Finding Trojan : Method 1.
In my project it is important to Identify an app is Trojan or not.
One way is by checking the permissions.

Dangerous permissions ::
https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous

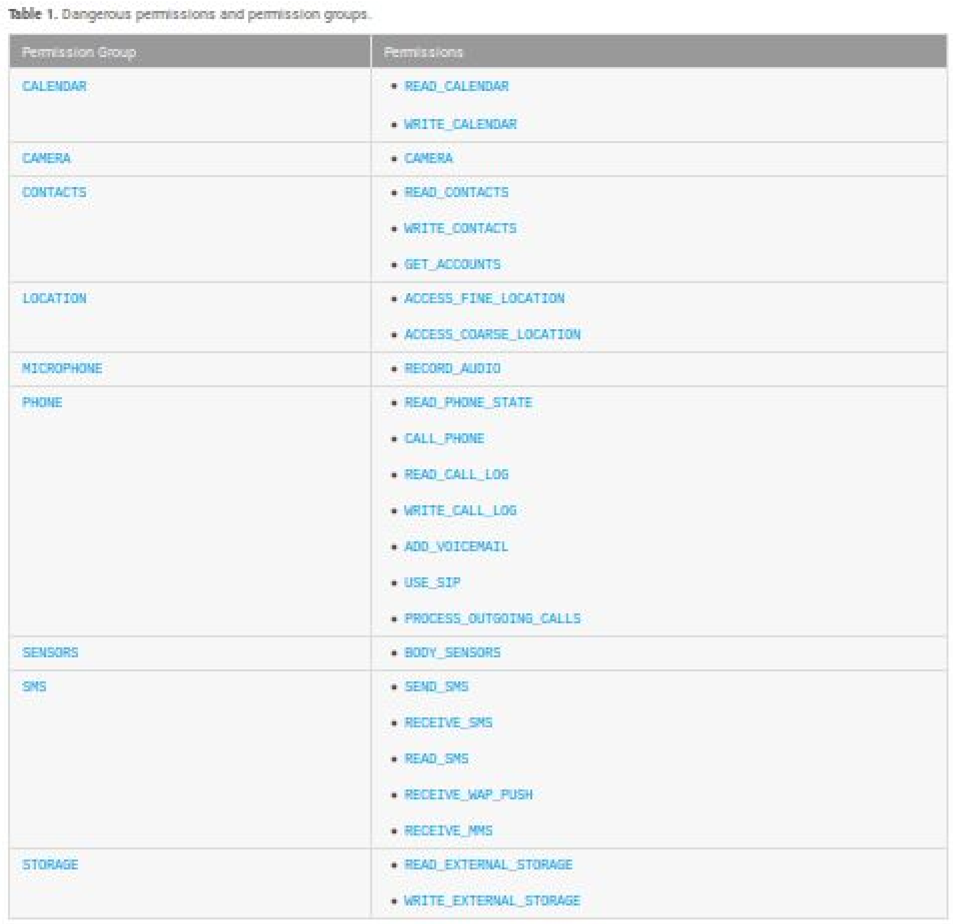

Table 1. Dangerous permissions and permission groups.

Permission Group	Permissions
CALENDAR	• READ_CALENDAR • WRITE_CALENDAR
CAMERA	• CAMERA
CONTACTS	• READ_CONTACTS • WRITE_CONTACTS • GET_ACCOUNTS
LOCATION	• ACCESS_FINE_LOCATION • ACCESS_COARSE_LOCATION
MICROPHONE	• RECORD_AUDIO
PHONE	• READ_PHONE_STATE • CALL_PHONE • READ_CALL_LOG • WRITE_CALL_LOG • ADD_VOICEMAIL • USE_SIP • PROCESS_OUTGOING_CALLS
SENSORS	• BODY_SENSORS
SMS	• SEND_SMS • RECEIVE_SMS • READ_SMS • RECEIVE_WAP_PUSH • RECEIVE_MMS
STORAGE	• READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE

Dangerous permissions are listed above. But all the above permissions can't be malicious, for example if it is   banking app it will need RECIVE_SMS permission to verify 2 factor authentication, but if the app is camera app it do not need the sms permission.
So here it is important to understand what type of app user intent , what user is believing, to use. The intention of user can be clarified by crowd sourcing, so while installing an app, users can answer what type of app is that (banking, gaming, camera, text editors etc).

In the cloud server we can check the permissions of the app against the list of dangerous permissions. But it is not enough, as I mentioned for an banking app it may need sms permission for authentication. So it is important to have more detailed analysis of permissions.

Here we will have a three data sets of permissions.

Data set 1; Expected permissions:

Here we have different set of permissions lists based on type of application, such as banking, gaming, text editors, camera etc. This permissions can be said as expected permissions of the app, such that this permissions are enough for a particular type app to works.

While crowdsourcing user can select what type of app is installing, based on that we will categories app to a group.

Data set 2; Actual permissions:

Here we will reverse the app and list out the actual permissions of the app.

Data set 3; Permissions based on Trojan apps

This data set have permissions needed for different type of Trojans to execute. The permissions need for each Trojan is categorised as different lists.This data set can be collected by analsing different Trojans.

Testing :

Permissions of data set 1 negation permissions of data set 2 will check against all data set 3. (p(set1) - p(set2)) / forEach(p(set3)) gives the rate of malicious permission in a particular app.

- p -> permissions
- set1 -> Expected permissions set
- set2 -> Actual permissions set
- set3 -> Trojan permissions set
- forEach -> each permission set of Trojans

This method may cause so many false positives.