The design of graph-based privacy protection mechanisms for mobile systems

Zhong Zhang
Dept. of
Computer Engineering
Myongji University
Yongin, South Korea
zhangzhong219017@hotmail.com

Sungha Yoon
Dept. of
Integrated Program of
Security and Management Engineering
Myongji University
Yongin, South Korea
ysh5811@gmail.com

Minho Shin
Dept. of
Computer Engineering
Myongji University
Yongin, South Korea
mhshin@mju.ac.kr

***Abstract*—In the range of mobile privacy, there are many attack methods which can reveal the user's private information. The attacker can use the communication between applications to violate permissions and access the private information without the user's authorization. Therefore, many researchers focus on privilege escalation. However, the attacker can increase their knowledge about the user without achieving privilege escalation through various inference techniques. For this reason, we extend the concept of privilege escalation attack to a more general information escalation attack, and propose a privacy protection mechanism based on the inference-graph.**

***Index Terms*—mobile privacy, information escalation, inference algorithm**

I. Introduction

In recent years the amount of malicious applications has been growing [1], [2]. Since the systems in mobile devices are different from personal computer (PC), existing protection mechanisms for PC are not compatible with the mobile device. Many researchers proposed mechanisms to protect the user's privacy in the mobile device, especially the location information. However, there still exist abundant different attacks on the user's private information utilizing various information either directly accessible through the platform or indirectly inferable from available resources. The lack of systematic approaches against such attacks restricts the development of privacy-preserving safe mobile platforms.

The mobile device contains a lot of private information. One of such sensitive information is location information. Applications can get the user's location by Global Positioning System (GPS). The mobile system manages the access permission of applications to the GPS information. Nevertheless, there are alternative methods for the application to trace the user's location. One of them is pedestrian dead reckoning. It is a navigation process using only the motion sensors, and the current location will be estimated based on the previous location along with the user's moving speed and distance [3]. Many researchers made accurate pedestrian dead reckoning algorithms to get either the user's indoor locations [4] or outdoor locations [5].

Motion sensors' information not only allows for the improvement of the location information [6], but it can even allow the attacker to infer the user's password [7], [8] or activities [9]. In addition to physical sensors, the attacker can leverage logical sensors such as short message service (SMS), call history and calendar to infer various information including the user's emotion [10]–[12] and social relations [13].

The device platform sets permissions of the application to the resources in the device, so that only the application with permission can access the permitted resources. Even so, the attacker can circumvent the permission mechanism by communicating with other applications with a different permission set; either by *colluding* [14], [15] with other malicious applications or by leveraging legitimate interactions with a benign application, called *confused deputy* attack [16], [17]. In literature this is called privilege escalation attack. Moreover, the attacker may promote the accuracy of the private information available to themselves simply by increasing the number of samples from the sensor. Such *inference attack* [8] leaks more information than allowed to the attackers, not necessarily violating the access permission policy. To model all such permission based attacks as well as inference-based attacks, we define *information escalation attack* as an attack, that the attackers can obtain more information or higher quality information than what they are allowed to have.

Recent studies were focusing on privacy mechanisms that either allow or disallow the access of the applications when it violates the privacy policy. We claim that, nevertheless, the privacy is not a matter of access control but more of information control. As privacy trade-offs with

978-1-7281-1288-6/19/$31.00 ©2019 IEEE

utility, the privacy policy that deliberately limits the amount of information revealed to third-parties provides more benefit to both the user and the service providers, than the all-or-nothing access controlling policies. To enable such privacy mechanisms, we need a systematic design of a privacy policy and an enforcement mechanism against information escalation attacks.

In this paper, we aim to contribute as follows:

- Define a generic privacy attack model called information escalation attack.
- Propose a novel privacy-policy model based on the inference-graph.
- We design our privacy-policy engine based on Android systems.

The rest of this work is structured as follows: Section 2 describes the privacy problem in the mobile device and introduces the concept of information escalation attack. Section 3 explains the concept of our approach and the system design.

II. PROBLEM FORMULATION

There are various kinds of mobile devices such as smart phones, laptops and tablets. Such mobile devices share to a great degree the same system model and attack model. In this section, we describe the system model and the attack model in detail. Our approach can be applied to any mobile devices under the same model.

A. System Model

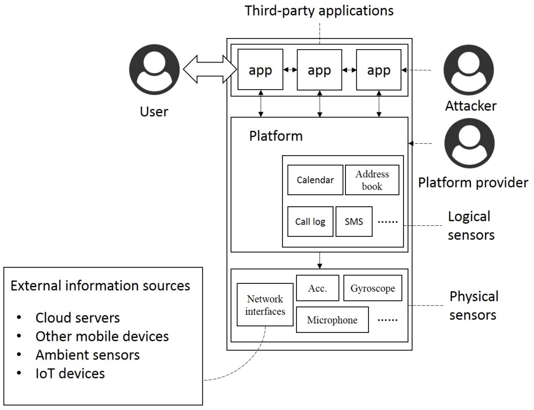

Fig. 1. System Model

Figure 1 shows the system model of the mobile device. The system consists of the application layer, the platform layer and the physical sensor layer. The user can install the third-party applications and interact with applications in the application layer. The third-party applications may come from the attacker. The platform layer is produced by the platform provider. Via offering an operating system (OS) and services, the platform enables the third-party applications to access the physical sensors (such as accelerometer, microphone, and gyroscope) and the logical sensors (such as address book, calendar, and SMS). The physical sensors and the logical sensors may directly expose the user's private information. For Example, a third-party application can learn the user's location using GPS sensor or the user's activities from schedules in the calendar. Especially, the logical sensors' information is usually the personal information.

The information, that the application can use, may come from the mobile device itself or from external sources such as the cloud servers, other mobile devices, ambient sensors, and Internet of Things (IoT) devices. Although information from external sources may not directly reveal personal information, it can still expose the user's context information such as location, physical environment, activities, and interests. In this paper, attacks relating external sources are out of scope.

Recent mobile platforms already provide permission control mechanisms either static (e.g. Android, version $\leqslant 5$) or dynamic (e.g. Android, version $\geqslant 6$) [18]. We propose a more sophisticated mechanism that supersedes existing permission mechanisms.

B. Adversary Model

The Attacker could be a malicious developer or a compromised application provider. Both of them can provide malicious application to the user.

The adversary may attempt to access the user's private information such as location, contract list, and password without permission. The obtained information might be sold to the data requester or be published. To access the user's private information, the attacker can develop malicious applications and upload them to application stores. The user may install malicious applications without notice. It is also possible that the attacker sends an email, which contains the download link of a malicious application, to the victims. The attacker can also develop a malicious website and force the user to download the malicious application with or without notice.

We only trust the system platform and the system-provided applications, which do not compromise to the user's privacy and are not considered to have vulnerability.

C. Attack Model

Although the mobile devices' system restricts the applications to access the information sources, the user's private information may still be revealed without the user's authorization. Under this situation, privilege escalation attack is discussed in many researches. However, we find out, that the escalation is not limited to privilege, the information can also be escalated.

Information escalation attack means that the attacker accesses more information without permission or gets information with higher accuracy than allowed. Privilege escalation is a part of the information escalation. If privilege escalation happens, the user's private information is revealed. Therefore, this concept is always in the range of information escalation.

- **Privilege escalation:** This attack is also named as permission escalation attack. The malicious application aims to access the protected resources in the mobile device. With this attack the malicious application can get more privileges than it intended and reveal the sensitive information without the user's authorization [17]. Privilege escalation could be achieved with confused deputy or colluding [19]. The original sensitive information can either be revealed without inference process or be used to infer other information.
- **Non-privilege information escalation:** When information escalation happens, the attacker might not need to violate any permission. The information that the attacker wants to get might be the information not including in the permission control or the information with higher quality. Non-privilege information escalation can also be done with confused deputy and colluding. The sensitive information can be revealed directly without inference or indirectly with inference. In the following text we will explain some examples.

Confused deputy is a kind of attack, that the attacker can control both malicious and compromised applications to violate permissions or access resources. For instance, the attacker can let the system browser download files or use system Scripting Environment to send text messages without the user's authorization [19]. In case of the non-privilege information attack, application A has information of the user's health status and application B does not. If application B uses vulnerability of A to get the health status information, this attack is non-privilege information attack via confused deputy. The system does not have permission control of the user's health status information, so this case is not a privilege escalation attack.

Colluding attack means, that a set of malicious applications cooperate to access more resources than only one of them can access. If these applications are developed by the same developer and have the same certificate, they can share their permissions and resources. In other words, if one application from the set of applications can access one kind of permission or resource, all the other applications in the same set can share its permission or resource [15]. In the example for confused deputy in the previous paragraph, if A and B are developed by the same attacker, A can send the information to B, in this case, A and B made a non-privilege information escalation attack via colluding.

Information escalation attack does not only mean, that the malicious application accesses not allowed information directly from another application, but also mean that the malicious application infers some other information, which is different from the information used as input to the inference process, or enhances the information's accuracy. Inference attack is an attack method that the attacker gains the sensitive information by data analysis. The attacker can access the information without directly permission [20], [21]. Combining the inference with the colluding or confused deputy attack, the attacker can make a more complex attack. In the previous mentioned example, application B can use the user's health status information and behavior information to infer the user's disease information. This attack is non-privilege information escalation with inference.

One example for combining the privilege escalation with inference, the application A has permission of GPS and the application B is allowed to access the accelerometer and gyroscope with high frequency. There is a pedestrian dead reckoning algorithm [6], which can infer more accurate locations with accelerometer, gyroscope, and GPS. The attacker makes colluding or confused deputy using A and B, so that B can use the information from GPS. With the previous mentioned algorithm B can infer the more accurate location. This location information is not allowed to B and its quality is higher than that from GPS.

We can set the control of all information available to the application to protect the user's privacy, when the attacker tries to get the sensitive information without inference. If the attacker makes an information escalation attack using inference, it is difficult to detect. Many researches about inference algorithms are made for various information. Consequently, information escalation risks many sensitive information. We attempt to make an approach, which can protect the user's privacy considering this kind of attack.

Besides colluding, confused deputy, and inference, there are other possible attacks for privilege escalation. The attacker can use the vulnerability from the system kernel to make the privilege escalation attack [22]. In this work we do not consider the privilege escalation attacks via the vulnerabilities of the platform itself.

III. APPROACH

Our approach is based on *inference escalation graph* that illustrates the inference relationships between different types of information. We monitor the data accesses dynamically for each application and check them against the information escalation graph to identify the possibly inferred information and their corresponding quality. We compare the accessible information and its quality with the privacy policy set by the user to determine the allowance of the access. When the access violates the policy, we either disallow the access or manipulate the returned values to meet the privacy policy.

A. Definitions of information escalation graph

Information escalation graph (IEG) is a directed graph consisting of nodes of three types and edges. A node represents an information type or information processing type. On the other hand, a directed edge indicates a direction of inference from raw data or less abstract information to more abstract or processed information. The following describes in detail three kinds of nodes and edges.

- **Source node:** A source node represents a physical or logical sensor denoted by a rectangle. A source node provides raw data to other types of nodes.
- **Process node:** A process node represents an implementation of inference algorithms taking multiple inputs and generating higher-level information. Such inference includes simple arithmetic computations to more sophisticated machine learning algorithms. A process node is denoted by a rhombus.
- **Information node:** An information node represents all types of information other than raw data. This includes simple combination of raw data or significant abstraction of raw data. An information node is denoted by an oval.
- **Edges:** An edge indicates the direction of inference from lower-level to higher-level information. An edge is indicated by an arrow.

B. Graph based approach

Figure 2 is an example of information escalation graph for three location trace inference algorithms using different combination of accelerometer, magnet and gyroscope. The inference algorithms use various processes to infer the various lower level information, which is desirable to infer the location trace. There are 5 processes to infer various information. The information to be inferred in this graph are moving distance, turn angle, and location trace. All the inference processes use time information. Process 1 is used to infer the moving distance using data from accelerometer [23]. Process 2, 3, and 4 infer the turn angle using accelerometer, magnet, and gyroscope. At the end process 5 infers the user's location trace using one location from GPS as start point, the moving distance, and the turn angle, which indicates the user's moving direction.

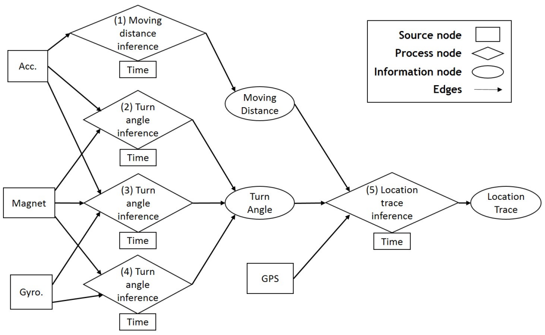

Fig. 2. An example of an information escalation graph

The three pedestrian dead reckoning algorithms in the graph use accelerometer to infer the moving distance. Differences between the three algorithms are their use of different processes to infer the turn angle. Algorithm 1 uses accelerometer and magnet, algorithm 3 uses magnet and gyroscope, and algorithm 2 uses all the three sensors. The three inference processes will return the turn angle information with different quality. The quality of the turn angle will then affect the quality of the location trace during inference process 5. Suppose the accuracy that the inference processes 2, 3, and 4 can provide are 80%, 90%, and 50%, and the corresponding inferred location trace accuracy are middle, high, and low. When the user allows the application to get the location information with the middle level, the application should be forbidden to use the gyroscope. If the user chooses to allow the high level, the application is allowed to use all the three sensors. But if the user wants to choose the low level, in this case, it seems like that there is no solution. Both the algorithm 1 and 3 need all three sensors to infer the location trace. The algorithm 1 can infer more accurate location trace than 3. If we disable accelerometer or magnet to break the inference of algorithm 1, algorithm 3 is also broken. We cannot disable algorithm 1 and at the same time allow algorithm 3. Under this situation, we can adjust the frequency of sensors to break algorithm 1 while keeping algorithm 3. Another similar situation is that the application needs all the three sensors to work normally. Our approach can keep the application's utility while protecting the user's privacy.

From this example we can see how the graph can be used to realize a privacy policy. The policy enforcement will be made based on some predefined logical rules.

C. System Design

In our system, the policy manager realizes the user's decision on their privacy into the privacy policy. There are also privacy service and policy database for further functions.

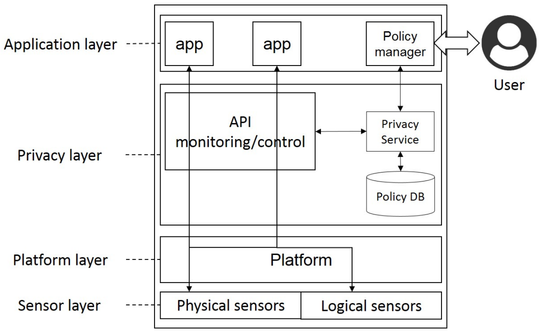

Fig. 3. System Design

Figure 3 is an overview of our system design. From the point of entities, in application layer the policy manager interacts with the user. The privacy service monitors and controls the application programming interface (API) usages from applications. The data in the policy database will be used to help the privacy service to make policy enforcement. From the point of work process, first, the API usages from each application will be monitored and collected by privacy service. From the API usages the privacy service can get the information that the application wants to access. With the help of the IEG in policy database the privacy service learns the set of information accessible or inferable from the set of resources the application is permitted to use. When the application tries to access certain resources causing an access to the banned information, the privacy service will block or allow the API call. In some cases, it can also manipulate the results of API calls to ensure the compliance to the privacy policy.

For instance, there is one application which is allowed to access accelerometer, magnet, and gyroscope and GPS. As explained in section two, this application can infer location with higher accuracy using pedestrian dead reckoning. Our approach detects the usage of resources from the application, predicts that the higher accuracy location can be revealed, and makes corresponding policy enforcement.

Since our system design is applied to Android systems, for the API monitoring and the data flow detection that we need for this work, we refer to API monitoring using Xposed [24], TaintDroid [25], and our previous work [26].

IV. CONCLUSIONS

In this work the concept of information escalation is introduced and explained with examples. The protection mechanism based on the IEG enables the user to control the private information considering information's quality. The policy can be applied to the physical sensors, logical sensors and the other information that we can monitor and control. We presented our system design with figure and explained it in detail. The future work will be the further development of protection mechanism and the evaluation.

REFERENCES

[1] R. Unuchek, (2018). Mobile malware evolution 2017. [online] Securelist.com. Available at: https://securelist.com/mobile-malware-review-2017/84139/ [Accessed 22 Nov. 2018].

[2] Mcafee.com. (2018). McAfee Labs Threats Reports Threat Research — McAfee. [online] Available at: https://www.mcafee.com/enterprise/en-us/threat-center/mcafee-labs/reports.html [Accessed 22 Nov. 2018].

[3] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC06), pp. 27–35, 2006.

[4] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," IEEE Sensors J., vol. 15, no. 5, pp. 2906–2916, May 2015.

[5] A. Mosenia, X. Dai, P. Mittal, and N. Jha, "PinMe: Tracking a Smartphone User around the World," IEEE Transactions on Multi-Scale Computing Systems, 2017.

[6] X. Zhu, Q. Li, and G. Chen, "APT: Accurate outdoor pedestrian tracking with smartphones," in Proc. IEEE INFOCOM, pp. 2508–2516, 2013.

[7] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in Proc. WiSec, pp. 1–12, 2012.

[8] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "ACCessory: password inference using accelerometers on smartphones," in Proc. HotMobile, pp. 1–6, 2012.

[9] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," Future Gener. Comput. Syst., vol. 81, pp. 307–313, Apr. 2018.

[10] B. Andrey, B. Lepri, and F. Pianesi, "Happiness recognition from mobile phone data," Social Computing (SocialCom) 2013 International Conference on, 2013.

[11] A. Bogomolov, B. Lepri, M. Ferron, F. Pianesi, and A. Pentland, "Daily stress recognition from mobile phone data, weather conditions and individual traits," in Proc. ACM Int. Conf. Multimedia, pp. 477–486, 2014.

[12] R. LiKamWa, Y. Liu, N. D. Lane and L. Zhong, "Can your smartphone infer your mood?," in PhoneSense workshop, 2011.

[13] A. Devlic, R. Reichle, M. Wagner, M. K. Pinheiro, Y. Vanrompay, Y. Berbers, and M. Valla, "Context inference of users social relationships and distributed policy management," in Proceedings of IEEE International Conference on Pervasive Computing and Communications, pp. 1–8, 2009.

[14] 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 18th Annual Network and Distributed System Security Symposium (NDSS), pp. 17–33, Feb. 2011.

[15] C. Marforio, A. Francillon, and S. Capkun, ”Application collusion attack on the permission-based security model and its implications for modern smartphone systems,” ETH Zurich, 2011.

[16] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin. ”Permission re-delegation: Attacks and defenses,” In 20th USENIX Security Symposium, San Fansisco, CA, Aug. 2011.

[17] L. Davi, A. Dmitrienko, A.-R. Sadeghi, and M. Winandy, ”Privilege escalation attacks on android,” In international conference on Information security, pp. 346–360. Springer, Berlin, Heidelberg, 2011.

[18] Android Developers. (2018). Permissions overview — Android Developers. [online] Available at: https://developer.android.com/guide/topics/permissions/overview [Accessed 22 Nov. 2018].

[19] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A. Sadeghi, ”Xmandroid: A new android evolution to mitigate privilege escalation attacks,” Technische Universitt Darmstadt, Technical Report TR-2011-04. 2011.

[20] J. Krumm, ”Inference attacks on location tracks,” International Conference on Pervasive Computing. Springer, Berlin, Heidelberg, 2007.

[21] G. Shafer, ”Detecting Inference Attacks Using Association Rules,” 2001.

[22] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer, ”Google android: A comprehensive security assessment,” IEEE Security & Privacy 2, pp. 35–44. 2010.

[23] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. Gutie andrrez Boronat, ”Comparison and evaluation of acceleration based step length estimators for handheld devices,” in Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on, pp. 1–6, 2010.

[24] S. D. Yalew, G. Q. Maguire Jr., S. Haridi, and M. Correia, ”T2Droid: A TrustZone-based dynamic analyser for Android applications,” In Trustcom/BigDataSE/ICESS, 2017 IEEE, pp. 240–247. IEEE, 2017.

[25] 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 smartphones,” ACM Transactions on Computer Systems (TOCS) 32, no. 2, pp. 5. 2014.

[26] M. Shin, and J. Kim, ”Privacy Preserving Watchdog System in Android Systems,” In Platform Technology and Service (PlatCon), 2017 International Conference on, pp. 1–5. IEEE, 2017.