ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
Research Article
www.ijptonline.com

CONTEXT BASED ACCESS OPTIMIZING SYSTEM FOR ANDROID MOBILE DEVICES

[1]E Koteswara Rao*, [2]R.Vijayan
M.Tech IT (Networking), VIT University, Vellore, India.
Asst. Professor [SG], SITE, VIT University, Vellore, India.
Email: koti0514@gmail.com

Received on 09-08-2016 *Accepted on 05-09-2016*

Abstract

The Context based Access Optimizing system is the mechanism which grants application access permissions dynamically when the applications requests for system resources. This mechanism helps all the android users to restrict their application permissions towards users device. In normal android systems when the application requests for a service first it checks in the previously issued permissions and based on that it'll grant the permissions, Sometimes it may lead users data expose to unauthorized usage. This can be solved by restricting the mobile applications to certain level by restricting their access to system resources. To work on this few modifications to be done in the android operating system according to the user strategies to be considered dynamically. The existing system is not good enough to restrict the applications and it is not possible to restrict the applications dynamically based on the users location, these are done in context based access optimizing system. To control the device dynamically user need to capture its location every time, for this GPS, cellular ID are not accurate enough to point the device location in the sub areas. To overcome this Wi-fi Based positioning technique used here to find the device location. Whenever user changes the location according to wi-fi based positioning technique it captures and checks the user predefined strategies and applies them according to the location.

Keywords: Context Based Access, Access Optimizing, Smartphones, Android Devices, Android Applications, Security and Privacy risks.

1. Introduction

The usage of modern smart phones are increasing rapidly with more advanced features, processors and computational capabilities. Almost in every one's life smart phones became an integral part. Some of the developers looked into this advantage and using all these features they are building applications. An example of this is Samsung released its

Galaxy S5 with almost 9 CPU cores and 10 sensors which increases the device capabilities and other computational resources[1]. If applications use those features and processors properly it'll be good, if not this will collect sensitive data from users device and expose to security risks which cause malicious behaviour. Threats will arise when the malicious applications use mobile resources to spy on the user's personal and confidential data without users knowledge. Users while carrying and using their device in different places unknowingly they may expose their personal data to unauthorized access as they are not knowing about malicious activities happening in their device. To control or to prevent all such types of threats they need to have better control of the device and the applications running and also an idea about how android will give permissions to applications. Users must restrict their devices from the spying of data while in meeting rooms and in the confidential places like banking sectors and in the research centers.

To achieve this, android system must control application permission system dynamically according to the context data from the users device, But such feature is missing in all the android mobile devices, It is very difficult to provide this type control on device users.

The need of this policies which are running on android device based on context data is required for all type of Smartphone users. For example, banking people who are working with critical financial issues are not allowed to bring their Smartphone which is camera enabled and the mobiles which are having internet and wi-fi access. Users may need their devices with them all the time but it is not possible, may be by using the Context based control system, employees may carry their android mobile devices into workspace. With this system users can disable and enable all applications from using any of the system resources like camera, internet, call recording and any other mobile resources according to time and location.

Once user change his location, device can get back all its permissions and privileges. Context based system is very useful for the persons who are working in military operations, and government officials, they have to disable micro phone, camera and location detecting services from their smart phones during confidential meetings and devices need to retain all its permissions according to locations.

With this policies, users can add their own policies where and when the applications can access device resources, with this chances of hacking also reduce. Previous work on security of mobile android Operating System focuses on restricting mobile applications from accessing mobile resources and sensitive data, But it lacks in good techniques which can perform action according to the context data and different locations which are closely located subareas.

2. System Model

2.1 Modules

Here is the overview of access optimized framework, It is having different components and roles to be performed and to achieve optimized services on application. It deals with accessing data, processing, collection of data and gathering the context information and applying device strategies according to the context information. To achieve this, it all contains few components those are context data, Strategy manager, Strategy executor and Access Optimizer.

2.1.1 Context Data

Context Data is a component whose main theme is to collect context data which is related to the physical parameters of the device, location and time with the help of device sensors. These details can be gathered by using GPS, Cellular IDs and wi-fi parameters. As soon as it gets this context data it links with user defined logical locations. Apart from this it notifies and updates when ever device moves from one location to another.

2.1.2 Access Optimizer

Access Optimizer manages and optimizes the unauthorized usage of device resources from the people or applications. Android has its own permission system which is stored under AndroidManifest.xml, Access optimizer will work with already existing android permission system to provide better services, with merging this to devices will be having more controlled methods and control capabilities on applications. Access optimizer provides few more services which android is not able to, those are once android issues any permissions to the applications they cannot be revoked and controlled by it, Whereas by using access optimizing technique it can be controlled. For example - RECORD_AUDIO state gives permissions to privileged application to record the users phone calls and audio clippings as well online calls like Skype, Gtalk and other sources. information about our device that is IMEI number, SIM serial number, Phone number and sends back all these details to application.

2.1.3 Strategy Manager

Strategy Manager is the place where users will create strategies to run on the device. By using the predefined context based strategies also users can create their own strategies. This will give whole control to user with the context information for creating strategies.

By using this users can even make time and location based restrictions to be applied on the mobile devices. For example - If user wants to restrict their device from using any service on every Saturday from 5 to 8'O clock, then user can create strategy according to the schedule.

2.1.4 Strategy Executor

Strategy Executor intensifies the mobile device restrictions by using its fine grained strategies. Strategy executor gives permissions to access optimizer regarding the restrictions and permissions of the system resources to particular application. Whenever user requests some permissions from the system the strategy executor will issue access t users by comparing the context data with the strategies already configured in device. Strategy executor is responsible for all the conflicts and restrictions applied through access optimizer on the applications. Whenever an application request for a service or resource first it goes to access optimizer[8], it checks that request weather it is authorized or not, meanwhile strategy executor checks for corresponding strategy in the strategy manager. If it present then it'll ask context data for the present parameters, if that also matches with it then strategy executor sends a report back to access optimizer stating that to apply all the restrictions in the strategy it provided to access optimizer.

3. Strategy Core Model

This section describes the strategy constructs that compose out strategy models. Here are the constructs that categorized according to the type of changes applied on the android OS to restrict applications and system resources.

3.1 Strategy model

The strategies have to be defined according to the restrictions that needs to be performed on the device and its applications. Strategy contexts shows that when and where the strategies should be applied and on what privileges i.e., of system methods, user data, system functionalities or in accessing device resource it should be applied. It has three set of subjects. they are

- Set of device application and services represents $D_{ser.}$
- Protected objects available for the application or system represents D_{obj}
- Restrictions that can be applied on the application represents $D_{act.}$

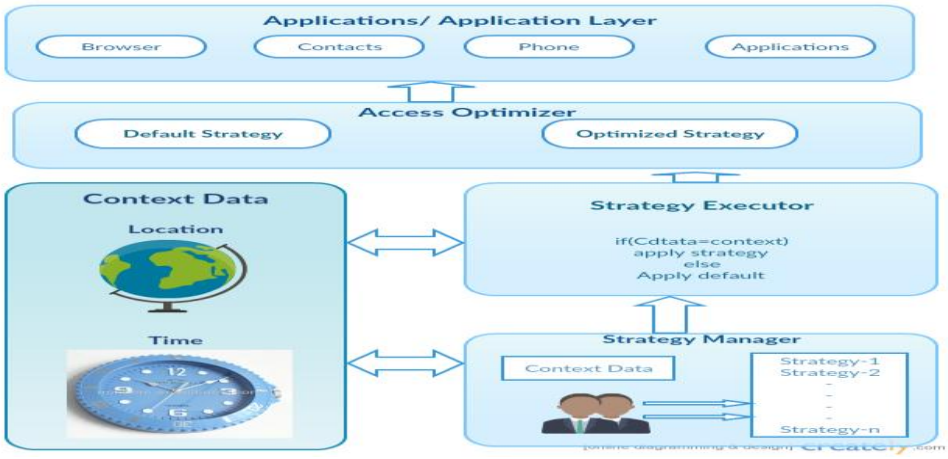

Figure 1: Architecture of the Context Based access optimizer system.

D_{app} is composed of all the application package names installed on the device, In addition to this * character is added to show that all apps intensify for same strategy instead of defining different ones to each application. Device actions will be of different types that is,

revoke_permission - Denys all the granted permissions.

shadow_Data - Hides the users sensitive data from device to unauthorized access.

Disable_Intent - Drops all intent messages to system.

Save_State - Disables from reading the system state(ON/OFF).

Statement-1

Consider $D_s \in$ Device application services($D_{Ser)}$, Do $\in D_{Obj\ and}$ $D_a \in$ Dact. With this strategy restrictions can be defined as one tuple, that is [D_s , D_o, D_a]

here D_a =

- shadow_Data if type(D_o) = Data
- revoke_permission if type(D_o) = Permission
- disable_Intent if type(D_o) = Intent
- save_State if type(D_o) = System peripheral

Access Optimizer strategies are linked to a context which specifies to be intensified. All strategies here are location based for that I am using geometric model that describes the locations on earth. To do this referred in spatial model compliant with OGC which uses GIS notion and its features. Apart from the physical locations which is identified by the device, users can even mention their logical locations which as living rooms and work places. The main use of this is users can even reuse these locations for defining new strategies without capturing the physical location of the device.

Time intervels will be provided to the strategies to be applied on the device. Specific time and date should be provided to the device to follow strategic restrictions. Date and time formats are mentioned as YYYY-MM-DD-hh-mm-ss. Apart from this one additional flag to be used that is called as recursive which is defined by a character R. The time intervel can be mentioned as [Once, Daily, Weekly, Monthly, Yearly]. The actions to be performed based on the recursive events mentioned by the user.

If user gives R value as W, that means the event has to be performed every week and to be performed weekly basis. Time also to be mentioned like 2016-04-16-17:00:00 as starting time and 2016-04-16-20:00:00 as ending time.

Statement -2

Consider L_{Loc} is the logical location which represents some sub area, and [ST, ET, R] are the tuples which represent start time, end time and repetation or recursions to be occurred. This can be considered as

L_{Loc} = [ST, ET, R]

Statement - 3

Strategy is defined by combining the restriction and context data.

Strategy = { $[D_{S,}$ DO, DA] , [ST, ET, R]}

[r, c]

where r -restrictions

c -context.

Example of disabling the camera option for imo and all other applications at business meeting happened on room number 10, every month 1st noon at10'O clock.

Strategy={[com.imo.raider, android. permission. CAMERA, revoke_Permission], [Lab10, [2016-04-01-10:00:00, 2016-04-01-12:00:00, M]]}.

Table 1: Strategy Categories with Examples.

Strategy Category	Example	Strategy Restriction
Resource restriction strategies	Disable GPS or Camera	[com.imo.raider, android.permission.GPS, revoke_permission]
Data Access strategies	Shadow contacts for imo	[com.imo.CONTACTS, android.permission.CONTACTS, shadow_Data
User security Strategies	Disable installing applications	[*, android.intent.action.INSERT, disable_Intent
System peripheral state strategies	Disable Bluetooth toggling	[*, android.BLUETOOTH, save_state]

4. Results and Discussion

Results are analyzed after running the project in android emulator and after that creating the project into .apk file which is of application type then installed into android device and then check all the functionalities of the project. To achieve this different logging commands are performed on the android operating system few of the mechanisms observed are mentioned below.

4.1 Performance Overhead and Memory Overhead

Performance overhead is nothing but checking the time delays of the system when this is integrated and used with the android operating system. As this system is modifying all the methods in the android it needs time to do this, But compared with actual android system performance level it takes little overhead in the case of time delays. For example to check the component permission of the device it is taking around 10 milli seconds longer than the actual system.

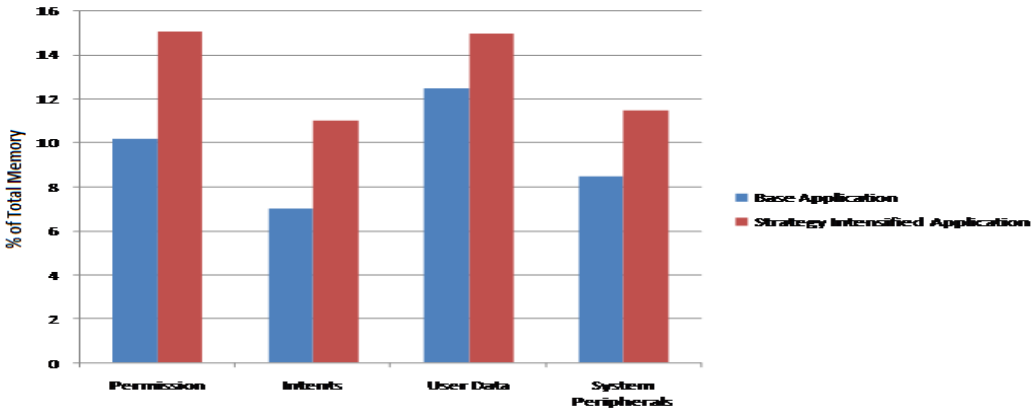

Figure 2: Memory overhead comparison in normal and in the designed system.

In the case of memory overhead, the system needs to capture and store all the strategies and their data sets which is used to set data policies on the user data as well as this system needs to store location details and to compare all those details. So it takes little more space than the base application memory. The above figure 4.1 shows the graphical representation of the memory overhead in both the cases.

4.2 Android system Impact on permission restrictions

By this check it is confirmed that the permission restrictions applied on the applications and the device is not showing any impact on the android device. The impact on the system when it is running is almost same as when the device is running default. This results shown that the system developed for restricting permissions will not cause any system crashes or performance degrades to the actual system.

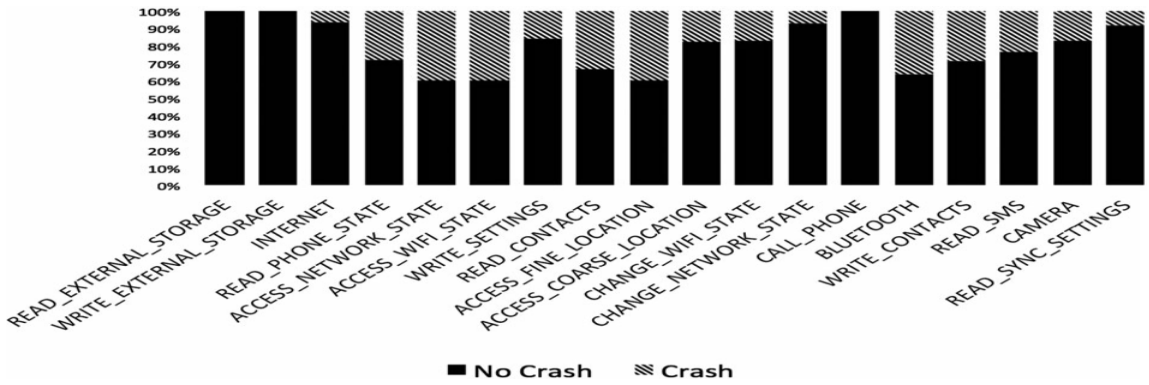

Figure 3: Shows the level of impact while permission restrictions on applications.

4.3 Battery Consumption

Through this test few analysis are done in the case of battery consumption in both the cases when our strategies are applied on the system it needs to check for the location updates or strategy updates in every time interval for that it'll take little more battery but after that it'll close all the background running process which are not required by the users in this case it'll saves some battery power. The consumption of the battery power is varies with the time interval mentioned to check for updates in the most cases time interval will be 30seconds or 1 minutes. In both the cases the battery consumption is better than the normal system.

5. Conclusion

The Modified version of the android Operating System should support Context Based Access optimizing system strategies, the strategies support users to restrict the unauthorized usage, all this restrictions are applied automatically when the devices context matches with the already mentioned strategies. The previous sections shown that the accuracy of this system in detecting location and while applying the policies. The setting up of strategies at user side needs some knowledge on the working functionalities of the application mentioned at the time of installing device. The extension of this work can be done for networks which blocks unauthorized usage of the data or entry into the network and also it restricts users privileges within that network. This will be very good and useful approach for the corporate networks to control the user privileges inside their network.

6. References

1. Wi kipedia, Samsung galaxy s4 specifications. Available: http://en.wikipedia.org/wiki/Samsung_Galaxy_S4.
2. A. Kushwaha and V. Kushwaha, “Location based services using android mobile operating system”, Int. J. Adv. Eng. Technol., Vol. 1, no. 1, pp. 14–20, 2013.
3. Bilal Shebaro, Oyindamola Oluwatimi, Daniele Midi, Elisa Bertino, " IdentiDroid: Android can finallyWear its Anonymous Suit" , In proceedings of Computer Science, Cyber Center and CERIAS, pp. 328–332, 2012.
4. S. Bugiel, L. Davi, A. Dmitrienko, S. Heuser, A.-R. Sadeghi, and B. Shastry, “Practical and lightweight domain isolation on android,” in Proc. 1st ACM Workshop Security Privacy Smartphones Mobile Dev, pp. 51–62, 2011
5. Gil Heo, Genong Yu, Liping Di, "A Reconfigurable Open GeoSMS Mobile Client App Design for Android Smartphones", In proceedings with Center for Spatial Information Science and Systems (CSISS) George Mason University, pp.962-759, 2013.

6. Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee and Zhenhui Tan, Member, IEEE, "Received Signal Strength based Indoor Positioning using Compressive Sensing", In proceeding to IEEE TRANSACTIONS ON MOBILE COMPUTING, pp.952-959, 2013.

7. F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowan, "User-driven access control: Rethinking permission granting in modern operating systems," in Proc. IEEE Symp. Security Privacy, pp. 224–238, 2014.

Corresponding Author

E.Koteswara Rao*,

Email: *koti0514@gmail.com*