UNIT 1 NOTES

Introduction to Android Security :

Reference: Slides - Dr. Digvijay Singh Rathod.

Defn: Android is an open-source operating system based on Linux with a Java and Kotlin programming interface for mobile devices. Android was developed by Open Handset Alliance (OHA)
↓
2007

Defn: Open Handset Alliance (OHA) is led by Google. It is a consortium of multiple companies like Samsung, Sony, Intel etc. to provide services & deploy handsets using the android platform.

Timeline:

2007 - beta version of Android SDK.
2008 - Android 1.0 Alpha
2012 - Android 4.1 Jelly Bean
2014 - Android 5.0 Lollipop
Aug 3 2020 - Android 10
2024 - Android 14.0

→ The Android Architecture.

1) Linux Kernel - provides a level of abstraction between device hardware & upper layers. Kernel contains drivers to understand hardware instruction.

1992 - Java v1.0

2) Libraries - Libraries that handle different types of data and are usually written in C or C++ programming language. (→ set of functionalities.)
Surface Manager, Media Framework, SQLite
3) Android Runtime - Dalvik Virtual Machine
4) Application Framework - Activity Manager, Content Manager, View System
5) Applications - Home, Contacts, Browser

Defn: Dalvik byte code is an optimized byte code suitable for low memory & low processing environments.

- JVM's byte code consists of one or more .class files but dalvik code has only one .dex file

JVM: Java Source Code → Java Compiler → JBC → JVM

DVM: Java Source Code → Java Compiler → JBC → Dex Compiler → DBC → DVM

Defn: Each android application is assigned a unique identifier (UID) and is run as a seperate process. This application sandboxing is done at the Kernel level. It applies to both native applications & OS applications.

06/01/25

→ How does JAVA achieve platform independency?
→ Difference b/w byte code & executable code?
  - Intermediate notation for JAVA.
  - The JAVA compiler remains common on all platforms.
→ The Java Virtual Machine remains different for different operating systems. Based on your operating system, the JVM will convert the Java Byte Code to executable code using its interpreter.
→ Compiled Java byte code is stored in a .class file.
→ Difference b/w .dex & .exe files?

09/01/25 UNIT 2 – Pen Testing.

→ RAID Model
→ Dynamic Hooking - FRIDA framework
→ Genymotion, AVD, Santoku, Diva_Beta

Reference: Android Security [Video 12]

Defⁿ: Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with a device.
It is a logical bridge b/w your computer & the android device.

1) Client → which sends commands, runs on development machine
2) Daemon (adbd) - runs commands on a device, background process
   ↳ Genymotion (AVM) → Santoku

- adb connect <ip address>
- adb devices
- adb kill-server
- adb start-server
- adb shell (ctrl+D to exit) -s
- adb push <local> <remote>
  ↳ santoku ↳ AVM
- adb pull <remote> <local>
- adb logcat

/data/data : application related pkg

→ Genymotion is a rooted virtual device

DATE
PAGE

Reference: Android Application Framework [video 02]

- developer.android.com

→ Components of Android Application:

1) Activity - UI, handle user integration with phone
2) Service - background processing
3) Broadcast Receiver - handle communication b/w Android OS & app.
4) Content Providers - handle Data & DBMS operations

Defn: AndroidManifest.xml (root of project) describes essential information about your app to the Android build tools, the Android O & google play.

* android.id

Defn: Resources are the additional files & static content that your code uses such as bitmaps, layout definitions, UI strings etc.

Defn: An Intent is a messaging object you can use to request an act from another app component.

(a) Action (b) Category

1) Explicit Intent - specify which app will satisfy the intent supplying either app's pkg name or a fully-qualified component class name.
2) Implicit Intent - declare a general action to perform which a component from another app to handle it.

Reference: [Video 13] Android Boot Process.

1) Boot ROM - information regarding booting-up of a computer stored on system-on-chip (SOCs) in read-only memory.
2) Boot loader - small program that loads the operating system into a computer's memory. (BIOS → BL)
3) Kernel - manages operations of memory & CPU time. It is the core component of an operating system.
4) Init - deamon (first) process that runs until the system shuts down in a Unix-based operating system.
5) Zygote - a process that forks itself in response to spawn requests from a master process. It acts as the root of all system & app processes.
6) System Server - a computer program or service that provides service to another program / user (client).

* Insecure logging
* Hardcoded Coding [redacted].

DATE
PAGE

Reference: [video 08] - Android Service

Explicit intent with service → passing explicit intent like an object

Run in the background ; no user-interaction (GUI) ; long-running service ; Inter-process communication (IPC)

1) Foreground Service - visible to the users (Music Player)
2) Background Service - not visible to users (Syncing | Storing)
3) Bound Service - run as long as another application's component is bounded to it.

* Started service can be a bounded service but a bounded service cannot be a started service.

1) Started service - startService() ; background process ; stopService() or stopSelf().

return START_STICKY

Methods of Android Service.

1) onStart()
2) onBind()
3) onUnbind()
4) onRebind()
5) onCreate()
6) onDestroy()

Reference: Android Sandboxing - Secure Interprocess Communication

* Linux user-based protection model -
  1) each user is assigned a unique UID → Linux identity
  2) all resources under a particular user are run with the same priviledges.

Since the application sandboxing is applied at the kernel level, it applies to both native & DS applications.
↳ Consistency across all platforms.

System services run in seperate processes & have more privileges.
↳ need IPC to interact with other system components securely.

Defn: A binder framework provides the capabilities required to organize all types of communication between various processes. Using this framework it is possible to perform a variety of actions such as invoking methods on remote objects.
↳ RPC, RMI, REST, SOAP

/dev/binder

Client — Proxy — Binder Driver — Stub — Service
A (kernel) B

1) Context Manager - name service (DNS)
2) Token - 32 bit unique ID

Reference: App Permissions.

* Purpose - to protect the privacy of an android user.

Normal ← Automatic Prompted → Dangerous

→ Android Manifest file

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

→ Protection Levels:

1) Normal Permissions - app needs to access data outside app's sandbox ; little risk.
2) Signature Permissions - granted at install time if app is signed by the same certificate as the app that defines the permission
3) Dangerous Permissions - data or resources that involve the user's private information.

. keystore

* Runtime requests (Android 6.0 & higher) - marshmallow.

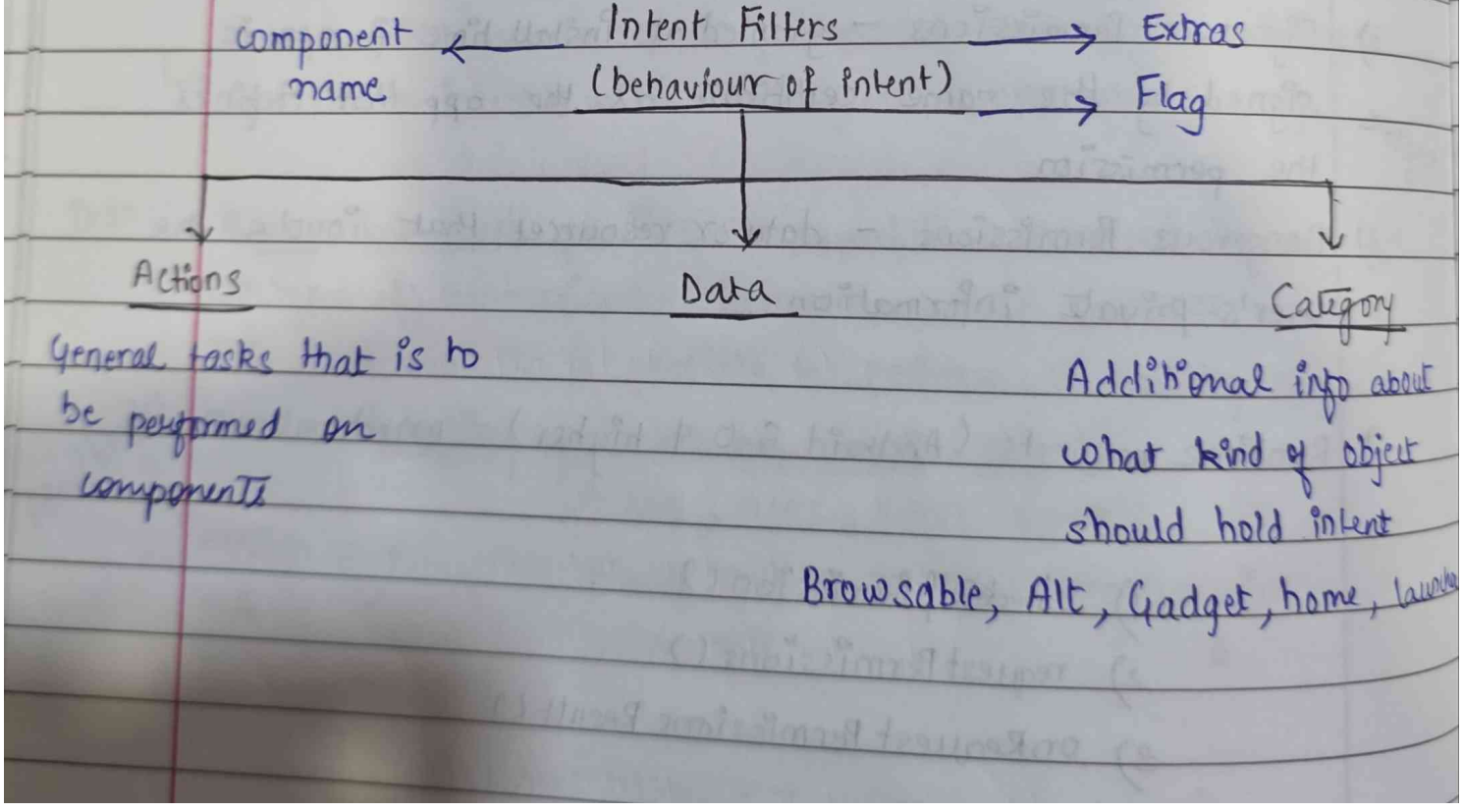
1) checkSelfPermission()
2) requestPermissions()
3) onRequestPermissionsResult()

Reference: [video 03] Intent

Defn: Intent Resolution is whenever android is given an intent (a messaging object), it has to figure out which activity or activities can handle it.

Explicit: Intent intent = new Intent (getAppContext(), ActivityTwo.class);
startActivity (intent)

Implicit: Intent intent = new Intent();
intent.setAction = (Intent.ACTION_DIAL)



General tasks that is to be performed on components

Additional info about what kind of object should hold intent

Browsable, Alt, Gadget, home, launcher

Reference: Content Providers.

Defn: Content Providers provide the content / data to Android application from android system or other android applications.

Defn: We need to use the content resolver to communicate with the content providers for data access. (DNS Resolver)

```
ContentResolver resolver = getContentResolver();
Cursor cursor = resolver.query(uri, projection, selection,
selectionArgs, sortOrder);
```

* Methods: Content provider hides the details related to database, files etc.
create, read, update, delete → insert, query, update, delete

Defn: URI Identifies data in a provider. (authority + path)

Defn: Cursors are created when you are executing a SELECT statement that returns more than one row.

Defn: Projection is the no. of columns returned.

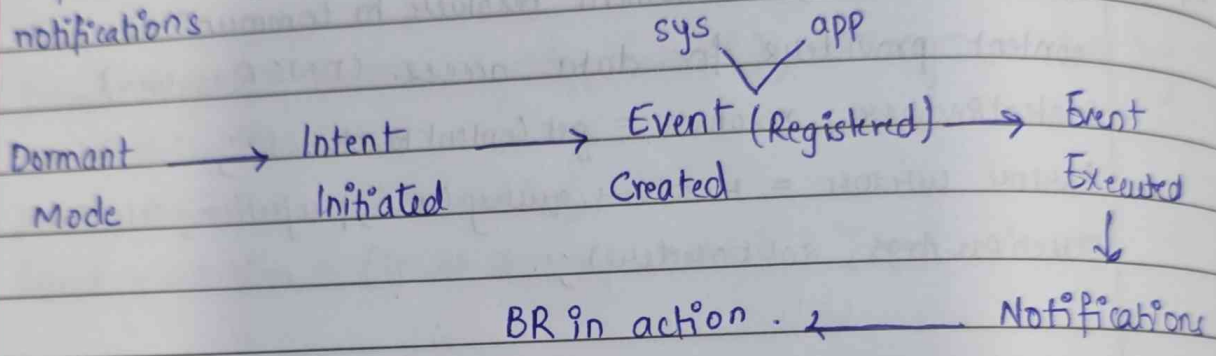
selection — WHERE col = 3

selectionArgs

Reference: Video 06 - Broadcast Receiver

Defn: Android broadcast receiver is a component that is used to broadcast the messages to system or other applications & responds to system's wide broadcast announcements.

Example: notifications



* Used for asynchronous inter-process communication.

Registering a Broadcast Receiver:
1. Static : <receiver>
2. Dynamic : Context.registerReceiver()

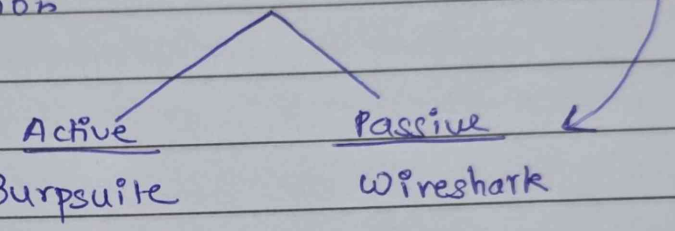
Classes of Broadcasts :
1. Ordered - synchronous ; android : priority ; receivers in queue
2. Normal - asynchronous ; not ordered

* Broadcasting custom intents - sendBroadcast()

Traffic Analysis for Android Devices

→ HTTPS protocol
- token sent via HTTP
- weak encryption

* tcpdump



→ tcpdump -s 0 -v out.pcap