Trust Based Portable Security Model For Android Based Devices

Santosh Varshney

Abstract: Android based mobile phones are widely used by variety of users the security of these phones is of major concern and is a topic of research from the beginning. The security of Android phone is based on the permission based security model which requires that any third party application must get the permission before using a resource. This model is broadly criticized for its complexity, coarse-grained access to resources and inappropriate authorization of permissions by application developers, advertisers and end users. This paper analyses the issues with current security model and propose a trust based portable security model for Android based devices. The proposed model provides effective security with ease. The proposed model does not require user intervention once the trust levels are set for an application. The trust levels can be given a logical name representing its nature so they are easy to understand especially by a novice user who has very little knowledge about the technology. Further for each logical trust level there is a value associated this value is accumulation of all trust values of resources which are grouped under the trust level. The permission grant function takes trust level and resource trust value and grant permission only if given resource trust value contributes to the given trust level. The trust levels can be preconfigured by experts and can be reconfigurable by an expert user, the trust level configurations and mapping can securely be ported to another device.

Index Terms: Android Security, Permission Based Model, Trust Level, Resource Trust Value, Permission Grant Function, Application Permissions, Manifest Declared Permissions, Coarse Grained Security

1 INTRODUCTION

In today’s worlds the technology like smart phone is something which became a part of life of a common man irrelevant of his work area education or age but this use of a smart phone has created new challenges for the security of the user information. There is a requirement for a security model which should be suitable for all ranging from a novice user to expert user. The Android based smartphones are used by a major percentage of population all over the world in this paper the permission based security [1] is explored and suggested a trust based security model. This model is easy to understand by novice users and has the scope of customization by experts or power users. A novice user can simply set the trust label for an application and an expert user may customize the predefined trust levels as per his requirement this add an flexibility with ease at end user label. The major problem with current permission model is it asks permission for each individual resources like for SMS, Location, Contacts etc. which a novice user may not completely understands and he will be in dilemma to give permissions for the resources he may not understand completely. One example of such permission is “android.permission PACKAGE_USAGE_STATS” , using this permission an app can find out which app is active and it may misuse to overlay its screen to get sensitive information from user. We propose a very simple trust based security model which is simple and easy to use by a novice user and has the flexibility to configure by an expert user for his customize needs. This model not only solves issue but provide an easily understandable model by everyone, here everyone is a wide variety starting from a novice user to technology experts. The logical names for the preconfigured trust levels can be chosen such that they reflects the exact meaning and purpose so just by looking at trust level name one can easily guess that whether the level is suitable for his targeted application or not. Further this paper is composed in the following parts first part contains introduction to existing Android permissions model, in the second part, we’ll briefly discuss about the issues with existing permission based application security, In the third part we discuss the concept of trust based model and its implantation, next will finish up with conclusion and future work.

2 EXISTING SECURITY MODEL AND ISSUES

2.1 Existing permission based model

In Android 5.1.1 (API 22) or lower, the permission is requested at the installation time. Starting from Android 6 (API 23) or higher, the permission is requested at the run time during the running of the application. Before Android 5.1.1 (API 22) it is only required to declare the permission in Android manifest file but starting from Android 6 (API 23) it is required to get the permission explicitly at run time. For getting permission at run time Android provided an API “ActivityCompat.requestPermissions” which takes an array of required permission and it prompt to the user for those permission and conveys the result by another call back API. However asking permission at run time give some relief to the user still it does not address lot of issues with variety of users. In simple way this mechanism is depicted in Fig. 1.

In Android 5.1.1 (API 22) or lower, the permission is requested at the installation time. Starting from Android 6 (API 23) or higher, the permission is requested at the run time during the running of the application. Before Android 5.1.1 (API 22) it is only required to declare the permission in Android manifest file but starting from Android 6 (API 23) it is required to get the permission explicitly at run time. For getting permission at run time Android provided an API “ActivityCompat.requestPermissions” which takes an array of required permission and it prompt to the user for those permission and conveys the result by another call back API. However asking permission at run time give some relief to the user still it does not address lot of issues with variety of users. In simple way this mechanism is depicted in Fig. 1.

In the second part, we’ll briefly discuss about the issues with existing permission based application security, In the third part we discuss the concept of trust based model and its implantation, next will finish up with conclusion and future work.

In the second part, we’ll briefly discuss about the issues with existing permission based application security, In the third part we discuss the concept of trust based model and its implantation, next will finish up with conclusion and future work.

Fig 1: existing permission based model
2.2 ISSUES WITH PERMISSION BASED Model
Before Android version 4.3, the android permission model was static and was based on coarse granule permissions. For example, simply granting INTERNET permission to an application enables it to communicate via HTTP(S) connection, and interface with self-assertive connections and ports [2] which is an serious security threat to the user privacy. Thus the INTERNET permission does not provide sufficient information to the user and another view is not all users aware of seriousness of such permission due to their limited exposure or knowledge. Trust based security plays an ideal role here the novice users do not need to understand permissions individually like INTERNET they have to simply select say how much they trust an application so trust based model allows a better trade-offs between privacy and expertise. The greater part of the prior research is concentrated on making the permissions model fine grained as well as dynamic in nature. Probably the most conspicuous works done on this problem are Apex [3], Appguard [4], Flaskdroid [5], MockDroid [6] and TISSA [7]. Nonetheless, despite the fact that the idea trust based permission model is crucially needed, it has been critically overlooked by the vast majority of researches. There is crucial role of both end user and developer in the authorization process of an application [8] and it completely depends on the knowledge and perception of both of them. There is possibility of asking unwanted permissions due to the development and maintenance process of application code [9] for example a feature is removed but asking related permission code or manifest file is not modified so even the feature for which the permission required is removed the application will ask the related permissions. Trust based model overcome it by having the predefined trust levels which are well defined by the experts and user can simply select a trust level.

3 PROPOSED TRUST BASED PERMISSION MODEL
Proposed trust based security model is based on trust values which is associated with a resource. Resources can be physical like camera, GPS etc. or logical like contact list, SMS etc. both physical and logical resources are similar for this model. We can fine grain resources as much as possible and this can be configurable by an expert user. The permission can be associated with single resource or to a group of resources and permission is ultimately depends on the trust value of each resource and trust level. It is not necessary that trust values are unique for each resource, resources having common security concerns may be assigned same trust values. Trust level is the accumulated value of all the trust values of the resources grouped under that trust level. Trust levels can be defined by grouping the resources based on their sensitivity to security. For example all resources having nil concern to security can be grouped and we can say it has logical 0 trust level, here there are two things one is logical name of trust level it can be a name well explain that level another is accumulated value of all trust values of the resources in that group this is used internally by permission grant function. Suppose X represent the trust value associated with a resource i and there is N number of resources in this group then

\[
\text{Trust Level (TL)} = \sum_{i=1}^{N} X_i
\]

The logical name of trust level will be shown to user in a list of preexisting trust levels so he can choose from these preconfigured trust levels. The accumulated value will be used at the time of evaluating permission by the permission grant function. Permission grant function abbreviated as PGF() is a function which will take two parameters out of which one is Current Trust Level (abbreviated as TL) and other is Recourse Trust Value (abbreviated as RTV) of the resource for which the permission is asked. The implementation of permission grant function PGF() is to find out whether the resource trust value of given resource is contributing to the given trust level if yes then the permission will be granted it will be denied otherwise.

Permission Grant (Boolean) = PGF(TL, RTV)

So basically if the return value of permission grant function PGF() is true then then the permission will be granted and if return value is false then the permission will be denied. The Trust level is associated with an application by the user. An application may be associated with more than one trust levels. In such cases the permission grant function must be called for each trust level associated and if any permission grant function returns true then the permission is granted otherwise permission is denied. The whole process is well depicted in the below figure.

![Proposed Trust Based Permission Model](https://example.com/fig2.png)
Suppose there are n number of trust level associated with an application then the permission grant function is defined as below:

**Permission Grant (Boolean) = PGF(TL-1, TV) || PGF(TL-2, TV) || ... PGF(TL-n, TV)**

Where TL-1, TL-2, ..., TL-n are trust levels associated with the application and TV is the trust value of resource for which permission is asked. In android devices each resource has identification and is same in all Android devices, it is possible to create the profile of trust values and there association, trust levels defined and application trust level association. This profile can be exported to an encrypted file and can be imported in another Android device this way we can port the trust level configurations from one Android device to another.

![Diagram](Fig. 3: Architecture of Trust Based Permission Model)

In the architecture proposed the permission grant function can be implemented as middleware which has access to kernel layers by system API calls. The mapping of resource Id Vs Trust value, Application ID vs Trust level and configuration of trust levels may be stored in kernel layer for security so no other component can access this data without proper authentication and authorization. As the application request for a resource to use the request is directed to permission grant function (PGF) at middleware, PGF then retrieves the application Trust Level from application ID Vs Trust Level mapping data base, Trust Level Value from trust configuration database, trust value of resource from resource trust value data base and using Trust Level Value and resource Trust Value it find out the contribution of resource trust value to the application trust level and grant permission.

### 4 IMPLEMENTATION

Here a simple implementation using bitmask is demonstrated. The major limitation of this implementation is that it is not scalable. Use bitmask to define trust value of a resource, we will use a 16 bit binary number to illustrate our concept. Let us say we have three resources R1, R2 and R3 we can define the trust value of these resources as follows:-

\[
R1 = 0000000000000001 \\
R2 = 0000000000000010 \\
R3 = 0000000000000100
\]

Now suppose we group these three resources in a trust level say TL-1 then

\[
TL-1 = (R1 \| R2 \| R3) = 0000000000001111
\]

So 0000000000001111 is the cumulative trust vale of the TL-1. Note that here OR is doing the work of accumulation as defined in the model.

Now let us have 2 more resources say R4 and R5 Let us define trust values for these resources as

\[
R2 = 0000000000000100 \\
R3 = 0000000000010000
\]

Now define another trust level TL-2 which has these two resources in the group

\[
TL-2 = (R4 \| R5) = 0000000000011000
\]

So for we defined trust values for few resources and 2 trust levels TL-1 and TL-2. The important thing here is when we define trust value we should group resources comply to same level from security point of view, here we assumed that R1, R2 and R3 is not sensitive and R4 and R5 are sensitive from security point of view. Now permission grant function has to find out contribution of resource trust value to the given trust level, Here we can define it as bitwise AND function of both of them.

**PGF(current Trust Level, Trust Value of the resource) = <current Trust Level> AND <Trust Value of the resource>**

Obviously it will return true if trust value contributed to the trust level otherwise it will return false so it is exactly same as the model expects. Now suppose we have an app say App-1 which we do not trust so App-1 will assigned trust level TL-1 to it and there is another app App-2 which we trust then we will assign trust level TL-2 to it. Now if App-1 want to access resource R1 then we will calculate AND function between trust level of App-1 and trust value of resource R1, If the result of function AND is true the access is allowed otherwise not allowed. It can be clearly seen that if App-1 try to access the resource R4 then (TL-1 AND R4 ) will result to false and App-1 want be allowed to access R4. Same is the case for resource R5. Now we trust App-2 so we will associate TL-1 as well as TL-2 with it so that it can access all resources. It can be seen that if App-2 wants to access the resource R1 then the AND function between trust level TL-1 and trust value of resource R1 will result in true and will allow to access R1 and if App-2 wants to access R4 then AND function between trust...
level TL-2 and trust value of resource R4 will result in true and will allow to access R4. It’s not necessary that all resources have unique trust value if resources closely comes under one category like normal or dangerous then they can be assigned the same trust value. For example suppose R1, R2 and R3 all are normal resources never create challenge to security we can have
R1 = R2 = R3 = 0000000000000001

5 CONCLUSION AND FUTURE WORK
This paper first discussed the existing security model of android based on the permission and changes in the security model from static permission at installation time to run time dynamic permission. It then discussed the issues in current security model it is then discussed that current model expect everyone to interpret it similarly and this is an serious issue specially for novice users so we can say that current security model does not address effectively all range of users. A trust based portable security model is proposed where user can simply associate a trust level with an application. The trust level is easy to interpret by everyone including a novice user. The trust level is accumulation of trust values of each resources grouped under that trust level. The permission requested by an application is granted only if the resource trust value for which the permission asked is contributing to the trust level associated with the application. It may be possible to assign same trust value to multiple resources if they address the same security concern also more than one trust level may be associated with an application. The trust level configurations can be securely portable to another Android device. The implantation given in this paper is for illustration only it is not scalable. In future there is much scope to find out an efficient and scalable implementable and implement it in the Android source code and test the practicability of the concept.

6 REFERENCES