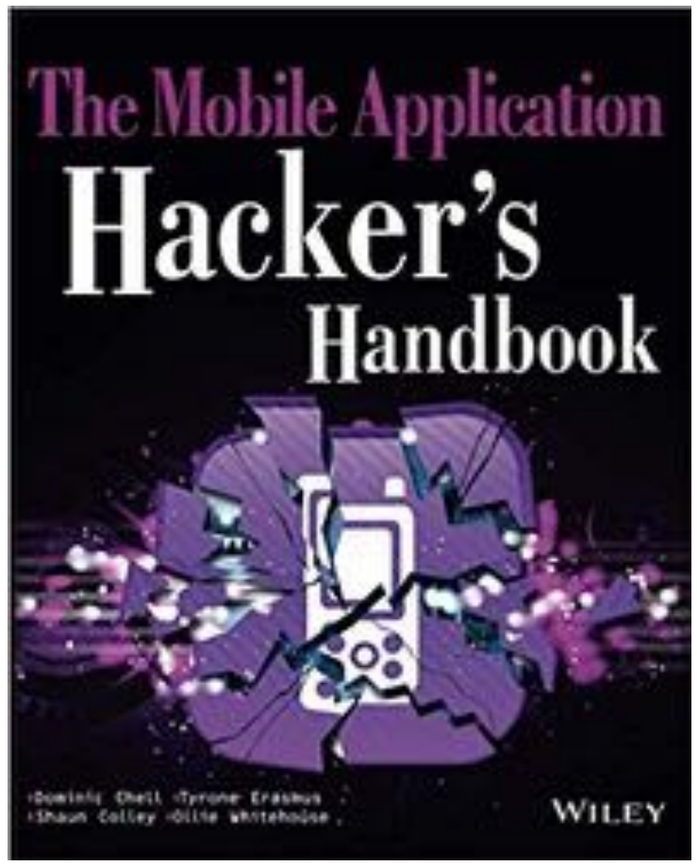
CNIT 128
Hacking Mobile Devices



6. Analyzing Android Applications Part 1

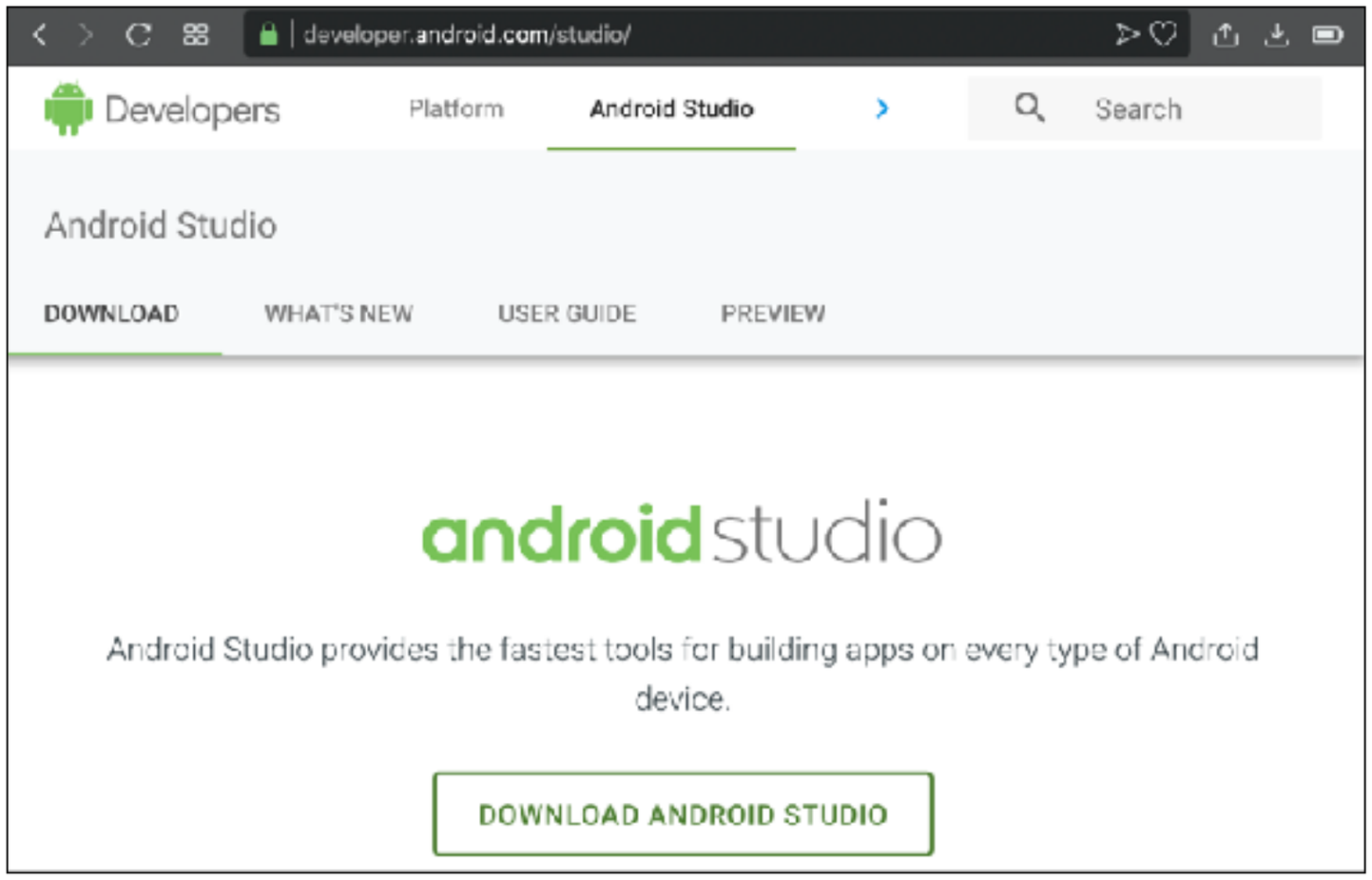
Topics

- Part 1
  - Creating Your First Android Environment
  - Understanding Android Applications
- Part 2
  - Understanding the Security Model
  - Reverse-Engineering Applications

Creating Your First Android Environment

Android SDK

- Android Studio: complete app IDE
  - Integrated Development Environment
- Command line tools only



Host OS

- Mac or Linux is best
- Windows causes many problems
  - Bluestacks is best emulator for Windows
- We'll use Kali Linux
  - Android tools already installed

Android SDK Tools

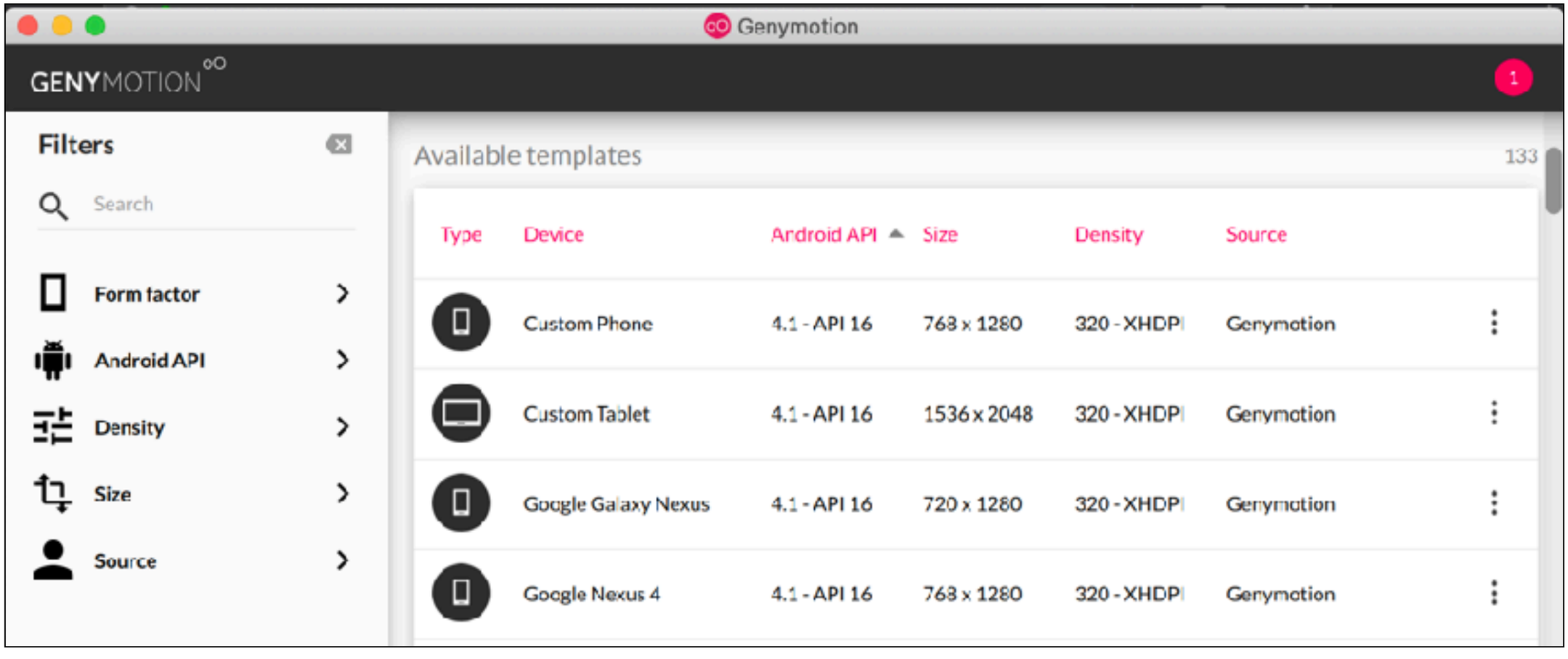
- **adb**
  - **The main tool you'll use**
  - Install apps
  - Pull files off the device
  - Get a shell
  - Read logs

Android SDK Tools

- **monitor**
  - See running processes
- **android**
  - Manage Android emulators
- **aapt**
  - Used by apktool to package source code into binary form when building apps

Emulators

- Genymotion has 133 of them available



Emulator Restrictions

- Have root access
- No physical USB, headphones, Wi-Fi, Bluetooth
  - Cannot make phone calls
    - Can simulate calls and SMS

Popular Emulators

- Virtualbox running Android x86
- BlueStacks, YouWave, Windows Android
  - All run on Windows

Understanding Android Applications

Android OS Basics

- Modified Linux kernel
- Virtual machine that runs Java-like apps
  - Dalvik Virtual Machine

User Accounts

- On Linux, every app a user launches runs with the same user ID (UID)
- This is the same as Windows

Activities Terminal Wed 15:03

Easy
Moves Left: 20 00:00

student@ubuntu: ~

File Edit View Search Terminal Help

```
student@ubuntu:~$ ps uT
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
student   11064  0.0  0.0  29552  4768 pts/0    Ss   14:54   0:00 bash
student   21155  0.5  0.4 474632 33076 pts/0    Sl   15:00   0:00 gnome-mahjongg
student   21159  0.3  0.5 844000 45368 pts/0    Sl   15:00   0:00 gnome-calculator
student   21470  6.1  2.8 1773032 230632 pts/0  Sl   15:01   0:03 /usr/lib/firefox/firefox https://samsclass.info
student   21534  3.6  2.1 1628788 176824 pts/0  Sl   15:01   0:02 /usr/lib/firefox/firefox -contentproc -childID 1 -is
student   21594  0.5  1.2 1508008 103376 pts/0  Sl   15:01   0:00 /usr/lib/firefox/firefox -contentproc -childID 2 -is
student   21648  0.2  0.9 1492448 80740 pts/0   Sl   15:01   0:00 /usr/lib/firefox/firefox -contentproc -childID 3 -is
student   21695  0.0  0.0  46772  3668 pts/0    R+   15:02   0:00 ps uT
student@ubuntu:~$
```

samsclass.info: Sam Bow samsclass.info: Sam Bow

https://samsclass.info

Sam Bowne

CCSF · CCSF Student Email · CNIT · VMware & Microsoft Software · Evals · Azure & Parallels · News · Contact & Bio · Old Classes · Use Policy · Disclosure Policy · Hall of Fame · Check-in · Videos & Podcasts · Student Videos · CTFs · Defcon Materials · Recommended Training Events · Uptime

7 8 9 ÷ ⌫ C
4 5 6 × ( )
1 2 3 − x^2 √
0 . % + =

Android UIDs

- Each app runs under a different UID
  - Unless a developer chooses to set several apps with the same signature share a UID

Android UIDs

- adb shell
- ps -A
- Each app has an account starting **u0_**
- There are also special accounts like **system** and **root**

```
u0_a47        5959  3466  849068  84376 ep_poll      ef818bc9 S com.android.email
u0_a7         5971  3466  877052 100464 ep_poll      ef818bc9 S com.android.dialer
u0_a105       6173  5420    7620   2708 __skb_rec+   ee743bc9 S libestool2.so
u0_a5         6194  3466  853624  76844 ep_poll      ef818bc9 S com.android.chrome:webview_service
u0_a73        6479  3466  838824  75676 ep_poll      ef818bc9 S com.google.android.ext.services
system        6503  3466  839248  70404 ep_poll      ef818bc9 S com.android.keychain
u0_a52        6547  3466  920908 153448 ep_poll      ef818bc9 S be.delhaize
root          6756   157    5736   2960 sigsuspend   f0225bc9 S sh
root          6770  6756    7560   2888 0            f7014bc9 R ps
vbox86p:/ #
```

/data/data

- Home directory for apps

```
vbox86p:/ # ls -l /data/data
total 1224
drwx------  4 system         system         4096 2019-01-13 18:42 android
drwx------  8 u0_a52         u0_a52         4096 2019-01-30 18:04 be.delhaize
drwxr-x--x  8 u0_a87         u0_a87         4096 2019-01-13 19:13 com.absmallbusinessmarketing.askalawyer
drwxr-x--x  4 u0_a40         u0_a40         4096 2019-01-13 18:42 com.amaze.filemanager
drwx------ 21 u0_a101        u0_a101        4096 2019-01-30 18:04 com.amazon.venezia
drwx------  4 u0_a28         u0_a28         4096 2019-01-13 18:42 com.android.backupconfirm
```

Android Packages

- APK file
  - Zipped archive
  - Contains code, resources, and metadata

APK Packaging Process

- **aapt** converts XML resource files to binary form
- **aidl** converts **.aidl** files to **.java**
- All source code and output from **aapt** and **aidl** compiled into **.class** files by Java compiler
- **dx** converts .**class** files to a single **classes.dex** file

APK Packaging Process

- **apkbuilder** combines all compiles resources, images, and DEX file into an APK file
- **jarsigner** signs the APK

Structure of an APK

- **/assets**
- **/res**
- **/lib**
- **/META-INF**
- **AndroidManifest.xml**
- **classes.dex**
- **resources.rsrc**

Structure of an APK

- **/assets**
  - Files developer wants to include
- **/res**
  - Layouts, images, etc, in **raw** subdirectory
- **/lib**
  - Libraries, in subdirectories **x86**, **ARM**, **MIPS**

Structure of an APK

```
root@kali:~/apk/ask/ask/original/META-INF# ls -l
total 28
-rw-r--r-- 1 root root 1090 Jan 30 15:25 CERT.RSA
-rw-r--r-- 1 root root 9230 Jan 30 15:25 CERT.SF
-rw-r--r-- 1 root root 9177 Jan 30 15:25 MANIFEST.MF
```

- **/META-INF**
  - Certificate of application, file inventory with hashes
- **AndroidManifest.xml**
  - Configuration of application and security parameters

AndroidManifest.xml

```
root@kali:~/apk/ask/ask# cat AndroidManifest.xml
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.andr
oid.com/apk/res/android" android:hardwareAccelerated="true" package="com.absmallbusinessmarketing.
askalawyer" platformBuildVersionCode="19" platformBuildVersionName="4.4.2-1456859">
    <supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens=
"true" android:resizeable="true" android:smallScreens="true" android:xlargeScreens="true"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:hardwareAccelerated="true" android:icon="@drawable/icon" android:label="@
string/app_name">
        <activity android:configChanges="locale|keyboard|keyboardHidden|orientation|screenSize" an
droid:label="@string/activity_name" android:launchMode="singleTop" android:name="CordovaApp" andro
id:theme="@android:style/Theme.Black.NoTitleBar" android:windowSoftInputMode="adjustResize">
            <intent-filter android:label="@string/launcher_name">
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <provider android:authorities="absmallbusinessmarketing.askalawyer.plugin.emailcomposer.at
tachmentprovider" android:name="de.appplant.cordova.plugin.emailcomposer.AttachmentProvider"/>
    </application>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Structure of an APK

- **classes.dex**
  - Executable file containing Dalvik bytecode
- **resources.rsrc**
  - Application strings
  - Resources the developer chose to place here instead of **/res**

Installing Packages

- GTalkService
  - Maintains a connection to Google via pinned SSL
  - Fetches apps from Google Play
- Alternative apps stores are available
  - Amazon, Samsung, GetJar, etc.
  - Often even less safe

adb Commands

- **adb install** -- installs an app
- **adb shell** -- opens command shell
- **adb push** -- pushes file onto device
- **adb pull** -- pulls file from device
- **adb forward** -- forwards a TCP port
- **adb logcat** -- shows the syslog

Busybox

- A single binary with many useful Linux tools, including:
- chmod, cp, echo, grep, ifconfig, mv, nc, netstat, pwd, rm

Standard Android Tools

- **pm** -- Package Manager
- **pm list packages** -- show all packages
- **pm path** -- find stored APK for an app
- **pm install**
- **pm uninstall**

Standard Android Tools

- **logcat** -- view system logs
- **getprop** -- show system properties
- **dumpsys** -- status of system services

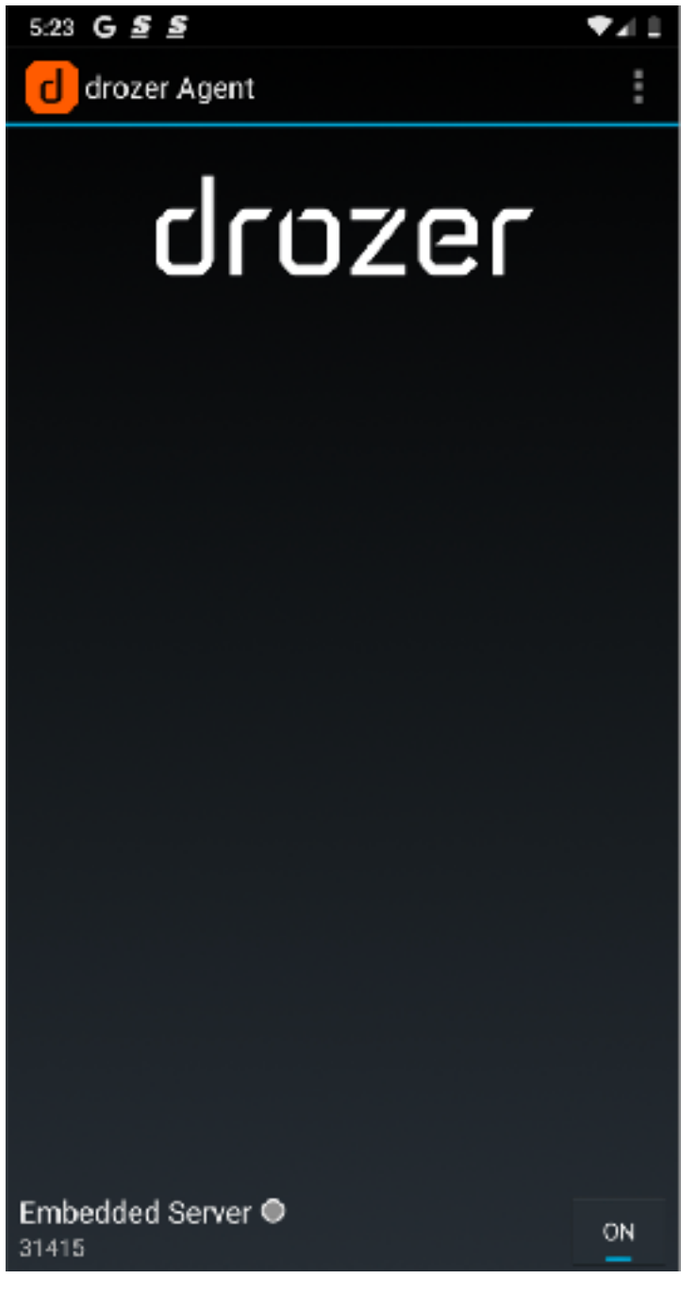
Drozer

- Security assessment tool
  - Finds vulnerabilities
  - Develops exploits
- Community edition is free
- Pro version costs money

How Drozer Works

- **Agent**
  - App on device, a Remote Administration Tool
- **Console**
  - Command-line tool on your computer
  - Interacts with the **Agent**
- **Server**
  - Routes sessions between **Agent** and **Console**

Agent and Console




```
root@kali:~/drozer# drozer console connect
Selecting f1c7150be9f0b60c (unknown Google 9)
                           ..                    ..:.
                          ..o..                  .r..
                           ..a..  . ..... .  ..nd
                             ro..idsnemesisand..pr
                             .otectorandroidsneme.
                           .,sisandprotectorandroids+.
                         ..nemesisandprotectorandroidsn:.
                        .emesisandprotectorandroidsnemes..
                      ..isandp,..,rotectorandro,..,idsnem.
                      .isisandp..rotectorandroid..snemisis.
                      ,andprotectorandroidsnemisisandprotec.
                     .torandroidsnemesisandprotectorandroid.
                     .snemisisandprotectorandroidsnemesisan:
                     .dprotectorandroidsnemesisandprotector.

drozer Console (v2.4.4)
dz> help

drozer: Android Security Assessment Framework

Type `help COMMAND` for more information on a particular command, or `help
MODULE` for a particular module.

Commands:

cd     contributors  env   help  load    permissions  set    unset
clean  echo          exit  list  module  run          shell

Miscellaneous help topics:

intents

dz>
```

Binder

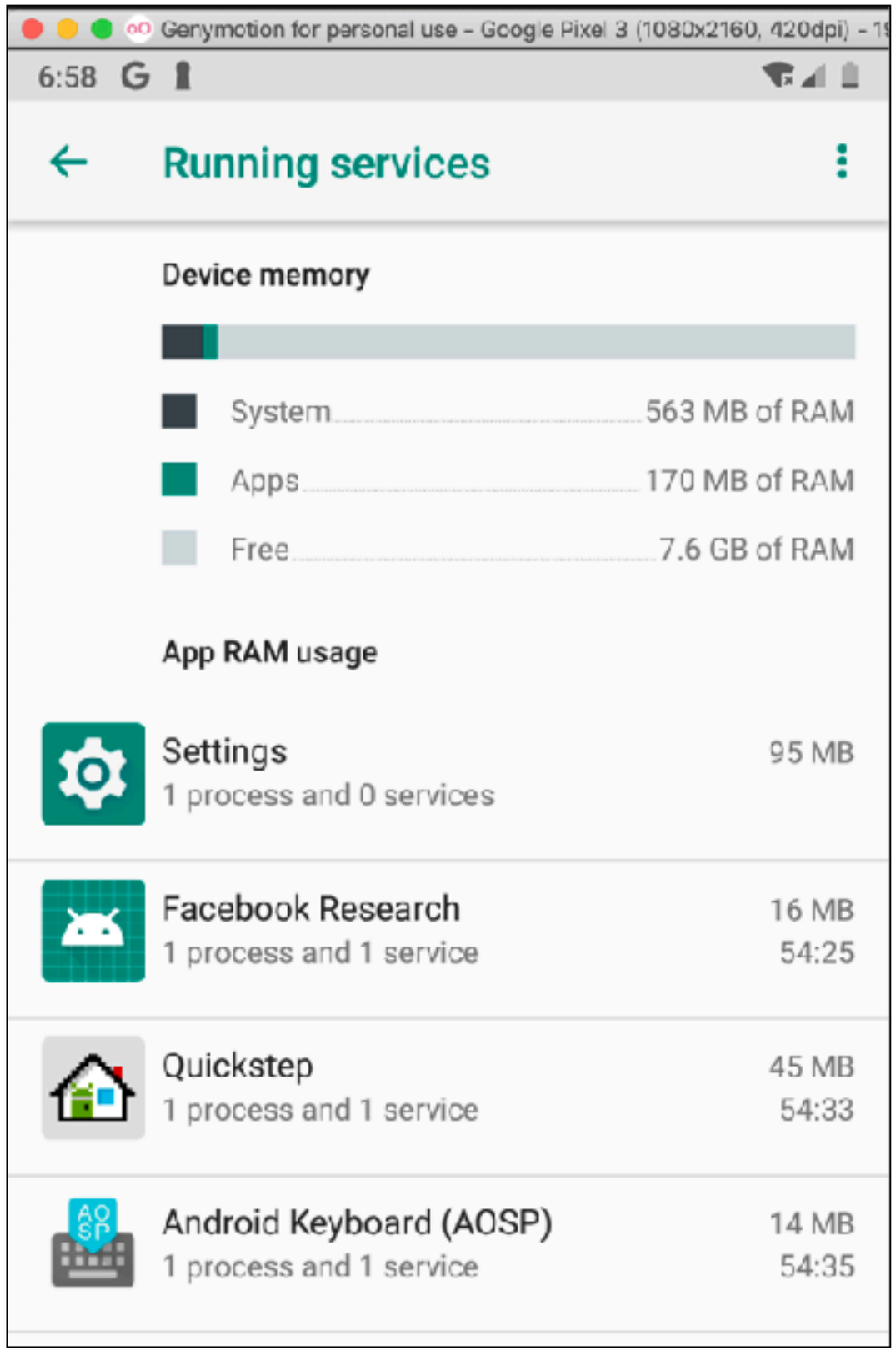
- Kernel module for Inter-Process Communication (IPC)
- A character device at **/dev /binder**

Application Components

- **Activities**
  - Visual screens of an app
- **Services**
  - Components with no GUI
- **Broadcast receivers**
  - Can detect events like an incoming SMS
- **Content providers**
  - Data storehouses, often SQLite

Running Services

- On Genymotion
  - **Settings**
  - **System**
  - **Advanced**
  - **Developer Options**
  - **Running services**



AndroidManifest.xml

- Lists all components usable in application
  - Except broadcast receivers

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.simple.mahh">

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.simple.mahh.MainActivity"
            android:label="@string/app_name"
            android:theme="@style/Theme.Base.AppCompat.Light.DarkActionBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.INTERNET" />

</manifest>
```

- Lists permissions

Intents

- An object used for messaging between apps
- Works by calling **binder**
- This intent opens google.com in a browser

```
Intent intent = new Intent(Intent.ACTION_VIEW);

intent.setData(Uri.parse("http://www.google.com"));

startActivity(intent);
```

Implicit Intent

- The code on the last slide does not specify the target app
- Any app that can respond to a VIEW action on a URL is eligible to receive the intent
- If only one app can handle it, it goes there
- Otherwise an application picker appears

Intent Filters

- An app with this filter can diplay Web pages

```
<activity android:name="MyBrowserActivity">

    <intent-filter>

        <action android:name="android.intent.action.VIEW"/>

        <data android:scheme="http" />

    </intent-filter>

</activity>
```

Explicit Intent

- Sends URL specifically to the Android browser

```
Intent intent = new Intent(Intent.ACTION_VIEW);

intent.setData(Uri.parse("http://www.google.com"));



String pack = "com.android.browser";

ComponentName comp = new ComponentName(pack, pack + ".BrowserActivity");

intent.setComponent(comp);



startActivity(intent);
```

Running an App

- When OS boots, a single VM starts
- A **zygote** process listens for app launch requests
- Each app that launches causes a **fork()**
- Core libraries are shared between VMs


Kahoot!

Demonstrations