Android security issues

* In the latter part of 2010 and early 2011, two vulnerabilities were discovered in Android versions 2.2 and 2.3, respectively. The vulnerability is essentially the same one, in which an attacker can copy any file that is stored on the device’s SD Card without permission or even without a visible cue that this is happening. The user might open a html file, this file is then downloaded to the SD card, the html files contain JavaScript code that uploads the files in the SD card without any permission. [1]
* If applications run as root then they have access parts of the OS, the entire device and all its contents are at risk.
* The idea for android being open source itself is a problem, attackers can analyze each line of code to determine its weaknesses.
* Google play store is a bit of concern because of the relative ease of getting apps approved for sale in the Play Store, malware apps can squeak through.
* Third-party android stores: like Aptoide.
* In Android, other than google play store, it is possible to install the applications from unknown sources. But, in iOS, the apps can be only installed from AppStore. It is one of the major security breaches in Android. Due to various security breaches in Android, attackers already regard smartphone as the target to steal personal information using various malware.[2]
* In 2013, Mohd Shahdi Ahmad et al. indicated the analysis of Android and iOS regarding security and declared iOS more secure than Android. In 2014, A. Kaur et al. indicated that it is possible to revoke granted permissions from android application.[2]

**SECURITY ATTACKS IN ANDROID** [3]

1. **Permission Escalation Attack**

It allows a malicious application to collaborate with other applications so as to access critical resources without requesting for corresponding permissions explicitly.

1. **Collision Attack Android**

It supports shared user ID. It is a technique wherein two or more application share the same user id so that they can access the permissions which are granted to each other. For example. If application A has permissions to READ\_CONTACTS, READ\_PHONE\_STATUS and B has permissions to READ\_MESSAGES, LOCATION\_ACCESS, if both the applications use the same user id SHAREDUSERID, then it is possible for application A to use the permissions granted to itself and the permissions granted to B. Similarly, it is possible for application B to use the permissions granted to itself and the permissions granted to A. Every Android application has unique ID that is its package name. Android supports shared User ID. It is an attribute in AndroidManifest.xml file. If this attribute assigned with the same value in two or more applications and if the same certificate signs these applications. They can access permissions granted to each other. Collision attack has been classified as direct collision attack and indirect collision attack. A direct collision attack is wherein application communicates directly. In Indirect collision attack application communicates via third party application or component.

C. **Time of Check and Time of Use Attack**

The main reason for TOCTOU Attack is naming collision. No naming rule or constraint is applied to a new permission declaration. Moreover, permissions in Android are represented as strings, and any two permissions with the same name string are treated as equivalent even if they belong to separate applications.

D. **Spyware**

Spyware is a type of malware. It is an apk file which is downloaded automatically when the user visits malicious website and apps installed from unknown sources. In Android, other than google play store, it is possible to install the applications from unknown sources. Spyware is one of the main reasons for major security threats in Android operating system.

**Permissions** [3]

The Android operating system uses the permission-based model to access various resources and information. These permissions are not requests; they are declarations. These permissions are declared in AndroidManifest.xml file. Once the permissions are granted, the permissions remain static for Android versions less than 6 [8][9]. But, in Android versions, 7.0 and higher the app permissions are classified into normal permissions [10] and dangerous permissions.

**Dangerous Permissions**

Dangerous Permissions can access critical resources of the mobile. Dangerous permissions can give the app access to the user's confidential data. If app lists a normal permission in its manifest, the system grants the permission automatically. If app list a dangerous permission, the user has to explicitly give approval for the app for the successful installation of the app. Example:

CONTACTS

READ\_CONTACTS, WRITE\_CONTACTS,

GET\_ACCOUNTS

LOCATION

ACCESS\_FINE\_LOCATION,

ACCESS\_COARSE\_LOCATION

SMS

SEND\_SMS, RECEIVE\_SMS, READ\_SMS,

RECEIVE\_WAP\_PUSH, RECEIVE\_MMS

STORAGE

READ\_EXTERNAL\_STORAGE,

WRITE\_EXTERNAL\_STORAGE

# Root: it’s parent directory. To illustrate this, we will take PCs as example, in PCs there is physical partitions (C, D, etc...), partition C root is C:/, D partition root is D:/.

# Partitions in android are directories under the root. In windows you can reach partition using the drive name followed by ‘:’ (ex : c:/) , in android : (/drive-name).

**References**:

[1] <https://www.informationweek.com/mobile/8-android-security-concerns-that-should-scare-it/d/d-id/1319412?image_number=1> October , 25th , 2017

[2] <http://ieeexplore.ieee.org/document/7975551/figures?part=1>

[3] <https://www.researchgate.net/publication/318412307_Android_security_issues_and_solutions>