Prevention of Data Overcollection In Smart Devices

V R S Sam Anand[1], Dr E Srie Vidhya Janani [2]
[1]PG Scholar, [2]Assistant Professor,
Department of Computer Science and Engineering,
Anna University Regional Campus,
Madurai, India.

Abstract— Mobile Security emphasizes on data overcollection which amounts to data being collected by any application or by any process, other than the required functionalities which comes under permission of the users. It seems to be unavoidable since the devices leak privacy data with user's permission but without their knowledge. Such applications run as a service in background process without the user's knowledge. Every device has the user's privacy data such as their personal information and certain sensitive information like the bank account details and so on. This data over-collection becomes an important severe potential hazard for privacy in devices. This privacy issue is mainly focused in the all smart devices which uses smart operating systems like Android, IOS, Windows phone Operating System etc. Such problems are to be treated as malwares. But unfortunately, the data over-collection is complex to solve unlike the malware because of its undue legitimacy with user's permission. This highly necessitates displaying to the user what data has been collected by the apps thereby allowing the user to impose control on permission to the apps with relevance to privileges found in the user's device. The data collection issues in Mobile environment has been resolved with the data overcollection prevention algorithm using mobile-cloud framework proposed. By using mobile-cloud framework, the users can store the information in the cloud environment where a huge number of users are involved. Mobile-Cloud framework is created by using Android Studio for developing an application which runs in android platform, collects the information from the apps installed in the device and verify the authentication. The cloud environment can be created by using any open source tools. The results are calculated by the performance of the framework.

Index Terms—Cloud, Data overcollection, Mobile Security

1 INTRODUCTION

Smartphones are widely used electronic devices now-a-days, because of its portability. The usage of mobile users is enormously high due to its portability, usage and handy. The users store all their information which includes the personal and the privacy. Hence security is seeming to be a threatening factor. There are various ways which abuse the user's security. Using smartphones, the user can access to the Internet via everywhere Wi-Fi, take online courses, pay their bill online, sign a contract online, and receive medical treatment by tele-health. The smartphone not only stores user's data, but also generates data. These data may consist of user's accounting numbers and passwords, emails and house addresses, photos, and other kinds of sensitive information. As a result, the security and privacy of smartphone data becomes an important issue to achieve the blueprint of smart city. With the development of electronics technology, all kinds of smart phones flood into the market, and smart phones are increasing usurping on peoples' life with go-anywhere apps offering a wide array of enterprise, social, financial, and recreational services. The streamline of marketing, installation, and update creates low barriers for developers to bring apps to market, and even lower the barriers for users to obtain and use them. Besides the popularity and functionality, however, apps bring us enormous security problems.

The security breaches include the both web-based threats and application threats. The web-based threats include the phishing, web browser exploits, add-on such as formatting issues, scripting issues, security by-passes and protocol handling and downloading executable files. The application-based threats include all the offline threats such as software tampering, authentication, authorization, configuration management, sensitive information, session management, and cryptography [1]. These threats mainly focus on retrieving the information of the users from their devices. Collecting the information or data from the user device is known as the Data collection.

The Data collection includes all types of information gathered by the device. Since the smartphones not only stores the data, but it processes and produces the result. The Data collection contains all the information such as temporary information and the privacy information. The Temporary information is used by any application which is offline or online, can be used for storing the information of the user frequently accessed. The privacy information includes all the data collected to identify the user and their activity. There is no 100% of blocking the leakage of the information in the smart devices. So, the user must be aware of the privacy information collected by the user.

The purpose of this paper to identify the current state of data over-collection and identify some most frequent data over collected cases. This mobile-cloud framework is an active approach to control the data over-collection. By putting all

- *Sam Anand V R S is currently pursuing masters degree program in Computer Science Engineering in Anna University Regional Campus Madurai, India. E-mail: vrssam.anand@mail.com*
- *Dr. E. Srie Vidhya Janani, Assistant Professor, Department of Computer Science and Engineering, Anna University Regional Campus, Madurai, India. E-mail: esrievidhya@mail.com*

user's data either in the cloud or on their device. The user's data can be greatly improved if the data is stored in the cloud, since the device's processing speed and memory are low. So, the information can be stored in the cloud for best performance [2].

The data overcollection can be avoided by verifying the application's what, how and which, data has been processed or requested by that application and whether the application is allowed to use/get/process that data from the user if the user permits to access. A service has been created to run background along the operating system so that whenever any application request to process or access a data, this service has been triggered and performs the required action.

2 Related Work

In this section, how the current solutions solve the data overcollection is discussed. The defense against the security hazards have two approaches, such as active and passive methods. The Passive method uses relevant to monitoring and detecting and the Active method uses relevantly to prophylaxis. In data overcollection trajectory solutions always uses the passive measures.

Title	Method	Operation	Disadvantages
PiOS: Detecting Privacy Leaks in iOS Applications [3]	Analysis to detect the privacy data in order to identify the aim of privacy leak.	Monitoring and Detecting methods	These solutions just provide methods to find out the behaviors of data overcollection, but leave other operations to users, such as stopping them by uninstall those apps.
TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones [4]	Real time analysis by leveraging Android's Virtualized Execution Environment		

These approaches or tools adds the consumption of energy, which is particularly valuable in smartphones with limited resources.

Title	Method	Operation	Disadvantages
Understanding Android Security [5]	Automatically extract the security manifest of Android apps.	User aware approaches	The development of technology, app developers will have increasing methods to program and increasing methods to achieve the goal of collecting users' data.
User aware privacy control via extended static information flow analysis [6]	A user-ware privacy control approach to reveal how private information is used inside applications		

The coarse-grained permission authorization may weaken the effects of this kind of approaches. After all, passive approaches cannot be competent for protect users' data.

Title	Method	Operation	Disadvantages
Making the Case For Computational Offloading in Mobile Device Clouds [7]	Environment in which computational offloading is adopted amongst mobile devices.	Offload in Mobile Cloud Computing	These researches is to achieve efficient utilization of cloud resources and to save mobile resources.
A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing [8]	A framework to provide runtime support for the dynamic computation partitioning and execution of the application.		

The mobile cloud computing technologies focus on augmenting the execution of mobile applications on smartphones using cloud resources, while in our framework, not only the execution but also the native data are in a cloud.

3 System Implementation

3.1 System Architecture

The proposed architecture is based on the following main components: Smart phone, Cloud service or storage devices. The architecture is illustrated in Figure 1

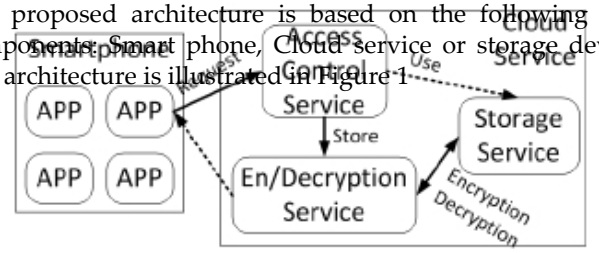

Figure 1 System architecture

It is impossible to enforce app developers not to share user's data with advertisement networks and other third party organizations, and it is unreasonable to expect that all smartphone users can understand permissions clearly and protect their privacy information carefully. In fact, the security problem is identified and to solve it, this have to change patterns of thought, not to deal with aftermath but to eradicate it. This present a mobile-cloud framework, which is shown in Figure 3.1. In this framework, all user's data is stored in the cloud storage, and smartphones only deal with some basic operations of apps, such as managing the apps and showing the result of them. Smartphone users can be totally free of managing their data and have larger volume to install more apps by putting all data into the cloud and let the cloud service to manage data and security. Although the security of a cloud service is not perfect now-a-days, cloud service providers are much more professional than app developers and users. Using cloud service, the operation of encryption and decryption of data can be finished in Cloud, and apps work as Data Requester requesting data from Cloud. This process can be implemented based on KP-ABE (Key Policy Attribute Based Encryption) [9] framework or its improvements such as PP-CP-ABE (Privacy Preserving Cipher Policy Attribute Based Encryption) [10]. Besides smartphones, this framework also support for other smart devices. By putting all data into remote cloud storage, it is not necessary for local device to have a hard drive which costs much space.

3.2 Data Flow Diagram

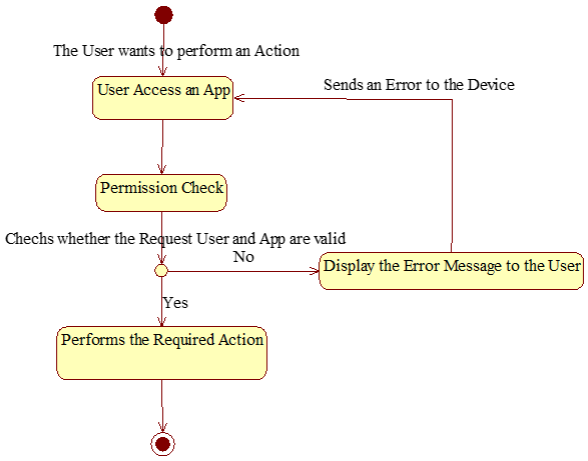

Figure 2 Working of An App

Figure 2 shows the step by step procedure of an app requesting the storage device from the smart phone using data overcollection prevention algorithm and finally evaluates permissions based on the algorithm. This algorithm is found to be better when compared to the other techniques.

3.3 Description of Process

Splitting the process reduces the complexity of the work. This paper is carried out in three different process as follows:

Storing the Data: Each app stores the data from the smart device into the cloud storage.

App access the data from the cloud: The app request the data from the storage device to the device.

Authorizing the app: The security risk of the data and the security risk of the app. Then judging that whether the authorization to the app will outstrip the default security risk. If it outstrips the security risk, authorization fails. If it doesn't outstrip the security risk, authorization passes.

3.3.1 Storing the Data in Storage Device:

This process describes the app, authorize to access the hardware of the user device. Since some app wants to use user's data sends its request to the storage device. The storage device access control service provides fine-grained permission authorizations for every app and is in charge of authorizing different permissions for various operations of apps. The user's data are stored in the storage device which can provide various levels of storage solutions for sensitive or normal data using encryption and decryption service, user's encrypted data can be decrypted and sent back to that app.

Input: appID, userID, data.

Output: Successful or unsuccessful message to the user device

3.3.2 App Uses the Data from The Cloud:

The app sends request of accessing User's specific data to Cloud. To authorize various apps with different fine-grained permissions, the Access Control Service has detailed lists about every operation of every app has what kind of permissions to user's data. When receiving a request, from some app for permission, it check these lists to find what specific permissions are authorized to this app, and return the result of yes or no.

Input: appID, userID, preview information of requesting data PD.

Output: concrete content of requesting data D.

3.3.3 Authorizing the App:

The security level of an encryption algorithm is based on its execution time. This approximatively equals the execution time of an encryption algorithm as its complexity. Furthermore, there is roughly a linear relation between consumption and encryption complexity.

To totally release the operation burden of users, this module has design to authorizing the permissions to apps. Users only need to set a default security level, similar to the security level of a browser. This security level can be expressed as "extreme high", "high", "normal", "low", and "extreme low". In our algorithm, we calculate the security risk of the data and the security risk of the app. Then judging that whether the authorization to this app will outstrip the default security risk. If it outstrips the security risk, authorization fails. If it doesn't outstrip the security risk, authorization passes.

Input: appID, userID, size of requesting data n and the type of the data t, default security risk DSR.

Output: The permission to access.

4. Result and Discussion

To simulate the data overcollection behaviors, four real smartphones and one simulative cloud is used to build a simple mobile-cloud environment. The approach is evaluated for feasibility and the performance is evaluated through extensive experiments.

4.1 Data Over-Collection Avoidance

We emphasize on two scenarios to avoid data overcollection simulated in mobile-cloud framework environment. Finally, to simulate the prototype of Mobile-Cloud framework, a

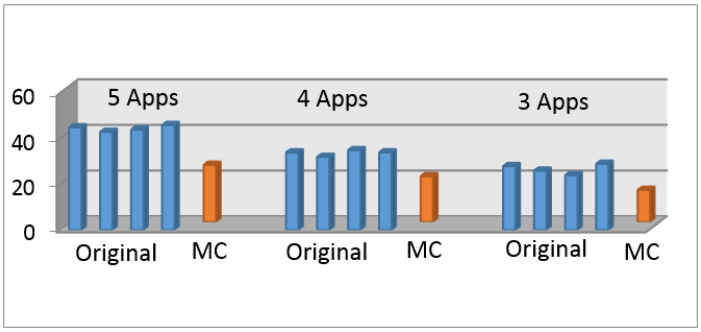

Figure 3 Security Risk

Figure 3 shows how the Service Risk is handled between the four devices and Mobile Cloud (MC) environment. These results are based on the usage of the physical properties and the security measures.

	Total	Used	App	Photo	Music	Movies
Device 1	64 G	16 G	10 G	242 M	4 G	0
Device 2	16 G	14 G	11.5 G	1.4 G	0	311.5 M
Device 3	16 G	7.8 G	529 M	2 G	270 M	1.52 G
Device 4	64 G	18.14 G	1.72 G	7.05 G	254 M	102 M

Table 1. Details of the Smart Phone's usage.

	UDID & Location	Photos	Contacts
Security Level	3	2	1

Table 2. Security level of the app uses the details from the smart devices.

The security risks of a smartphone is the sum of security risks of every apps installed in the smart phone. We evaluate four smartphones with five and four experimental apps separately in two environments, and the result is shown in Figure 3. The security risks of these four smartphones, installed five apps, are 41.5, 38.05, 38.91, and 41.51 in original environment, while in mobile-cloud framework environment the security risk is less than 25.95. Meanwhile, the security risks of these four smartphones, installed four apps, are 34.58, 31.58, 35.46, and 33.73, while the security risk is less than 20.76 in mobile-cloud framework environment. The security of these four smartphones, installed three apps, are 24.21, 21.62, 19.89, and 25.08, while the security risk is less than 14.69 in mobile-cloud framework environment. It is obvious that the security risks of smartphones in mobile-cloud framework are much lower than in original environment.

5 Conclusion And Future Enhancement

The granularity is the unit piece of data, which directly determines the effect of this approach. The original approach has the biggest granularity which ultimately leads to the data overcollection behaviors. In a contrast, finer granularity intrinsically supported by our approach ensures to protect data. Second, we set the default security level as the "normal". Assuming that an app sends a request to access to a piece of data with a particular level of security, the system can authorize this app to access data with a security level conforming to its privilege. Due to lack of consensus about what kind of data are highly sensitive, we assign security levels to different kinds of smartphone data based on the security level of data.

In future, the Mobile-Cloud framework should attempt to analyze the flow of the data in the app together effectively identifying the sensitive data of the user.

References

[1] Yibin Li, Wenyun Dai, Zhong Ming and Meikang Qiu (2015), " Privacy Protection for Preventing Data Over-Collection in Smart City", pp.1-11.

[2] Mobile Security Threats - Android Security Issues - Kaspersky Lab US , https://usa.kaspersky.com/internet-security-center/threats/mobile, 2016 , [Online; accessed 08-February-2016]

[3] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting privacy leaks in iOS applications," in Proceedings of the *18th Annual Network & Distributed System Security Symposium (NDSS)*, 2011.

[4] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: An information-flow tracking system for realtime privacy monitoring on smart phones," in *USENIX 9th Conference on Operating Systems Design and Implementation*, 2010, pp. 1–6.

[5] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android security," *IEEE, Security Privacy*, vol. 7, no. 1, pp. 50–57, Jan 2009.

[6] X. Xiao, N. Tillmann, M. Fahndrich, J. De Halleux and M. Moskal. (2012), "User-aware privacy control via extended static information flow analysis," in *IEEE/ACM 27th International Conference on Automated Software Engineering*, pp. 80–89.

[7] Fahim, A. Mtibaa, and K. A. Harras, "Making the case for computational offloading in mobile device clouds," in Proceedings of the *19th Annual International Conference on Mobile Computing; Networking*, Miami, Florida, USA, 2013, pp. 203–205.

[8] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 23– 32, 2013.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM 13th Conference on Computer and Communications Security*, 2006, pp. 89– 98.

[10] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in Proceedings of the *8th International Conference on Network and Service Management*, 2012, pp. 37–45.