Survey on Enhancing Security using Android Permission Model

Mayura Devani[1] Tejas Bhatt[2]
[1]Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]Sardar Vallabhbhai Patel Institute of Technology, Vasad Gujarat Technology University, Gujarat, India

Abstract— More and more people rely on smart phones to manage their personal data. Smartphone's are today the repositories of our secrets (photos, email), of our money (online e-commerce) and of our identities (social networks accounts). Therefore mobile applications have the responsibility of handling such sensitive and personal information in a proper, secure way. For the protection of user data from third party applications, it is common for smartphone's Operating Systems to use permission model control the Permissions granted to third party applications. The user receives a dialog with the list of permissions requested by the application before installation. Once the application is installed, the user gives chance to third Party application developer to attack. Nowadays attackers are developing such a malicious application. Devices to various attacks. To protect against such attacks, different solutions are being given. We studied and detection of attacks focused on behavior-based approach. It is difficult for general users to permissions which are potentially harmful and those which are not. We are proposing a scheme to the user, so that he/she can give access to the requested permissions, or can deny and finish the installation.

Key words: Android Security, Permission Model, Android Operating System

I. INTRODUCTION

Smart phones are growing. For many people, a smartphone, not only because of its traditional phone capabilities, but also because of its smart features has become a constant companion. Those, along with a calendar application program management, web surfing, or involve the use of location-based services. By all means, several smart phones are used for holding personal data. Are using the smart phone, to conduct criminal activities amongst which are those that are not a surprise. For investigators, the data stored on the smart phone to resolve a criminal case is likely to contain important evidence.

Increase in the level of comfort and convenience of smartphones, smartphone causes a significant increase in the number of users. For example, the phone calls are about within formation call log, the user has made, these are all personal information, it is not that they do history etc., photo browsing, the users that includes contact get an address book fall into the wrong hands. The malicious user from apps of smartphones to protect the personal information, privacy, a new mode of access a user's personal information, you can modify the apps need smartphones. This new modes. This information is accessible to the app in a way that the use of personal information for the user to control and which cannot be possible. In addition, to allow the user to modify previously granted must be time control Run.

We have studied the related papers and provide an outline of the papers studied.

A. Architecture of the Android Operating System:

In many versions of Android yet, 4.0 Ice Cream Sandwich is the latest of which have been released, although the core architecture remains the same. The base Linux kernel, libraries, applications for managing a virtual machine to run an application framework consisting of applications and with the layers is a layered architecture.

Android application is made up of different Application components. These components are the different entry points for the system. There is no strict or set policy defined governing where a component can actually enter the system. There isn't even a main() function defined for application processes. For user, each and every component doesn't serve as entry point. Components can be inter-dependent, but every component has a specific role and has its own entity. The application's overall behavior is depending on these building blocks i.e. components. Different types of components are as follows;

Fig. 1: Android OS architecture

1) Application Framework:

Application framework layer defines how applications are built and how they behave. The Android application framework includes [1]:

- lists, grids, text boxes, buttons, and embedded web browser, which is made by including the ideas for the application user interface

- to share your data with other applications, which enable applications to access data a set of content providers.
- A notification manager
- Access to resources provided by the resource manager's performance enables custom alerts
- A transaction manager for the application and application life cycle also manage provides common navigation stack[2].

2) Android Security:

In Android, the specific identities of each application are different parts of the system or a different system Id. Some Linux User ID and group identity. Other applications in Linux systems are isolated from each. Additional finer grained security features to allow the system is used. Using this mechanism, restrictions on specific tasks that can perform a special procedure.

Android system to allow up to four security level 0 to 3:

Level zero (0) is called the normal permissions and typically these permissions affect only the application scope. without any permissions the user's explicit approval, system automatically is given by. Alternatively, the user is prior to installation of the application, request for permission requests can be notified.

Level one (1) permissions to allow as high risk, are a phone call or an on-level sensor device, the Internet, or allow access to sensitive user data. An interesting really allowed to use the device for initiating services such as Access is allowed to read files. The installation package installer and user of the grant or deny permissions set permissions for all his consent is requested, the application can be installed successfully or if the user decides, to the request dangerous set of permissions displays.

Level two (2) permissions are granted. Other developers to prevent applications to gain access to this information, these so-called signature permissions to share their personal information between application developers, such as, can be used by. Users consent, so even if signature permission signed with the private key corresponding to the application to another certificate can not be.

Level three (3) permissions, or system image as the signed with the same certificate that applications are contained in the system image that can be given to the application system. The highest category permissions manufacture the user and OS developers are reserved for a handset. The class representative's new application (package) to set or change security settings allow.

Processes isolation, memory management implementation and threading are depend on the Linux Kernel [2].

II. LITERATURE SURVEY

A. Mockdroid:

Mockdroid is the model which mocks the data given to the application, if permission is not given to access that data [3].

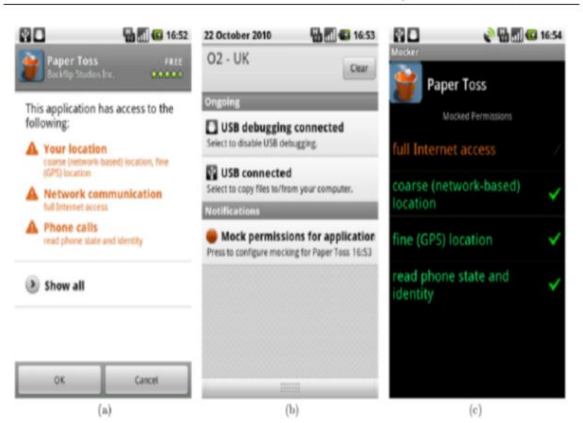

Fig. 2: Paper Toss: (a) installation asks for unnecessary permissions; (b) the user is notified via an unobtrusive notification if a mocked permission is used; (c) permissions that are can be modified[5]

During the installation of a new app from the market even though they may be user dangerous, is used to display all the permissions to allow them. Data is then copied to disk, which is still in memory are stored in a data structure that has been allowed to set permissions fails otherwise. Permission is granted to appropriate API call package that requires permission checks every API call. Without a permit by an app attempt is appropriate to use an API call, then throws an API package runtime exception.[4]

In the package manager has had some modifications Mockdroid and allow each to maintain a realistic and mocked version has been set so as to allow repeated. Are requesting that all the permissions are allowed and no one gets hurt yet. Mockdroid Android OS has been added, which is called fake an additional UNIX group uses.

Monitor the files in the directory, allowing it to be mocked and turned if any notify kernel service uses to see. Change the contents of the file in memory of and when mocked as allowing changes to update the cache.

Internet-related permissions are supported in a different way from other Android API, the kernel API calls from the Internet and is implemented through the virtual machine because the points are dangerous. Standard Android access the Internet wanting a process that checks the Cabinet belongs to the group. The Internet has allowed an application, the Android Activity Manager is the process of applying for Cabinet Group.[6][7]

B. Sorbet:

The model named Sorbet generalizes Android-style permissions and instantiate the current permission system of Android [8].

The Android currently does not follow some of the desired security properties with the help of the model is shown. Coarse-grained information flow to support sorbet Android describes a set of system improvements and Privilege-growth policies.

As the run-time component inventory data, it is another example of the run-time components, run-time and run-time instances is important to distinguish between. IC is a static component C several run-time instances. There are clear rules in shape, but [6] proposed a model in terms of the desired security properties can be obtained from the description section 2.2

1) Local Call Property:
If a component is called by another component B, then it is evaluated as true.
2) Local Caller Property:
Caller evaluates a component as a guard for the truth, and component B calls.
3) Delegation:
a P owns, or if there is such a component B is a component of a permit delegation chain P that satisfies the scope and range of a lifetime constraints imposed by each component and each component on the chain also has P .- a drink or drink is provided by the owner of a user; or- ends with a Saint in the series that has a delegation chain B (where B = A): BP, and the scope and lifetime of a satisfied constraints imposed by the B, - and P A. has not terminated
4) Cancellation:
P revokes a B, then B is a chain of delegation.
5) Revocation:
If A revokes P from B, then there is a delegation chain from A to B, or A owns P.
6) Privilege Escalation:
Given any component B protected by permission P, and any component A that does not have that permission, if SAB is a system that contains A and B(and other components), and SB is the same system without A, then a call chain ending with B exists in SAB if and only if it exists in SB . Additional call chains ending with Bmay exist in SAB if explicitly allowed by policy.
7) Information Flow:
Given any component B protected by permission P , and any component A that does not have that permission, if SAB is a system that contains A and B (and other components), and SB is the same system without A, then a call chain ending with B exists in SAB if and only if it exists in SB . Additional call chains ending with B may exist in SAB if explicitly allowed by policy. Given an undesired information flow from a component A guarded by P1 to a component B guarded by P2 , a call chain that ends with B exists in a system with A if and only if the same call chain exists in a system without A. Additional call chains ending with B may exist in the system with A only if explicitly allowed by policy.[9]

C. Tissa:

The goal of the TISSA [10] is to design a model which can prevent private information leakage by untrusted smartphone applications effectively and efficiently. TISSA has the following design requirements, in order to provide the protection to the private information with a considerable system performance, better user experience, and application compatibility.[11]

- Small Footprint: The necessary changes to the Android framework should be a minimum.
- Application Transparency: The model may not change the APIs provided by the default Android framework and should maintain the compatibility of the existing Android.

Read private data, the application sends a request to the provider. This is the response from the content provider privacy settings privacy settings until the content provider and the content provider is fired a query, the application has requested. Privacy setting content provider, for its internal policy database by querying, privacy settings for a specific application will fetch the user's specifications. The results of the read operation are allowed, then request access provider. The provider returned to work the material, and the application returns to the normal results. The operation is not allowed to read the privacy settings, content providers; it may indicate possible ways to handle.

III. COMPARISON

Mockdroid can be successfully integrated in TISSA. The difference between Mockdroid and TISSA is that the permission which is mocked in Mockdroid can successfully be rolled back and legitimate permissions can be given while using the application which cannot be done in TISSA.

Mockdroid formal verification and integration of sorbet with the permissions of the concepts suggested by Tissa first with the model can be modified in order to find the permissions for applications that are used most often can be are used.[12]

IV. FUTURE WORK

TISSA is just a prototype. It is still not available for Android users. So An Open Source Model for Android users is needed. We can develop a Model, which enhance android permission model, and allow user to grant permission at runtime of an application.

REFERENCES

[1] http://stackoverflow.com/questions/2968016/android-framework-what-is-it
[2] https://source.android.com/devices/tech/security/overview/app-security.html#digital-rights-management
[3] http://timesofindia.indiatimes.com/tech/tech-news/Xiaomi-phones-send-user-data-to-remote-servers-F-Secure/articleshow/39950622.cms
[4] http://www.makeuseof.com/tag/app-permissions-work-care-android/
[5] http://blog.trendmicro.com/trendlabs-security-intelligence/bypassing-android-permissions-what-you-need-to-know/
[6] https://www.hackinparis.com/slides/hip2k12/Georgia-androidpermissions.pdf
[7] http://news.softpedia.com/news/Softpedia-Exclusive-Interview-Georgia-Weidman-on-the-Android-Permission-Model-277204.shtml
[8] http://www.itnews.com.au/News/285124,android-app-installs-shell-bypasses-permissions.aspx
[9] http://www.windowsecurity.com/blogs/chetcuti/vulnerabilities/permission-bypass-vulnerability-android.html
[10] http://resources.infosecinstitute.com/android-app-permissions-security-need-know/
[11] http://web.cs.ucdavis.edu/~hchen/paper/msr2013.pdf
[12] http://privacy-pc.com/articles/bypassing-the-android-permission-model-7-exploiting-open-interfaces-to-steal-permissions.html
http://www.leviathansecurity.com/blog/zero-permission-android-applications/

