Usage of Access Control Mechanism for blocking Application in Android Mobile Phones

Divya S
Assistant Professor, Department of CSE, RVS Technical Campus-Coimbatore, India.

Vanitha S
UG Scholar, Department of CSE, And RVS Technical Campus- Coimbatore, India

Dakshana V M
UG Scholar, Department of CSE, And RVS Technical Campus- Coimbatore, India

Sharmila N
UG Scholar, Department of CSE, And RVS Technical Campus- Coimbatore, India

Abstract – Mobile Android applications often have access to sensitive data and resources on the user device. Misuse of data by malicious applications may result in privacy breaches and sensitive data leakage. In order to overcome such privacy problems an access control mechanism is proposed.

Application requires installation and the admin registration takes place. The location for blocking is saved with the package name of the application installed in the user's device. Employee is added for user login, once added the abstract is shown in the admin device. If the admin enables the blocker immediately the user's application is blocked. This may avoid privacy data breaches.

Index Terms – Android, Wi-Fi, Smart phone device, JSON parser, etc.

1. INTRODUCTION

As smart phones are becoming more powerful in terms of computational and communication capabilities, application developers are taking advantage of these capabilities in order to provide new or enhanced services to their applications. For example, on March 2013 Samsung unveiled its Galaxy S4 device with 8 CPU cores and 9 sensors that enrich the device with powerful resources. However, the majority of these resources can collect sensitive data and may expose users to high security and privacy risks if applications use them inappropriately and without the user's knowledge.

The threat arises when a device application acts maliciously and uses device resources to spy on the user or leak the user's personal data without the user's consent. Moreover, users carrying their smart phones in public and private places may unknowingly expose their private information and threaten their personal security as they are not aware of the existence of such malicious activities on their devices.

Since such a feature is still missing in popular smart phone systems, such as in Android systems, it is crucial to investigate approaches for providing such control to device users. The need for configurable device policies based on context extends from high profile employees to regular smart phone users.

For example, government employers, such as in national labs, restrict their employees from bringing any camera-enabled device to the workplace, including smart phones, even though employees might need to have their devices with them at all times as their devices may contain data and services they might need at any time. With context-based device policies, employees may be allowed to use smart phones as they can disable all applications from using the camera and any device resources and privileges that employers restrict while at work, while the user device can retain all its original privileges outside the work area.

Context-based policies are also a necessity for politicians and law enforcement agents who would need to disable camera, microphone, and location services from their devices during confidential meetings while retaining these resources back in non-confidential locations. With context-based policies, users can specify when and where their applications can access their device data and resources, which reduces the hackers' chances of stealing such data.

2. SYSTEM ARCHITECTURE

Figure consists of an access control mechanism that deals with access, collection, storage, processing, and usage of context information and device policies. To handle all the aforementioned functions, framework design of the figure consists of four main components as shown in figure Location Source collects the physical location parameters (GPS, Cell

IDs, Wi-Fi parameters) through the device sensors and stores them in its own database, linking each physical location to a user-defined logical

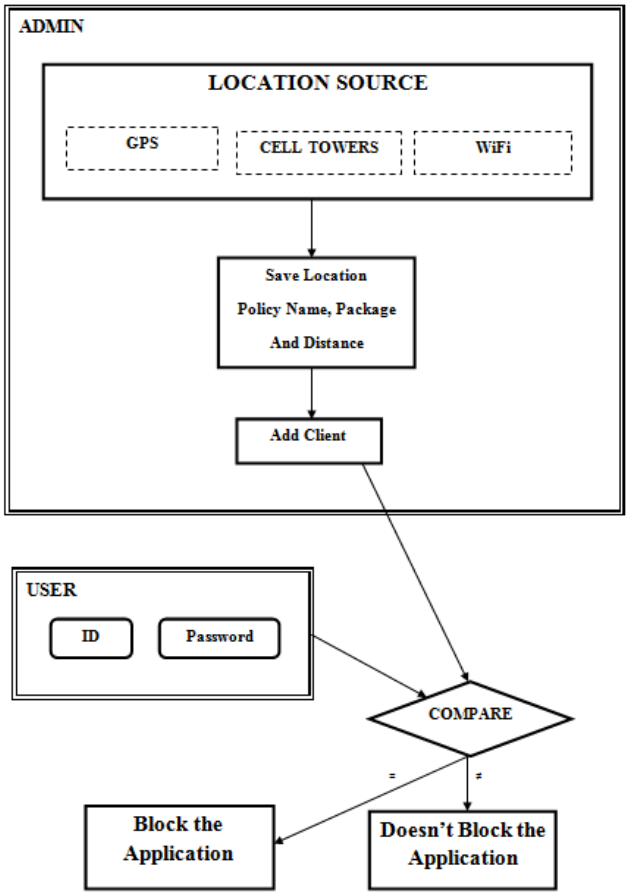


Fig.1 App blocker architecture

It also verifies and updates those parameters whenever the device is re-located. It controls the authorizations of applications and prevents unauthorized usage of device resources or services. Even though the Android OS has its own permission control system that checks if an application has privileges to request resources or services, the access controller complements this system with more control methods and specific fine-grained control permissions that better reflect the application capabilities and narrow down its accessibility to resources.

The admin(access controller)enhances the security of the device system since the existing Android system has some permissions that, once granted to applications, may give applications more accessibility than they need, which malicious code can take advantage of. For example, the permission READ PHONE STATE gives privileged applications a set of information such as the phone number, the IMEI/MEID identifier, subscriber identification, phone state (busy/available), SIM serial number, etc.

The client (policy manager) represents the interface used to create policies, mainly assigning application restrictions to contexts. It mainly gives control to the user to configure which resources and services are accessible by applications at the given context provided by the Location Source. As an example, the user through the Policy Manager can create a policy to enable location services only when the user is at work during weekdays between 8 am and 5 pm.

The user (policy executor) enforces device restrictions by comparing the device's context with the configured policies. Once an application requests access to a resource or service, the Policy Executer checks the user-configured restrictions set at the Policy Manager to either grant to deny access to the application request. The Policy Executer acts as policy enforcement by sending the authorization information to the Access Controller to handle application requests, and is also responsible to resolve policy conflicts and apply the strictest restrictions. Through the Policy Manager, users can create CBAC policies through configuring application restrictions and linking them to contexts. When an application requests a resource or service, the admin (access controller) verifies at run-time whether the application request is authorized and forwards the request to the Policy Executer.

If the request is authorized, the user (policy executer) then checks if there is any policy that corresponds to the application request. If such a policy exists, the Policy Executer requests from the Location Source to retrieve the context at the time of the application request. The Policy Executer then compares the retrieved context with the context defined in the policy. In case of a match, the Policy Executer enforces the corresponding policy restrictions by reporting back to the Access Controller to apply those restrictions on the application request.

The carefully design the access control framework so that the user-configured policies are securely enforced with minimal processing steps and execution time to avoid any significant delays in responding back to the requesting application. As our design should securely handle policy execution, maintain the context data provided by the Context Provider to make sure it is accurate, precise and up-to-date.

3. OVERVIEW

3.1 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a

written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

3.2 Objectives

Input Design is the process of converting a user-oriented description of the input into a computer-based system. The input design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

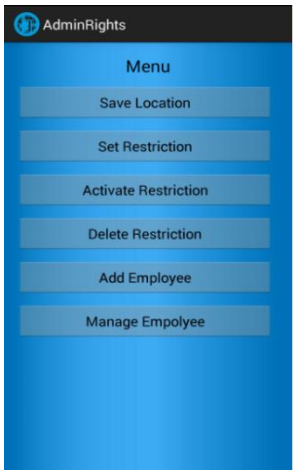
3.3 Output Design

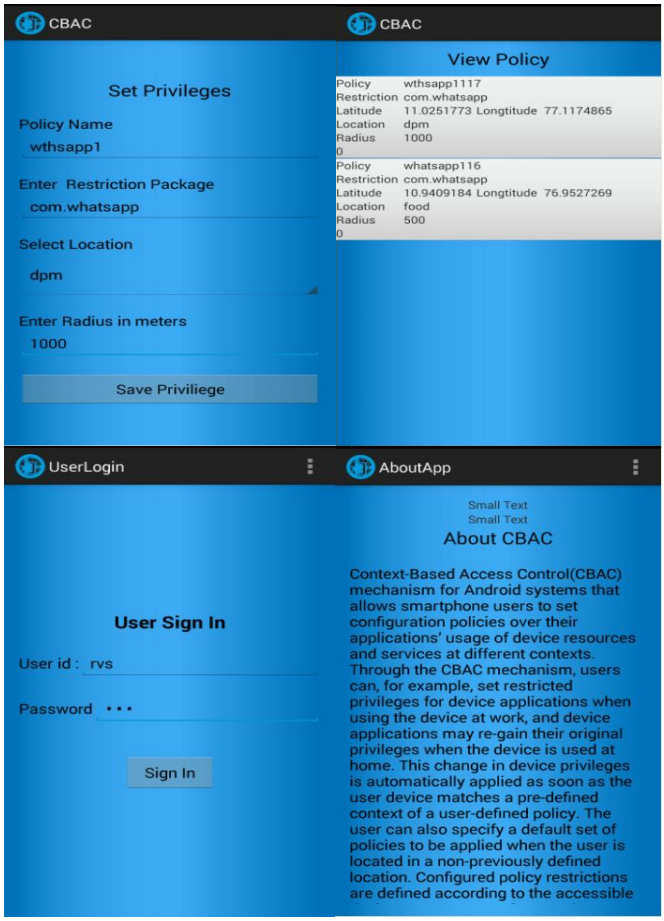
A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.





4. CONCLUSIONS

The modified version of the Android OS supporting context-based access control policies. These applications restricts the accessing of specific data and/or resources based on the user context. The restrictions specified are automatically applied as soon as the user input matches the pre-defined context associated. The experimental result shows the effectiveness of Android system and applications within a user-defined context.

5. FUTURE ENHANCEMANT

The approach requires users to configure their own set of policies the difficulty of setting up these configurations require the same expertise needed to inspect application permissions listed at installation time. However we plan to extend the approach to give network administrators of organizations the same capabilities once a mobile device connects to their network. Network administrators are able to block malicious application accesses to resources and services that may affect the security of their network. Approach is critical for assuring security of corporate networks when organizations allow users to "bring their own devices".

REFERENCES

[1] Wikipedia, "Samsung galaxy s4 specifications," http://en.wikipedia. org/wiki/Samsung Galaxy S4, May 2013.
[2] J. Leyden, "Your phone may not be spying on you now – but it soon will be," http://www.theregister.co.uk/2013/04/24/kaspersky mobile malware infosec/, April 2013.
[3] L. L. N. Laboratory, "Controlled items that are prohibited on llnl proporty," https://www.llnl.gov/about/controlleditems.html.
[4] A. Kushwaha and V. Kushwaha, "Location based services using android mobile operating system," International Journal of Advances in Engineering and Technology, vol. 1, no. 1, pp. 14–20, 2011.
[5] S. Kumar, M. A. Qadeer, and A. Gupta, "Location based services using android," in Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications, ser. IMSAA'09, 2009, pp. 335–339.
[6] P. under submission, "Identidroid: Android can finally wear its anonymous suit."
[7] W. Enck, M. Ongtang, and P. McDaniel, "Understanding android security," Security Privacy, IEEE, vol. 7, no. 1, pp. 50–57, 2009. [13] E. Trevisani and A. Vitaletti, "Cell-id location technique, limits and benefits: an experimental study," in Mobile Computing Systems and Applications, 2004. WMCSA 2004. Sixth IEEE Workshop on, 2004, pp. 51–60.
[8] J. LaMance, J. DeSalas, and J. Jarvinen, "agps: A low-infrastructure approach," http://www.gpsworld.com/ innovation-assisted-gps-a-low-infrastructure-approach/, March'02.
[9] "Sky hook," http://www.skyhookwireless.com/.
[10] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: retrofitting android to protect data from imperious applications," in 18th ACM conference on Computer and communications security, ser. CCS '11, NY, USA.
[11] I. F. Progri, "Wireless-enabled gps indoor geolocation system," in Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION, 2010, pp. 526–538.
[12] C. Feng, W. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," Mobile Computing, IEEE Transactions on, vol. 11, no. 12, pp. 1983–1993, 2012.
[13] S. Ali-Loytty, T. Perala, V. Honkavirta, and R. Piche, "Fingerprint kalman filter in indoor positioning applications," in Control Applications, (CCA) Intelligent Control, (ISIC), 2009, pp. 1678–1683.
[14] A. S. Paul and E. Wan, "Rssi-based indoor localization and tracking using sigma-point kalman smoothers," Selected Topics in Signal Processing, IEEE Journal of, vol. 3, no. 5, pp. 860–873, 2009.
[15] M. Moyer and M. Abamad, "Generalized role-based access control," in Distributed Computing Systems, 2001. 21st International Conference on., 2001, pp. 391–398.