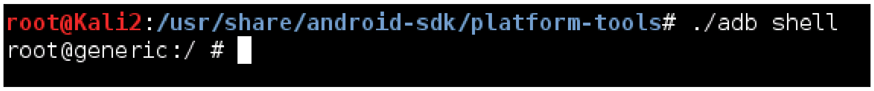


```
root@Kali2:/usr/share/android-sdk/platform-tools# ./adb devices
List of devices attached
emulator-5554   device

root@Kali2:/usr/share/android-sdk/platform-tools# 
```

12. Enter the command below to launch a Unix shell with the connected device.

```
./adb shell
```



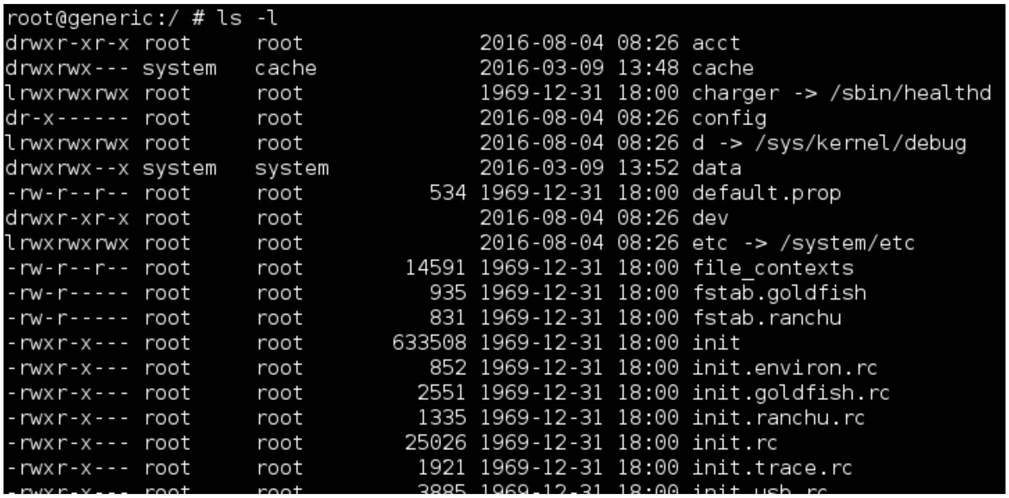
```
root@Kali2:/usr/share/android-sdk/platform-tools# ./adb shell
root@generic:/ # 
```

Notice the superuser status of being the *root* user.

2 Exploring the Android Filesystem

1. List the files in the current directory by entering the command below. Briefly analyze through the list of files on the *Android* system.

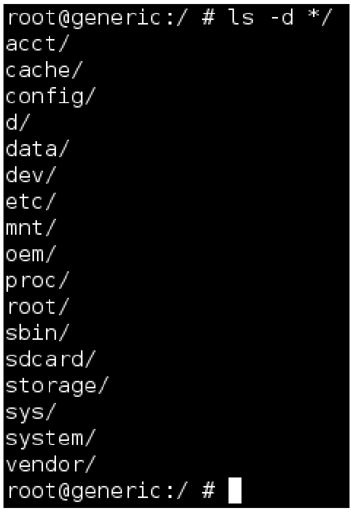
```
ls -l
```



```
root@generic:/ # ls -l
drwxr-xr-x root     root              2016-08-04 08:26 acct
drwxrwx--- system   cache             2016-03-09 13:48 cache
lrwxrwxrwx root     root              1969-12-31 18:00 charger -> /sbin/healthd
dr-x------ root     root              2016-08-04 08:26 config
lrwxrwxrwx root     root              2016-08-04 08:26 d -> /sys/kernel/debug
drwxrwx--x system   system            2016-03-09 13:52 data
-rw-r--r-- root     root          534 1969-12-31 18:00 default.prop
drwxr-xr-x root     root              2016-08-04 08:26 dev
lrwxrwxrwx root     root              2016-08-04 08:26 etc -> /system/etc
-rw-r--r-- root     root        14591 1969-12-31 18:00 file_contexts
-rw-r----- root     root          935 1969-12-31 18:00 fstab.goldfish
-rw-r----- root     root          831 1969-12-31 18:00 fstab.ranchu
-rwxr-x--- root     root       633508 1969-12-31 18:00 init
-rwxr-x--- root     root          852 1969-12-31 18:00 init.environ.rc
-rwxr-x--- root     root         2551 1969-12-31 18:00 init.goldfish.rc
-rwxr-x--- root     root         1335 1969-12-31 18:00 init.ranchu.rc
-rwxr-x--- root     root        25026 1969-12-31 18:00 init.rc
-rwxr-x--- root     root         1921 1969-12-31 18:00 init.trace.rc
-rwxr-x--- root     root         3885 1969-12-31 18:00 init.usb.rc
```

2. List only the directories in the current directory by entering the command below. Briefly analyze through the list of available directories on the *Android* system.

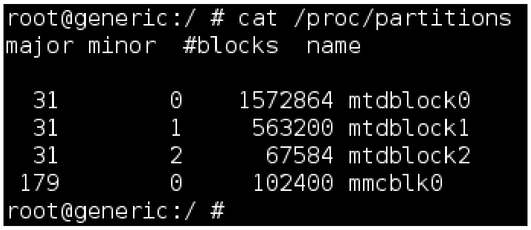
```
ls -d */
```



```
root@generic:/ # ls -d */
acct/
cache/
config/
d/
data/
dev/
etc/
mnt/
oem/
proc/
root/
sbin/
sdcard/
storage/
sys/
system/
vendor/
root@generic:/ #
```

3. Identify the partition structure by typing the command below followed by pressing **Enter**.

```
cat /proc/partitions
```



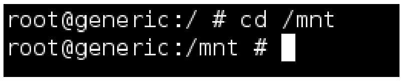
```
root@generic:/ # cat /proc/partitions
major minor  #blocks  name

  31        0    1572864 mtdblock0
  31        1     563200 mtdblock1
  31        2      67584 mtdblock2
 179        0     102400 mmcblk0
root@generic:/ #
```

Notice that there are three partitions listed of the *Memory Technology Device* (*mtd*) and one SD card *Multimedia Card* (*mmc*).

4. Identify the filesystem mount points by entering the command below.

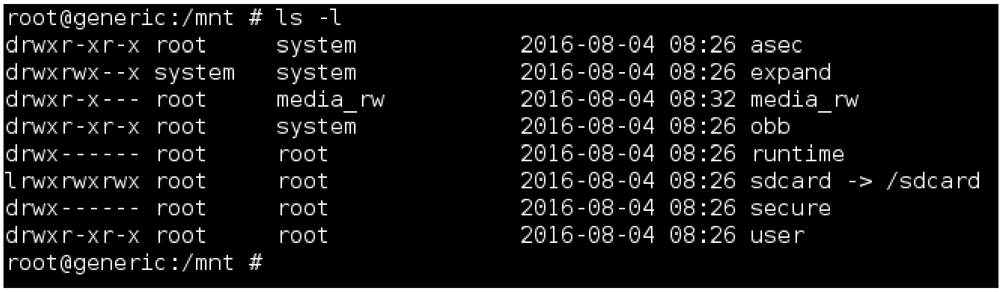
```
cd /mnt
```



```
root@generic:/ # cd /mnt
root@generic:/mnt #
```

5. List the files in the current directory by entering the command below.

```
ls -l
```

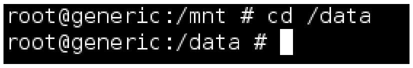


```
root@generic:/mnt # ls -l
drwxr-xr-x root     system           2016-08-04 08:26 asec
drwxrwx--x system   system           2016-08-04 08:26 expand
drwxr-x--- root     media_rw         2016-08-04 08:32 media_rw
drwxr-xr-x root     system           2016-08-04 08:26 obb
drwx------ root     root             2016-08-04 08:26 runtime
lrwxrwxrwx root     root             2016-08-04 08:26 sdcard -> /sdcard
drwx------ root     root             2016-08-04 08:26 secure
drwxr-xr-x root     root             2016-08-04 08:26 user
root@generic:/mnt #
```

Notice these are the mount points for all filesystems whether their external or internal.

6. Enter the command below to navigate to the **/data** directory which contains the user's applications and data.

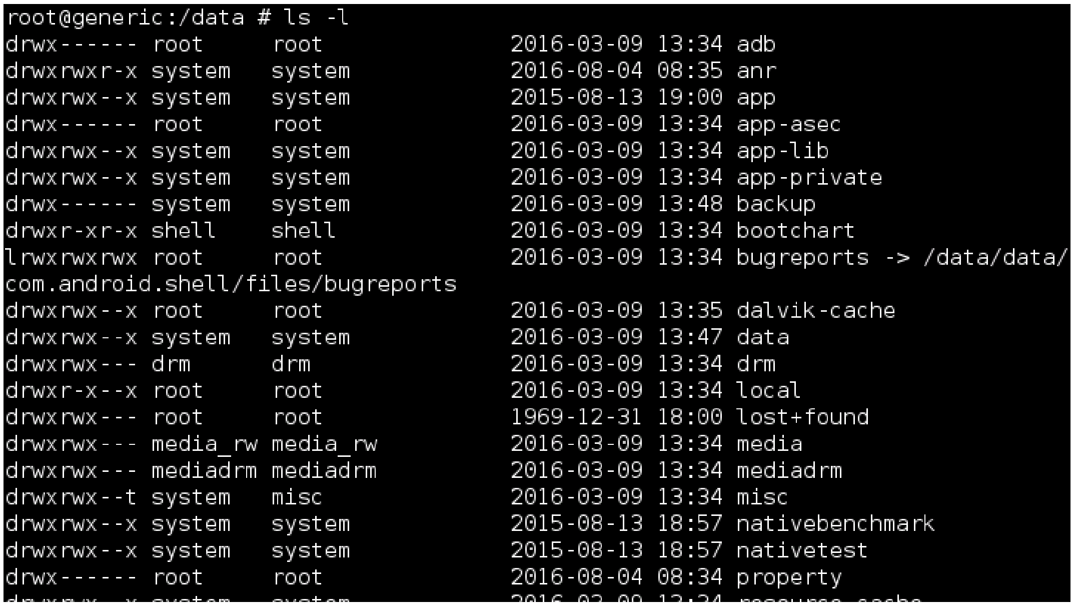
```
cd /data
```



```
root@generic:/mnt # cd /data
root@generic:/data #
```

7. Once in the */data* directory, enter the command below to list the files in a list view.

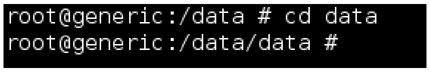
```
ls -l
```



```
root@generic:/data # ls -l
drwx------ root     root              2016-03-09 13:34 adb
drwxrwxr-x system   system            2016-08-04 08:35 anr
drwxrwx--x system   system            2015-08-13 19:00 app
drwx------ root     root              2016-03-09 13:34 app-asec
drwxrwx--x system   system            2016-03-09 13:34 app-lib
drwxrwx--x system   system            2016-03-09 13:34 app-private
drwx------ system   system            2016-03-09 13:48 backup
drwxr-xr-x shell    shell             2016-03-09 13:34 bootchart
lrwxrwxrwx root     root              2016-03-09 13:34 bugreports -> /data/data/
com.android.shell/files/bugreports
drwxrwx--x root     root              2016-03-09 13:35 dalvik-cache
drwxrwx--x system   system            2016-03-09 13:47 data
drwxrwx--- drm      drm               2016-03-09 13:34 drm
drwxr-x--x root     root              2016-03-09 13:34 local
drwxrwx--- root     root              1969-12-31 18:00 lost+found
drwxrwx--- media_rw media_rw          2016-03-09 13:34 media
drwxrwx--- mediadrm mediadrm          2016-03-09 13:34 mediadrm
drwxrwx--t system   misc              2016-03-09 13:34 misc
drwxrwx--x system   system            2015-08-13 18:57 nativebenchmark
drwxrwx--x system   system            2015-08-13 18:57 nativetest
drwx------ root     root              2016-08-04 08:34 property
drwxrwx--x system   system            2016-03-09 13:34 resource-cache
```

8. Dig deeper by navigating to the **/data/data** directory to find where the private user data is contained. Enter the command below.

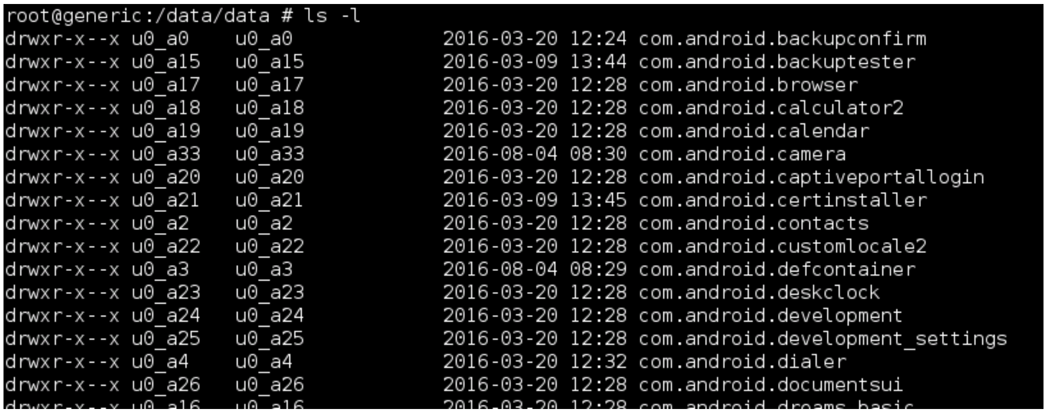
```
cd data
```



```
root@generic:/data # cd data
root@generic:/data/data #
```

9. Enter the command below to list the files in the current directory.

```
ls -l
```

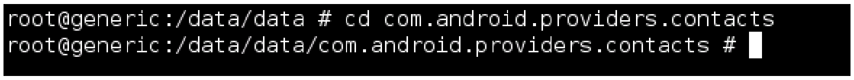


```
root@generic:/data/data # ls -l
drwxr-x--x u0_a0    u0_a0             2016-03-20 12:24 com.android.backupconfirm
drwxr-x--x u0_a15   u0_a15            2016-03-09 13:44 com.android.backuptester
drwxr-x--x u0_a17   u0_a17            2016-03-20 12:28 com.android.browser
drwxr-x--x u0_a18   u0_a18            2016-03-20 12:28 com.android.calculator2
drwxr-x--x u0_a19   u0_a19            2016-03-20 12:28 com.android.calendar
drwxr-x--x u0_a33   u0_a33            2016-08-04 08:30 com.android.camera
drwxr-x--x u0_a20   u0_a20            2016-03-20 12:28 com.android.captiveportallogin
drwxr-x--x u0_a21   u0_a21            2016-03-09 13:45 com.android.certinstaller
drwxr-x--x u0_a2    u0_a2             2016-03-20 12:28 com.android.contacts
drwxr-x--x u0_a22   u0_a22            2016-03-20 12:28 com.android.customlocale2
drwxr-x--x u0_a3    u0_a3             2016-08-04 08:29 com.android.defcontainer
drwxr-x--x u0_a23   u0_a23            2016-03-20 12:28 com.android.deskclock
drwxr-x--x u0_a24   u0_a24            2016-03-20 12:28 com.android.development
drwxr-x--x u0_a25   u0_a25            2016-03-20 12:28 com.android.development_settings
drwxr-x--x u0_a4    u0_a4             2016-03-20 12:32 com.android.dialer
drwxr-x--x u0_a26   u0_a26            2016-03-20 12:28 com.android.documentsui
drwxr-x--x u0_a16   u0_a16            2016-03-20 12:28 com.android.dreams.basic
```

Notice this is where all the application directories and user's private data is stored in each of the app's respective directories.

10. Navigate to the contacts application by entering the command below.

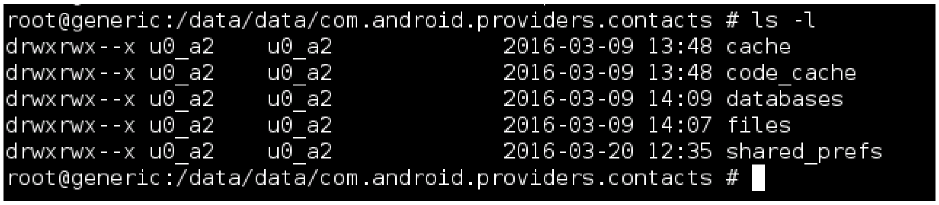
```
cd com.android.providers.contacts
```



```
root@generic:/data/data # cd com.android.providers.contacts
root@generic:/data/data/com.android.providers.contacts #
```

11. View the contents of the contacts application by entering the command below.

```
ls -l
```

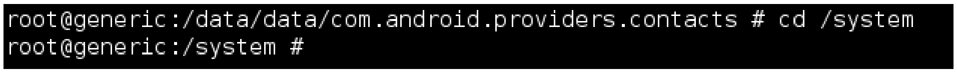


```
root@generic:/data/data/com.android.providers.contacts # ls -l
drwxrwx--x u0_a2    u0_a2              2016-03-09 13:48 cache
drwxrwx--x u0_a2    u0_a2              2016-03-09 13:48 code_cache
drwxrwx--x u0_a2    u0_a2              2016-03-09 14:09 databases
drwxrwx--x u0_a2    u0_a2              2016-03-09 14:07 files
drwxrwx--x u0_a2    u0_a2              2016-03-20 12:35 shared_prefs
root@generic:/data/data/com.android.providers.contacts #
```

Notice the databases folder. This is where the contacts are stored in *SQLite* format.

12. Navigate to the **/system** directory by entering the command below.

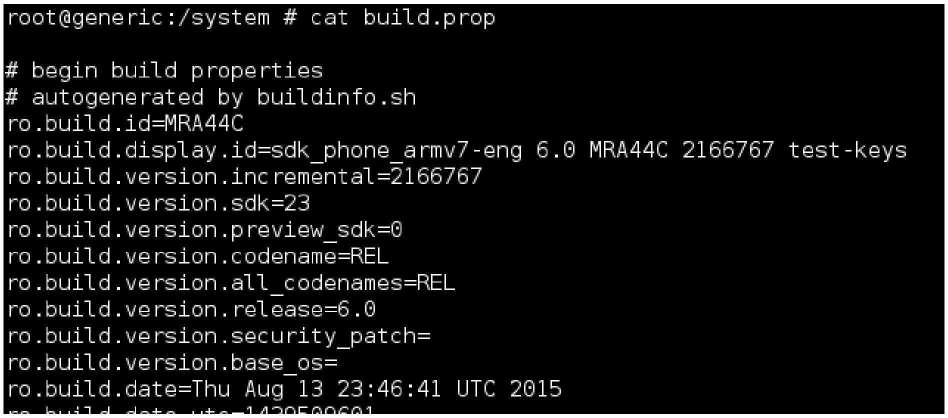
```
cd /system
```



```
root@generic:/data/data/com.android.providers.contacts # cd /system
root@generic:/system #
```

13. Identify the build properties of the *Android* device by entering the command below.

```
cat build.prop
```



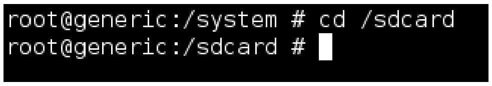
```
root@generic:/system # cat build.prop

# begin build properties
# autogenerated by buildinfo.sh
ro.build.id=MRA44C
ro.build.display.id=sdk_phone_armv7-eng 6.0 MRA44C 2166767 test-keys
ro.build.version.incremental=2166767
ro.build.version.sdk=23
ro.build.version.preview_sdk=0
ro.build.version.codename=REL
ro.build.version.all_codenames=REL
ro.build.version.release=6.0
ro.build.version.security_patch=
ro.build.version.base_os=
ro.build.date=Thu Aug 13 23:46:41 UTC 2015
ro.build.date.utc=1439509601
```

Notice the device properties, including *CPU* information can be found here.

14. Navigate to the **/sdcard** directory by entering the command below.

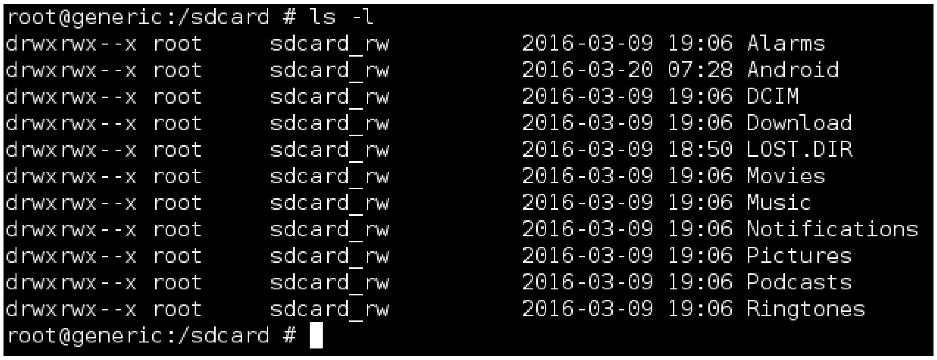
```
cd /sdcard
```



```
root@generic:/system # cd /sdcard
root@generic:/sdcard #
```

15. List the files in the current directory to identify the contents of the SD card. Enter the command below.

```
ls -l
```



```
root@generic:/sdcard # ls -l
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Alarms
drwxrwx--x root     sdcard_rw          2016-03-20 07:28 Android
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 DCIM
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Download
drwxrwx--x root     sdcard_rw          2016-03-09 18:50 LOST.DIR
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Movies
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Music
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Notifications
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Pictures
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Podcasts
drwxrwx--x root     sdcard_rw          2016-03-09 19:06 Ringtones
root@generic:/sdcard #
```

Notice the internal SD card is accessible where pictures and other data can be stored by applications like the camera app "*DCIM*".

16. Close all **PC Viewers** and end the reservation to complete the lab.