International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 121-123
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

CONTEXT BASED ACCESS CONTROL SYSTEM IN ANDROID SMARTHONES

Rahul P, Nihal P Sunil, Liviya Vincent, Joby George
Computer Science and Engineering Department,
Mar Athanasius College of Engineering,
Kothamangalam, Kerala, India

***Abstract*-- Android users do not have control over the application capabilities once the applications have been granted the requested privileges upon installation. In many cases, however, whether an application may get a privilege depends on the specific user context and thus we need a context-based access control mechanism by which privileges can be dynamically granted or revoked to applications based on the specific context of the user. Here such a system is implemented in android smartphones.**

Keywords— **Context-based access control, smartphone devices, security and privacy, policies, mobile applications**

I. INTRODUCTION

Modern smartphones have powerful computational and communication capabilities and the applications make use of these facilities to enhance their services. For example Samsung s7 comes with a powerful octacore processor and enormous 6GB RAM with lots of sensors the applications that can make use of these high tech facilities to enhance their service like heart beat sensor in Samsung S7.But these can also enable the applications to extract users data and this may lead to data exposure and serious privacy breach. With the user completely unaware of the existence of these malicious applications the user's privacy may be at risk. This is due to lack of complete control over the applications resource access. Since most of the android operating system except Android Marshmallow doesn't provide a mechanism to revoke the permissions of the android applications once they are installed there is a need of a control system that gives complete control over the device resources and manage the applications that uses those resources.

II. ANDROID

A. Operating system and API –

The Android operating systems are derived from Linux Based kernels and have enhanced support for security and Privacy. Android is designed with a multi-layered Security infrastructure, which provides developers a secure Architecture to design their applications. Android applications are sandboxed and run inside the Dalvik Virtual Machine, which isolates each application's processes and data. Each application is assigned a user ID (UID) with assigned privileges, and is given a dedicated part of the file system for its own data. The processes of applications with different UIDs run separately and are isolated from each other and from the system. While applications have limited access to device Resources, developers can explicitly request access for their applications through the Android permission system.

B. Permission System –

The Android permission system controls which application has the privilege of accessing certain device resources and data. Application developers that need access to protected Android APIs need to specify the permissions they need in the AndroidManifest.xml file which, if inaccurately assigned, can increase the risks of exposing the users' data and increase the impact of a bug or vulnerability. Each application declares the permissions listed in its AndroidManifest.xml file at the time of installation, and users have to either grant all the requested permissions to proceed with the installation, or cancel the installation. The Android permission system does not allow users to grant or deny only some of the requested permissions, which limits the user's control of application's accessibility..

III. LOCATION POSITIONING METHODS

A. GPS Trilateration

The Global Position System (GPS) is a positioning tool available in most smartphones which uses data signals from satellites to compute the position of the device. Data received from satellites contains the time stamp of sending, the orbital information, and the position of satellites. With at least three different satellite signals, the GPS uses the trilateration method to calculate the device's location by measuring the satellite signals time difference or their received signal strengths. The location information provided from the GPS includes the latitude, longitude, altitude, and time. The

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 121-123
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

accuracy of this method is estimated to be in the range of 50 to 100 meters.

B. Wi-Fi Positioning Method

Many smartphone devices nowadays rely on Wi-Fi-based positioning techniques due to its efficiency in computing the location of a device especially in places when GPS and cellular signals are weak or unavailable. These techniques are based on comparing the device's received Wi-Fi access points (AP) with database fingerprints containing Wi-Fi access points with known geographic locations. In our work, smartphone users will define their own database of Wi-Fi APs which only includes the user's areas of interest. In addition to the Wi-Fi APs, we also store the Received Signal Strength Indication (RSSI) of each AP to enhance the positing accuracy of the device, which in our case, needed to differentiate between nearby sub-areas. We describe in details how we capture and store these Wi-Fi parameters in Section 6, in addition to how we use them for detecting the device location.

C. Context Management

The main source of location-related information for our access control system is the Wi-Fi APs and their corresponding signal strengths. Location information acquired from GPS and cellular towers is also aggregated to our context definition but may not be sufficient for indoor localization especially that they may become weak or unavailable inside buildings or areas within building structures. However, location information retrieved from Wi-Fi parameters could be more precise to differentiate between closely located sub-areas within the same GPS location.

A spatial region is represented by combining GPS coordinates, cellular triangulation location data, and Wi-Fi APs and signal strengths. In Android, the GPS coordinates and cellular triangulation are obtained in a similar fashion by invoking the Android LocationManager service. Once the LocationManager is invoked, we request location updates by calling the requestLocationUpdates method that returns a Location object which contains latitude, longitude, timestamp, and other information. Wi-Fi is handled differently than the previous two location methods. We obtain the Wi-Fi parameters by invoking the WifiManager service to retrieve the Wi-Fi access points scans. We register our BroadcastReceiver mWifi receiver with an IntentFiler action to receive the broadcasted Wi-Fi scanned intent, and then request for and subsequently process the actual scanned access points data.

In our CBAC policy system, we provide users with a utility to define physical locations by either capturing snapshots of location data of the desired areas or by manually entering the area location coordinates. In the following sections, we show our design and implementation of the location capturing phase when users define and store physical locations, and the location detection phase when device detects its location and match it with a pre-defined policy context.

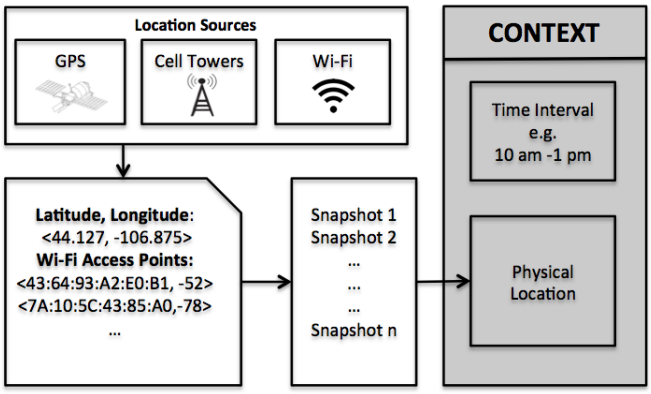

Fig. 1: Location Capturing Phase.

D. Location Capturing Phase

FIGURE 1 describes how location data is captured for each context defined by the user. Through the location scan interface, the user is able to capture several snapshots of location data in different sub-areas. For each sub-area, location data is accumulated from each snapshot; the GPS coordinates and the cellular triangulation, when applicable, import the latitude and longitude from the captured snapshots and only select those with the highest position accuracy. With respect to Wi-Fi, we noticed that the Wi-Fi access points signal strengths fluctuate even if the device is stationary or motionless. Therefore, our application scans the signal strengths of each access point for several seconds gathering the RSSI values at each particular sub-area. Finally, the accumulated data, which mainly consists of Wi-Fi access points with signal strength ranges in addition to GPS and cellular triangulation data as supporting location information, will represent one physical defined location to the user.

Any location that is not defined by the user or does not have location information saved on the device will be considered "Unregistered". Therefore, we designate default policy restrictions for the user to configure whenever the device is located in an unregistered location. In addition, we allow users to register locations that have not been previously visited. This is achieved through either manually entering the publicly known longitude and latitude of the desired location, or by acquiring the fine-grained Wi-Fi parameters from other devices who have saved those parameters. This becomes very practical when the user is switching between two devices and

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 121-123
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

needs to import previously saved policy contexts to the new device.

E. Location Detection Phase

Figure 2describes how device context is detected and matched with pre-defined context. Periodically, the location background service is re-instantiated to accumulate location-context data to determine the device's current whereabouts. Like when registering and scanning a sub-area in the location capturing phase, we scan the device's location-related data. The list of user-registered areas that have a subset of the scanned neighboring access points are extracted from the database first. Matching distinct access points is computationally less expensive than determining if coordinate position falls within the boundaries of a convex hull. Then

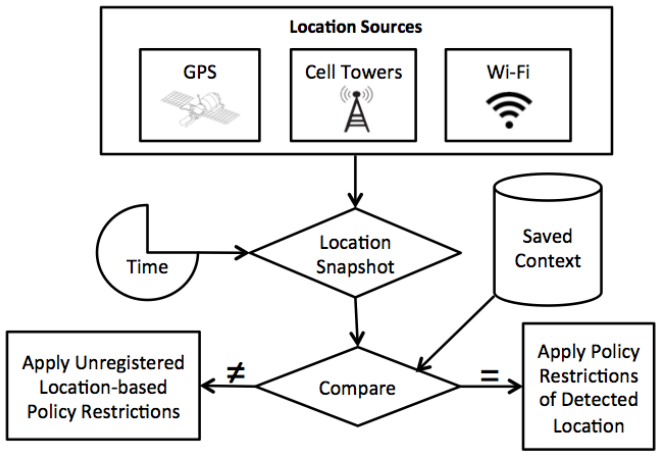

Fig. 2: Location Detection Phase.

, using the current signal strengths of the access points, we reduce the list to only a set of "best-match" list of physical locations whose access points fall within the current captured signal strength values. If the current scanned GPS or cell network coordinates fall within the convex hull of the associated sub-area, then it is highly likely that the sub-area has been located.

IV. CONCLUSION

An android application version of the android operating system supporting context based access control policies are used to avoid the misuse of data by malicious applications. The policies restrict applications from accessing specific and resources based on the user content. The restrictions specified in a policy and automatically applied as soon as the user device matches the predefined context associated with the policy. The implementation of CBAC mechanisms results a good effectiveness of the policies on the android system and the accuracy in locating the device within a user defined context. The user requires configuring their own set of policies. The difficulty of setting up configurations requires the same expertise needed to inspect application permissions listed on the installation time.
The CBAC android application can be further modified to record attendance and record individual works of employees. Industry and games in it are expected to be the next revolution in the industry

V. REFERENCE

[1] R. Templeman, Z. Rahman, D. J. Crandall, and A. Kapadia, "Placeraider: Virtual theft in physical spaces with Smartphones," CoRR, vol. abs/1209.5982, 2012.

[2] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones," in Proceedings of the hu18th Annual Network & Distributed System Security Symposium (NDSS), Feb. 2011.

[3] L. L. N. Laboratory, "Controlled items that are prohibited on llnl proporty," https://www.llnl.gov/about/controlleditems.html

[4] M. Conti, V. T. N. Nguyen, and B. Crispo, "Crepe: context-related policy enforcement for android," in Proceedings of the 13th international conference on Information security, ser. ISC'10.Berlin, Heidelberg: Springer-Verlag, 2011, pp. 331–345.

[5] 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.