Context-Related Security Control Mechanisms on Android Platform

Thiri The` Wut Yee
University of Computer Studies, Yangon, Myanmar
thirithewutyee@gmail.com

Abstract

The revolution of mobile phones era has brought the innovative smartphones technology. As a result, research addressing information access in smartphones environment has proliferated. The important feature in security of smartphones is to restrict the behavior of users using applications and services. In the meantime, the existing security mechanism on Android mostly holds a coarse-grained and incomplete security model. To enhance this into fine-grained manner, it is practical to use context-related information to specify precisely what resources and services an application should be authorized access to. While the concept of context-related security mechanism is not new, challenges are facing to adopt this technology to its full potential. In this paper, we attempt to review the existing fine-grained context-aware models. These approaches differ regarding of how control is initiated e.g. via SMS messages, how context is activated e.g. manually or automatically etc. From this study, we intend to build simpler as well as clear security control mechanism based on user context such as access patterns of resource or working set. We also aim to apply classification technique in the decision making process for access control without user intervention. This paper is initial endeavor of our research proposal.

1. Introduction

Mobile platforms are growing in importance and have complex requirements. The computational power of mobile devices is continuously increasing leading to smartphones [8]. Consequently, research addressing information access in smartphones environment has proliferated. Android is the most popular open source operating system for smartphones that is created by Google and Open Handset Alliance. Although Android tightly controls the behavior of the users using applications and resources, it mostly holds coarse-grained and incomplete security model.

Android uses mandatory access control mechanism and sandbox technique which controls access to files by process ownership [3]. That is if forbids one process from accessing another process's memory and files that are created by a particular 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.

A context can be defined by 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 [8]. 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-aware control model enhances Android's security mechanism to fine-grained manner.

Roadmap: In Section 2, we present background theory of context-aware system with motivating examples. In section 3, survey of existing well-known fined-grained context-related security control frameworks are listed. Finally, section 4 gives some conclusion remarks.

2. Background

Context-aware 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. The following section describes clear explanation of what context-aware mechanism is.

2.1 Motivating Examples

1. When a user loses his phones, the lost phone detects strange geographical location, peculiar dialed patterns or different time zone which exposes unfamiliar context.
2. An enterprise can prevent from the disclosure of company's secret data via phones by limiting the number and volume of SMS messages sent by employees each day.

3. A user might want to restrict the usage of phone resources and services such as contacts or Bluetooth devices by using temporal or spatial context information.
4. User can restrict the amount of SMS sent for a day to save charge fees by using contextual information.

3. Fined-Grained Security Control Mechanisms

In this section, we survey previous research work of context awareness, focusing on applications, how contextual information is gathered, how the control is activated etc.

3.1 Basic Design Principles

Context-aware systems can be implemented in many ways. The approach depends on special requirements and conditions such as the location of sensors (local or remote), the amount of possible users (one user or many), the available resources of the used devices (high-end-PCs or small mobile devices) or the facility of a further extension of the system. Furthermore, the method of context-data acquisition is very important when designing context-aware systems because it predefines the architectural style of the system at least to some extent.

In addition, user context information can be extracted from a variety of sources (sensors). One is physical sensors. Hardware sensors can be used to capture physical location e.g. phone camera. Another source is virtual resource. Context data are extracted from device's applications or services. For instance, location can be determined by browsing electronic calendar or email etc. There are also hybrid forms which collect information from both of the above sources to solve more complex tasks.

3.2 Survey

Apex: Android Permission Extension is a user-centric framework which allows users to specify detailed runtime constraints over the usage of sensitive resources by applications. This is achieved through an interface of extended Android installer; Poly. It also allows finer-grained control over usage through attribute updates but it still needs in bulk decisions of which constraint types are the most usuful [10].

CRePE is a fine-grained Context-Related user's Policy Enforcement mechanism for smartphones. CRePE enforces runtime policies as well as allows third party's application to be trusted. It is also a user-centric which user can enforce policies through SMS messages. Contexts are automatically detected and activated by CRePE unlikely from the existing context concept of smartphones e.g. outdoor or silent mode. CRePE places policy enforcement mechanism before the Android permission checked. With slightly updates, it enforces fine-grained context-aware security framework of smartphones. However, it suffers overhead of time for the location checking [8].

B. Guangdong, G. Liang also proposed a usage control framework which supports revocation and changes of an application's permission dynamically at runtime. It enhances security and resource usage on mobile phones platform especially for Android smartphones [4].

SCanDroid 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 [3].

ConUCON (Context-aware Usage CONtrol) model supports flexible data protection and resource usage constraints. It provides continuous usage control because its usage decisions are not only performed prior to the access but also during the access. Because of the usage decision, it also performs extra action which leads to some overhead [5].

Although the security model of Android system is system-centric, **Saint** (Secure Application INTeraction) offers application-centric model. It addresses install-time permission granting policies and run-time inter-application communication policies. It exploits operational policies to disclose the impact of security policy on application functionality [11].

4. 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, we hope to develop a valuable yet feasible framework which exploits user context information to provide fine-grained security control mechanism.

References

[1] A. Craig, N. Soules, Gregory R. Ganger "Connection: Using Context to Enhance File Search".

[2] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev "Google Android: A State-of-the-art Review of Security Mechanisms".

[3] Adam P. Fuchs, Avik Chaudhuri, Jeffery S. Foster "SCanDroid: Automated Security Certification of Android Application".

[4] G. Bai, L. Gu, J. Kong, Y. Guo, X. Chen "Context-aware Usage Control Mechanism for Securing Android Platform".

[5] G. Bai, L. Gu, J. Kong, Y. Guo, X. Chen "Context-Aware Usage Control for Android".

[6] H. Chen, T. Finin, A. Joshi "An ontology for context-aware pervasive computing environments", Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, Vol. 18, No. 3, pp.197–207.

[7] Jesse Burns "Developing Secure Mobile Applications for Android" .

[8] M. Conti, V. T. N. Nguyen, B. Crispo "CRePE: Context-Related Policy Enforcement for Android".

[9] M. Nauman, S. Khan, "Design and Implementation of a Fine-grained Resource Usage Model for the Android Platform".

[10] M. Nauman, S. Khan, X. Zhang "Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints".

[11] M. Ongtang, S. McLaughlin, W. Enck , P. McDaniel "Semantically Rich Application-Centric Security in Android".

[12] W. Enck, M. Ongtang, P. McDaniel "Understanding Android Security".