**Android security**

**Security**

Android incorporates industry-leading security features and works with developers and device implementers to keep the Android platform and ecosystem safe. A robust security model is essential to enable a vigorous ecosystem of applications and devices built on and around the Android platform and supported by cloud services. As a result, through its entire development lifecycle, Android has been subject to a rigorous security program.

* **Android is designed:** 
  + To Be Open: Android was designed with multi-layered security that is flexible enough to support an open platform while still protecting all users of the platform.
* For Developers: Developers less familiar with security will be protected by safe defaults. In addition to providing a stable platform to build upon, Android gives additional support to developers in a number of ways.
* For Users: Users are provided visibility into permissions requested by each application and control over those permissions. This design includes the expectation that attackers would attempt to perform common attacks

## Linux Security

The foundation of the Android platform is the Linux kernel. The Linux kernel has been in widespread use for years, and is used in millions of security-sensitive environments. Through its history of constantly being researched, attacked, and fixed by thousands of developers, Linux has become a stable and secure kernel trusted by many corporations and security professionals.

As the base for a mobile computing environment, the Linux kernel provides Android with several key security features, including:

* A user-based permissions model
* Process isolation
* Extensible mechanism for secure IPC
* The ability to remove unnecessary and potentially insecure parts of the kernel

As a multiuser operating system, a fundamental security objective of the Linux kernel is to isolate user resources from one another. The Linux security philosophy is to protect user resources from one another. Thus, Linux:

* Prevents user A from reading user B's files
* Ensures that user A does not exhaust user B's memory
* Ensures that user A does not exhaust user B's CPU resources
* Ensures that user A does not exhaust user B's devices (e.g. telephony, GPS, Bluetooth)
* **Security tips**

## The Application Sandbox

Android’s application security is enforced by the application sandbox, which isolates apps from each other and protects apps and the system from malicious apps

## Filesystem Permissions

In a UNIX-style environment, filesystem permissions ensure that one user cannot alter or read another user's files. In the case of Android, each application runs as its own user. Unless the developer explicitly shares files with other applications, files created by one application cannot be read or altered by another application.

## Cryptography

* is associated with the process of converting ordinary plain text into unintelligible text and vice-versa.
* It is a method of storing and transmitting data in a particular form .
* not only protects data from theft or alteration, but can also be used for user authentication

## Interprocess Communication

Processes can communicate using any of the traditional UNIX-type mechanisms. Examples include the filesystem, local sockets, or signals. However, the Linux permissions still apply.

* In software systems it is often the case that different processes need to communicate with one another in order to cooperate on some task.
* The Android IPC mechanisms are designed to operate at the level of components rather than processes.
* In other words a component in one app may use IPC to communicate with a component in another app, or with another component in the same app**.**

**Difference between encryption and cryptology?**

* Earlier cryptography was effectively synonymous with encryption but nowadays cryptography is mainly based on mathematical theory and computer science practice.
* **Encryption**
* So, firstly we are now going to check what are the advantages of having encrypted Android phone.
  + It is very easy to do
  + Do not have to install any extra app
  + App data also get encrypted
  + Data Security Increases
  + Secure your phone from being retrieved

**Cons:**

* + Not available on every Android phone
  + It cannot be done using any app
  + It’s a very lengthy taking process
  + It cannot be revered
  + Decreases your phone performance

**System Partition and Safe Mode:**

* The system partition contains Android's kernel as well as the operating system libraries, application runtime, application framework, and applications. This partition is set to read-only. When a user boots the device into Safe Mode, third-party applications may be launched manually by the device owner but are not launched by default.

**Apps security :**

## Cost-Sensitive APIs

A cost sensitive API is any function that might generate a cost for the user or the network. The Android platform has placed cost sensitive APIs in the list of protected APIs controlled by the OS. The user will have to grant explicit permission to third-party applications requesting use of cost sensitive APIs. These APIs include:

* Telephony
* SMS/MMS
* Network/Data
* In-App Billing
* NFC Access

Android 4.2 adds further control on the use of SMS. Android will provide a notification if an application attempts to send SMS to a short code that uses premium services which might cause additional charges. The user can choose whether to allow the application to send the message or block it.

## SIM Card Access

Low level access to the SIM card is not available to third-party apps. The OS handles all communications with the SIM card including access to personal information (contacts) on the SIM card memory. Applications also cannot access AT commands, as these are managed exclusively by the Radio Interface Layer (RIL). The RIL provides no high level APIs for these commands.

## Personal Information

Android has placed APIs that provide access to user data into the set of protected APIs. With normal usage, Android devices will also accumulate user data within third-party applications installed by users. Applications that choose to share this information can use Android OS permission checks to protect the data from third-party applications.

## Sensitive Data Input Devices

Android devices frequently provide sensitive data input devices that allow applications to interact with the surrounding environment, such as camera, microphone or GPS. For a third-party application to access these devices, it must first be explicitly provided access by the user through the use of Android OS Permissions. Upon installation, the installer will prompt the user requesting permission to the sensor by name.

If an application wants to know the user's location, the application requires a permission to access the user's location. Upon installation, the installer will prompt the user asking if the application can access the user's location. At any time, if the user does not want any application to access their location, then the user can run the "Settings" application, go to "Location & Security", and uncheck the "Use wireless networks" and "Enable GPS satellites". This will disable location based services for all applications on the user's device.

## Device Metadata

Android also strives to restrict access to data that is not intrinsically sensitive, but may indirectly reveal characteristics about the user, user preferences, and the manner in which they use a device.

By default applications do not have access to operating system logs, browser history, phone number, or hardware / network identification information. If an application requests access to this information at install time, the installer will prompt the user asking if the application can access the information. If the user does not grant access, the application will not be installed.

**Attacks:**

Attacks can be classified into two types :

1 – Passive (For Reconnaissance)

2 – Active (For Exploitation)

**Examples :**

1 – Remote Access Trojans

* Enables remote administrative control for Reconnaissance and attack.
* Monitor user behavior through key loggers and spywares.
* Activate webcam and record audio.
* Format drives
* Obtain access / login information of the victim.

**2 – Network attack**

* Occurs when an android device is connected to wifi or a malicious hotspot

**3 – Malicious file attack**

* Is a piece of code that is execute behind the scenes as an “Exploit” to some vulnerabilities in the victim’s system.
* This allows attackers to perform Remote code execution
* Such as : patches , fake links , encrypted text files and Fake images.

**4 – Root Permissions**

* on Android only the kernel and a small subset of the core applications run with root permissions. Android does not prevent a user or application with root permissions from modifying the operating system, kernel, or any other application.
* In general, root has full access to all applications and all application data.
* Users that change the permissions on an Android device to grant root access to applications increase the security exposure to malicious applications and potential application flaws