Context-Aware Access Control Mechanism on Android Smartphone

Thiri The` Wut Yee

University of Computer Studies, Yangon
thirithewutyee@gmail.com

Abstract

The proliferating numbers of smartphone applications which access device's functionalities and resources such as GPS, SMS etc imposes security challenges. In the meantime, the existing security and access control mechanism of smartphone mostly hold a coarse-grained and incomplete access control model. This paper proposes a fine-grained access control; a context-related access control mechanism for Android smartphone platform. Exploiting the current behavior of system and user as well as their interaction, the proposed system enables access control to be made not only at install-time but also at runtime. Moreover, users can impose security policies to their device usage without requiring technical knowledge. By using simple policy and context, the system fulfills necessity of existing access control mechanism within minimum performance overhead.

Keywords: access control mechanism, Android, context

1. Introduction

Smartphone platform is growing in importance and providing increased yet complex capabilities. Moreover, applications for mobile phones are proliferating that access device's functionalities and resources such as GPS, SMS messaging, WiFi etc. Consequently, several challenges are introduced especially in security issues. The most important feature is to restrict the behavior of users using applications and services. In the mean time, the existing security and privacy control mechanism mostly hold a coarse-grained and incomplete security model. Most of current systems work on device level security which is per application basis, particularly at installation. User installing third party applications has to trust that the application will not misuse device's resources. Similarly, if user wants to use that application, he must have to grant all permission requests. This all or nothing decision leads to coarse-grained access control mechanism. In addition, as soon as user grants the permissions, there is no way to restrict or revoke these permissions based on user current activities except from uninstalling that application. For example, a user might want to restrict amount of SMS sent for a day to save charge fees by using contextual information such as access patterns. To address these issues, this paper proposes a fine-grained access control framework for smartphone; context related role based access control mechanism for Android smartphone platform. Contributions are as follows Based on the combination of contextual information with role based access control mechanism, existing coarse-grained access control mechanism is enhanced to a finer mechanism. In addition, the paper presents a simple policy model which user to specify run-time policies on smartphone's resources and services.

Roadmap: Section 2 describes background theory context-related role-based access control system. In section 3, problem description with motivating scenarios is presented. Section 4 lists existing works and section 5 describes proposed system model and section 6 is the overview of proposed system framework. Finally, section 7 gives some conclusion remarks.

2. Background

Role based access control (RBAC) is the process of mediating every request to resources

and services maintained by a system and determining whether the request should be granted or denied based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles. While the concept RBAC is not new, challenges are facing when it comes to merge with mobile platforms and researches are still in progress to exploit this technology with full potential [7].

Fundamentally, RBAC require the identification of roles in the system, where a role can be defined as a set of actions or responsibilities associated with a particular activity. Users are then given authorizations to adopt roles. The user playing a role is allowed to execute all authorized accesses. In general, a user can take on different roles on different occasions. However, in smartphone system, role per user is limited to one and phone owner has the highest priority role. In combination with roles and user responsibilities, system wide security policies are constructed.

A context is described as a property of environment and system at the moment that user and system interact. Particularly, in mobile computing environment, a context can be the status of some variables (e.g. location, time, temperature, noise, and light), the presence of other devices, a particular interaction between the user and the smartphone i.e. accessing resources depending on an application's usage and different modes of corporation between different, or a combination of these . Lately, this idea is mixed with access control mechanism for mobile devices which offers new opportunities for developers and end users by gathering context data and adapting system behavior according to the security policies. Moreover, this technique is highly suited for smartphone security for 1) smartphone is centralized, user-centric system where user population is static and identities of users are known in advance 2) smartphone environment is rich of contextual information 3) it provides ease of security administration since access permissions are not assigned directly to the users but to abstractions known as roles.

3. Problem Description

Among current mobile computing trend, Android is the most popular open source operating system for smartphones especially for smartphones and is developed by Google and Open Handset Alliance. The purpose is to create an open platform for handsets and make mobile applications interoperable crossing venders. It is a software stack for mobile devices including an operating system, middleware in the form of a virtual machine, system utilities as well as core applications such as web browser, dialer, calculator and significant Google applications such as Google map. The Android SDK provides the tools and APIs necessary to built up applications on Android platform.

Android security mechanism uses mandatory access control mechanism and sandbox technique which controls access to resources by process ownership [2]. That is it forbids one process from accessing another process's memory and files that are created by a particular application. An application which has a specific user ID can't be read or written by other applications. When installing new application, Android requires application to request specific permissions for resources. To do so, the user has to trust that the application will not misuse phone's resources. Once the permissions are granted and the application is installed, the user cannot change these permissions except by uninstalling the application from devices. Thus Android provides coarse-grained security level i.e. neither to enforce or to change security policies at application run-time. In order to restrict access dynamically, it is practical to leverage context-related information. Context aware access control mechanism is a mobile computing paradigm in which applications can discover and take advantage of contextual information such as user location, time zone, nearby people and devices and user activity etc and is exploited in decision making of access control. Context-aware system offers new opportunities for developers and end users by gathering context data and adapting system behavior according to the security policies. In combination with mobile devices, this mechanism is of high value and is used to

increase usability. Moreover, context-related access control model enhances Android's security mechanism to fine-grained manner.

3.1. Motivating Scenario

In the open source scenario, Android encounters not only security but also privacy issues. Third party developers can submit their applications to Android Market and from which users can download and install. Whereas this provides the greater accessibility of variety of applications, it gives rise to serious problems. In order to demonstrate the existing Android security framework and its limitation, some application samples are outlined as follow.

Access control principle: Suppose that parents want to control their children's phone usage. They do not want their children to spend too much time playing online games or accessing Internet. Thus they need means to control their children's smartphone access and determine what application can run and how long they can run etc.

Privacy principle: A user might want to restrict the usage of phone resources and services according to the user role such as friend-using or admin.

Security principle: Consider as company's phones are handed out to employees for business use on which critical applications are installed to facilitate corporate operations and workflows. In the mean time, company can prevent from disclosure of company's secret data via phones by limiting the number and capacity of SMS messages sent by employees each day.

4. Related Work

As the above issues are nontrivial, it has gained intentions from researchers. The research community has been investigated secure solutions for smartphones. Yet, less convincing results have been obtained for enforcing security at application run time. This is because of the limited nature of smartphone in terms of battery and memory overhead.

Saint [5] proposed enhance security mechanism of Android by improving install and run-time policies. Communication between applications or components is subjected to security policies asserted by both the caller and callee applications. In such a away, device security is controlled by application provider's policy and not by the user's policy. It's onkly suitable for developer. Moreover, it can pose only install-time policies.

Another work concerned about fine-grained access control is SCanDroid which is a tool for automated security certification of Android applications. It statically analyzes data flows through Android applications, and makes security-relevant decision automatically. It is a reasonable model for offline certification [6]. It only tries to improve the existing Android security mechanism to be best practice. However, it mainly depends on source code inspection and Android manifest file; thus it is not applicable for average user.

In addition, the increased number of GNU GPL license applications results in a greater chance of installing Trojans and similar malware. W. Enck et. al. propose Kirin [2] security service for Android, which performs lightweight certification of applications to mitigate malware at install time.

In Paranoid Android [6], an alternative solution is proposed where security checks are applied on remote security servers which host exact replicas of phones in virtual environments. It is a security model that performs attack detection on a remote server in the Cloud where the execution of software on the phone is mirrored in a virtual machine. Although it can deal with security of applications, it mainly depends on the Cloud.

Finally, some researches have been carried out in the area of modeling context aware system to provide meaningful and valuable context information rather than raw context data to the system. In paper [9], ContextDroid is presented and it is designed to provide application developers with the services required to easily build context aware application with an eye towards reducing overhead.

Above all, the finer access control mechanism is still needed for smartphone system. Furthermore, the simpler and the more

general policy model is welcome in such a way for easy implementation of an access control system regardless of user defined policies. The endeavor of this paper is to fulfill such requirements to some extent.

5. Proposed System Model

The development of access control system requires the definition of regulations according to which access is to be controlled and their implementation as functions executable by a computer system. The concept of access control is clearly established with three phases: access control security policy, policy model and policy mechanism [7]. In this paper, a simple access control model is proposed. It classifies smartphone user based on their privileges and sets access control decisions according to the system defined policies. The objective is to boost the existing Android resource usage management by exploiting context information explicitly.

The proposed system model composes of 5 elements S, O, C, P, D where

- S: set of all subjects according to user roles
- O: set of all objects
- C: set of all environment context
- P: set of all user-defined policies
- D: set of decisions.

Subject: Subject is an entity that holds and exercises specified rights on objects. In this paper, subjects are users which are classified according to their role such as phone owner (system admin), end-user (guest-user) and other users who are not included in the first two classes (unknown-user).

Object: Object is an entity that subjects can access or use. To be exact, objects of a smartphone system can be system setting, services and applications etc. In the proposed system, the following objects are secured against current user contexts: SMS, contacts, S/W install/uninstall, GPS, bluetooth, prioritize applications.

Context: A context is described as a property of environment and system at the moment that user and system interact. In computer applications, context is acquired explicitly by requiring from sensors or implicitly by monitoring user and system. For example, explicit context can be acquired using camera for location context, humidity sensor to measure human body temperature etc. Implicit context can be attained by monitoring conditions within the computer system such as system clock, power usage etc. The former is also known as physical or real context and the latter is virtual or logical context. There is another model aspect of context which represents context parameters absolutely or relatively. Examples of absolute contexts are time, place, usage count, extra. After inference, relative contexts are induced such as on meeting, at home, travel, on vacation. The contexts exploited in the system are virtual, relative context such that user is at the location where he works between 8AM to 4PM then the inference context is at work (from 8AM to 4PM ^ lat# ^ long # etc).

Policy: Policy is the formal representation of the access control and its working. Proposed model represents policies as <Subject, Context, Object, Decision> quadruples. Formally, a policy $p \in P$ is a set $(S,O,C,D) \in (S \times O \times C \times D)$ where D={Allow, Deny}.

6. Framework Overview

To address issues related with access control on Android smartphone, the following framework is proposed. The framework consists of three components: context-aware access control (CAAC) engine which controls usage decisions, policy enforcement (PE) component and context inference (CI) component which customized captured context information to be suited for decision making process. The framework overview is illustrated in figure 1.

Firstly, phone owner (system admin) sets access policies and these policies are stored in policy database. Thereafter, as soon as phone is switched on, CAAC is instantiated. It prompts the user a home screen with two options (Admin home and End-user home). If user chooses end-user mode, system will launch home screen according to the normal user privileges. When

user selects admin mode, correspondence screen will be displayed via user authentication. From user point of view, end-user home screen will not differ from that of admin. However, only admin privilege can impose access control policy of phone resources for both user types.

CAAC: The main role of CAAC is to enforce user policies on the usage of system resources. It monitors user activities, leverages them with current context, matches against with policies and then makes access control decisions.

PE: Policies are stored as <subject, object, context, decision> quadruples. By mean of objects, all system resources are included such as contacts, notes, Bluetooth, camera. Subjects are users who are expressed in terms of their roles. Contexts here refer to relation rather than absolute contexts. These policy are stored according to policy model.

CI: For the sake of simplicity, our system exploits context captured from virtual context sensors. This component only reacts as soon as any user activity occurs. The captured absolute contexts are deduced to relative context.

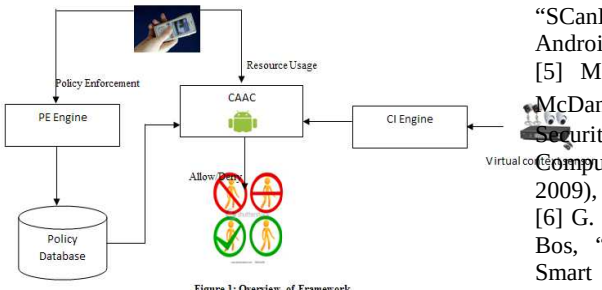

Figure 1: Overview of Framework

7. Conclusion

Android security mechanism is device level security, works on per application basis, typically at install. Obviously it is coarse-grained mechanism. To protect confidential content and the integrity of services, there should be a framework which dynamically allows and restricts access to resources and services. While this mechanism is achieved by user-centric, context-related security mechanism, the effort is still medium in research area since the security policies are hard to define and learn. Through ongoing study, a valuable yet feasible framework which exploits user context information to provide fine-grained security control mechanism is proposed. In addition, by using simple policy and context model with the aid of clear subject/object mapping mechanism, it can be concluded that the system will fulfill the security needs within minimum performance overhead.

References

[1] Jesse Burns “Developing Secure Mobile Applications for Android” , 2008.

[2] W. Enck, M. Ongtang, P. McDaniel, “On Lightweight Mobile Phone Application Certification”, Proceedings of the 16th ACM Conference on Computer and Communications Security, November 2009.

[3] W. Enck, M. Ongtang, P. McDaniel, “Understanding Android Security”, IEEE Security & Privacy Magazine, Vol. 7(1), January/February 2009, pages 10-17.

[4] Adam P. Fuchs, Avik Chaudhuri, Jeffery S. Foster “SCanDroid: Automated Security Certification of Android Application”, 2009.

[5] M. Ongtang, S. McLaughlin, W. Enck , P. McDaniel, ”Semantically Rich Application-Centric Security in Android”, Proceedings of Annual Computer Security Applications Conference (ACSAC 2009), December 2009.

[6] G. Portokalidis, P. Homburg, K. Anagnostakis, H. Bos, “Paranoid Android: Zero-Day Protection for Smart Phones Using the Cloud”, Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10), 2010, pages 347-356.

[7] P. Samarati, S. D. di Vimercati, “Access Control: Policies, Models, and Mechanisms”, Foundations of Security Analysis and Design, Tutorial Lectures, vol 2171, 2001, pages 137-196.

[8] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev “Google Android: A State-of-the-art Review of Security Mechanisms”, CoRR, abs/0912.5101, 2009.

[9] B. Wissen, N. J Palmer, R. Kemp, T. Kielmann, H. Bal, “ContextDroid: An Expression-Based Context Framework for Android”, 2010.