Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-12, (2017)
ISSN (Online): 2454-1362

MobiGuard: A Mechanism for Protecting and Controlling user's Personal Data on Android Smartphones

Sijjad Ali Khuhro[1*], Attaullah Burio[2], Kimlong Ngin[1] & Danish Vasan[3]

[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230026, Anhui, China.
[2]Department of Information Engineering and Computer Science, University of Trento, Italy.
[3]Department of Software Engineering, Tsinghua University, China.

ARTICLE INFO

Received 9 September 2017
Accepted 30 November 2017
Published 4 December 2017

ABSTRACT

Smartphones are the most widely used personal devices, these days. They know much more about their owners as compared to their family members do. These devices are meant to move with owner's (so re-members the owner's movement patterns), keep records of the contacts, keep tracking messages (SMS, emails, social, etc.), records browsing history (sometimes including the passwords, credit card details, etc.), and the installed apps and the owners most favorite apps. All these apps generate and store a lot of personal/private data, thus, finding/accessing someone's phone could lead to knowing almost everything about their owner. Hence, it becomes extremely important to control the very personal data from the stolen smartphone, remotely, to avoid any sentimental and financial consequences. In this paper, we present MobiGuard – a solution for the Android smartphone users to control their smartphone, remotely. MobiGuard is a 4in1 solution that helps the MobiGuard enabled smartphone users in (i) finding their misplaced smartphones (in the house or in the office), (ii) turning back to the normal mode the "Silent" mode, (iii) locks the smartphone, and (iv) wiping off the personal data from their smartphones. We show how to control MobiGuard enabled Android smartphones through simple commands sent from the classical mobile phones besides Android smartphones. MobiGuard is simple, effective and user-friendly solution for the protection of their very sensitive personal data.

Keywords: *Smartphones, Software Engineering, Android, Security, Per-missions.*

1. Introduction

Smartphones have become extremely popular in recent years because of their smaller size, higher portability, continuous hardware improvement and their offer of anytime-anywhere computing. Thus, the dedicated Android app exists for almost all the user requirements, e.g., social networking (twitter, What Sapp), financial apps (PayPal, Paytm), etc. On the positive note, all of these apps are mainly designed to facilitate their users and improve their quality of life; however, all such apps continuously collect and store users' very personal data, on the other side. Hence, losing a smartphone could end up in an unauthorized access and could lead to the financial or sentimental consequences to the theft victim.

Authentication is considered as the first line of defense to prevent unauthorized access to the smartphone. Several solutions based on PINs/password [2], physical biometric and behavioral biometrics [3][4][5] exists, and however, the users are reluctant to use any of them on their smartphones because they consider them more

annoying than other technology related problems, such as lack of coverage, small screen size, or low voice quality [1]. Thus, it becomes extremely easier for phone finders or an adversary to access the smartphone data and masquerade the owners identity.

Several remote-wiping schemes have been proposed in recent years for the purpose of deleting the sensitive data from the lost smartphones [6]. However, they are triggered based on some events, e.g., certain number of failed attempts. Similarly, most of the proposed schemes, e.g., iCloud etc., require internet connectivity which can easily be avoided by turning OFF the Internet.

In this paper, we present MobiGuard - a system to remotely control the Android smartphone designed mainly to offer the following services: Firstly, it can find the misplaced smartphone in the house of on the office, secondly, Mo-biGuard, reverts back the adversary turned silent mode to the normal mode, thirdly, MobiGuard can lock the device, and finally, MobiGuard offers to wipe out the sensitive data (clear SD-card, contacts, etc.). MobiGaurd helps the owner to perform all these actions remotely from any other mobile phone (not necessarily smartphone). Every action is defined by a command and that command is sent using the cellular network to remote device to find its position, turn it back to the normal mode (from silent mode) and wipe out its users' related sensitive contents.

The rest of the paper is organized as follows: Section 3 illustrates the threat model and our assumptions. Section 5 enlists the expectations from MobiGuard. Section 5 discusses the building blocks of the scheme, and implementation of MobiGuard is discussed in Section 6. We evaluate our system using different software testing tools and present their results in the Section 7. We compare MobiGuard with the proposed schemes over the years in Section 8. Finally, conclusion concludes the paper by summarizing the paper and providing some future research dimensions.

2. Related work

Smartphones have become a very important personal device in everyday life. Due to their increasingly improved hardware and software transformation, they have become the preferred choice to access emails, perform banking transactions, social networking, etc. It has been announced by US military that they will equip their soldiers with Android devices for accessing classified documents. All the installed apps generate and store a lot of personal data on the smartphone memory, i.e., SD-card. Hence, the data security on the stolen devices rises serious and yet unresolved concerns. It becomes extremely important to explore the possible ways to cater such issues as the smartphone theft has been increasing continuously.

Several Anti-theft schemes for data protection on the smartphones, are pro-posed in recent years, however, in a little work has been done in line with our work[10][11][12] and hence we compare our work with them.

The model proposed by Kuppusamy et al. [12] controls remotely the stolen smartphones via Short Messaging Service (SMS). Similarly, the system proposed by Joe et al. [11] uses Wi-Fi to get the devices communicates between each other. The downside of this scheme is the requirement of Network connectivity between the devices and the need of smartphones on both sides.

Tang et al. [12] introduced Android based scheme, namely, CleanOS. CleanOS identifies, tracks, and encrypts the users' related sensitive data stored in the smartphone's memory, and sends the key to the cloud. The downside of CleanOS is its requirement of Internet connectivity with the cloud all the time.

Several commercial solutions for remotely wiping the data and post-theft access control, i.e., Find of iPhone of iCloud, Avast Free Mobile Security, and Norton Mobile Security, already exist. All these schemes help in remotely wiping the data on stolen or lost smartphones, however, all of them require Wi-Fi connectivity. We consider the apps such as Avast Free Mobile Security and Norton Mobile Security very close to MobiGuard as they are SMS-based remote control systems. These apps allow the users to wipe out their data by just sending a special SMS.

MobiGuard is different from the proposed schemes in the following ways: (i) Firstly, it does not require any Internet connectivity between the devices to delete the confidential data, rather, it uses cellular network to perform the required actions (as shown in the Figure 1), (ii) secondly, MobiGuard can lock the smartphone remotely as and when the user requires to lock, and (iii) finally, it reverts back

the owner/adversary turned "Silent" mode to the normal mode.

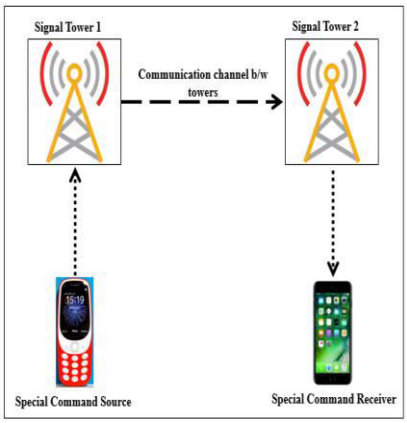

Fig.1. Illustration of our Approach

3. Problem Formulations and Solution

It is quite common that the smartphone user puts her smartphone in "silent" mode and forgets it somewhere. This situation becomes extremely annoying and frustrating for the legit user. Besides this, we assume that, the first thing the phone finder or an attacker would do will be to put the phone on the silent mode. MobiGuard helps the victim in easier and nicer way to revert back the user selected "silent" mode to the normal mode, thus, allowing the legit user to call on the number to track it.

Another situation could be the fact that the finder or attacker already possesses the misplaced or stolen smartphone. The finder or attacker obviously will try to access the device and its contents. Accessing of the contents can somehow be delayed because of the enabled authentication. However, the devices without these schemes are considered more vulnerable and the attacker could quickly access the device contents. We assume that the smartphone owner or legit user quickly realizes about the stolen or misplaced smartphone and hence can lock the device (in case it was unlocked) and could remotely wipe out the contents in no time.

4. MobiGuard System Expectations

This section explains the customer and developer requirements for the implementation of the proposed scheme. Functional requirements are the system functionalities the customer would demand from the system, whereas, the non-functional requirements are kept in mind by the developer to ensure the effectiveness, us- ability, and efficiency of the scheme.

4.1 Functional Requirement

These requirements describe the main customer demands from the developed scheme. These are considered very important because in any case if any of them is missing the overall system is considered incomplete. The main functional requirements of our proposed scheme, from user/owner point of view, are listed below:

-Account Creation

Requires user details for creating the account

- Account Login

Require User Name

Require User Password

-Phone No. Registration & Commands:

Add cell no. of other mobile.

Add command to change profile mode.

Add command to clear contact-list.

Add command to clear messages in inbox.

Add command to clear data in SD-Card.

Add command to clear Email/Facebook accounts.

- View Subscribed Phone Numbers & Commands:

Display the subscribed phone numbers & commands in users account.

- Delete Subscribed Phone Numbers & Commands:

Delete the subscribed phone number & commands in users account.

-Change Profile Mode:

Change the profile mode by ringing specific tone in full volume when Relative command is sent from subscribed phone number.

-Clear Contact-list:

Clear the contact-list when relative command is sent from subscribed Phone number.

-Clear Messages in Inbox:

Clear the messages in inbox when relative command is sent from subscribed phone number.

- Clear Data in SD-Card:

Clear the data in SD-Card when relative command is sent from subscribed phone number.

-Clear Email/Facebook Accounts:

Clear the Email/Facebook accounts when relative command is sent from Subscribed phone number.

- Lock Device:

Lock the mobile by setting login password when relative command is sent from subscribed phone number.

- Retrieve Commands by SMS:

Retrieve all commands by SMS when relative command is sent from Subscribed phone number.

- SIM Change Alert:

Send the phone number of new SIM by notification message on subscribed phone number.

4.2 Non-Functional Requirements

Non-Functional requirements are not necessarily requested by the users; however, the developers need to keep in mind these requirements in order to ensure the effectiveness (in terms of performance and usability, etc.) of the proposed system. We call them goals of the proposed scheme and they are listed below:

Multi-platform Delivery: The developed system should not be limited to just one device or one model and should rather have the multi-platform delivery attribute. MobiGuard is designed to run on all available mobile devices, such as Android powered smartphones, tablets smart watches (starting from Android version 5.0.1).

Usability: Usability is the very important non-functional attribute which reflects the user understanding and experience with the system. For example, highly secure but less usable authentication mechanisms are disabled by the users [7]. MobiGuard is interactive, easy to understand, easy to learn and easy to handle system.

Flexibility: Flexibility or the future enhancement ability is a very important non-functional aspect kept in mind during the designing of a new system. The system is tried to be designed so flexible that any technological advancement could easily be incorporated into the system. We claim that MobiGuard is extremely flexible because we followed all the object-oriented concepts in our minds while developing it and are hopeful that it can accommodate at least 10 years of technological advancements.

Efficiency: Efficiency here reflects the frequency and the corresponding required time with which the actions are performed. Our proposed system, i.e., MobiGuard is quick enough to transmit and complete the requested command, i.e., change the profile mode, etc.

Accuracy: The accuracy is another important non-functional requirement. The system must perform the desired functionality according to the command sent from another controlling mobile to the android mobile i.e. if the user wants to change the profile mode and sends the profile mode command from another mobile then the system must perform the desired functionality instead of doing something else, i.e., deleting the contact list. We claim that MobiGuard is extremely accurate in executing the commands and their corresponding actions.

Interface Portability: The interface portability is another important non-functional requirement. There are some devices which support Landscape screen mode like tablets and some devices like mobiles support Portrait screen mode. So our system should be designed in such a way that it must support Landscape and Portrait screen modes. Our scheme MobiGuard works perfectly on smaller screen devices (smart-watches, smartphones, etc.) and comparatively bigger (tablets 7&10 inches, etc.) Android Devices.

4.3 Process Model

We implemented our system MobiGuard using the incremental model approach defined in [8].We applied this approach mainly because of its easiness of division and management of the system. Additionally, we could track the system and could figure out the remaining work. We

modularize the system in some parts and then make increments by time at the end of each increment the working module is present. Overall system contains the three increments defined below:

1. Development of interface of the system by using Eclipse IDE.

2. Implementation of core features like change profile mode, clear contact-list, clear messages in inbox, and clear data in SD-card, clear email/Facebook accounts, lock device and retrieve commands by SMS.

3. Create the database to store the phone numbers and commands.

4.4 Android Architecture

Android is architected in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services, and libraries. Each layer of the stack and the corresponding elements within each layer are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices.

5. MobiGuard Design Philosophy

As the philosophy behind the implementation of MobiGuard is based on Object Oriented Analysis (OOA), we discuss our system in terms of OOA. MobiGuard is an Object Oriented System (OOS) and composed of various objects, meant to collaborate with each other using messages or parameters between them to know the behavior of the system. We discuss the modeling of the proposed system as below:

5.1 Actor identification

An actor is a person or a machine which is supposed to interact with the system. For MobiGuard, the user or owner is considered as the main actor and they are supposed to add phone numbers and commands and send these commands in the form of messages to control the Android mobile remotely.

5.2 Use case modeling

We applied Use-Case modeling approach, as it represents the abstract view of the system to analyze the functional requirements of the system [9].

5.3 Activity Diagram

Activity diagram basically is a sequential flow chart which represents the functionality of the system and shows how the function moves from one action to another action. The activity diagram of the proposed system is shown in the Figure 2. Activity diagram illustrates the different actions performed by our proposed scheme.

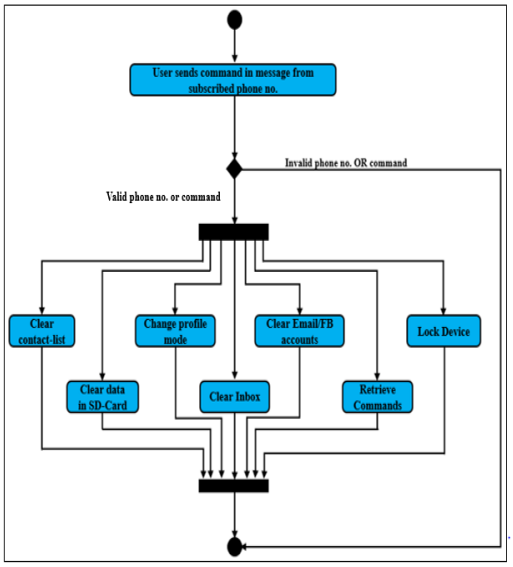

Fig. 2 MobiGuard activity diagram

5.4 System Scope

The primary purpose of the proposed scheme is to protect the very personal user data stored on the smartphone from the possible adversary. The system protects the contents of the smartphone before being accessed by the adversary, remotely. MobiGuard can help its users perform much-needed actions, such as clearing the contact-list, clearing the inbox, clearing the emails and different accounts and also can lock the device, etc. Additionally, it prevents "Silencing" the smartphone. Moreover, the scheme also helps smartphone owners to find their misplaced "Silent" (e.g., in their house or in their office) smartphones.

5.5 Interface

The user interface is an increasingly important aspect of a product that is often at least as important as the functionality of the system. Graphical User Interface (GUI) uses pictures and graphics for the input and output of the program. The main purpose of a Graphical User

Interface is to facilitate the handling of an application by means of graphical elements like text view, edit text and buttons.

6. MobiGuard Implementation

In this section, we discuss the different building blocks of our implemented system. We describe the required hardware, software, platform, and justify their selection. We show the screenshots of MobiGuard in the Figure 3. MobiGuard requires very little configuration, i.e., the user needs to create the account and add the corresponding commands.

6.1 Hardware Selection

We implemented MobiGuard specially targeting the new generation devices, i.e., smartphones, and tablets. We test the MobiGuard with the SAMSUNG GALAXY SIII Android device to verify the functionality of the MobiGuard. We did not rely on the Android emulator because the same does not fully show the entire functionality of the proposed system.

6.2 Software Selection

In software selection, we describe the platform used to develop the system and why we chose that platform. In software selection, we selected the platform Eclipse (Java-ADT v22.0.1-685705).Because it is the widely used platform for Android project development. One main reason for selection of this platform is that it provides an easy way to develop the system by providing the templates of graphical objects like buttons, text view, edit text etc.

In programming language selection, we selected the Java (Android) language because all the Android mobiles support Java language and it is one of the most popular programming languages. It is easy to use and therefore easy to write, compile, debug. Therefore, one of the main reasons for selection of this programming language is that the maintenance of the scheme in future will be easy to be used by developers.

6.3 Quality Consideration

Quality consideration describes the documentation of the source code and algorithm used in our system. In the documentation of source code, we add comments in the code, for the purpose of quicker understanding of code by the developers. In MobiGuard, we added important comments briefly describing the main purpose of each chunk of the code. Additionally, the developers are timely notified for any required maintenance by our proposed scheme.

6.4 Permissions Used

While implementing the MobiGuard, we added the following permissions in its manifest file to achieve the required actions:

- android.permission.RECEIVE_SMS
- android.permission.READ_SMS
- android.permission.WRITE_SMS
- android.permission.READ_CONTACTS
- android.permission.WRITE_CONTACTS

- android.permission.WRITE_EXTERNAL_STORAGE
- android.permission.AUTHENTICATE_ACCOUNTS
- android.permission.GET_ACCOUNTS
- android.permission.MANAGE_ACCOUNTS
- android.permission.USE_CREDENTIALS
- android.permission.BIND_DEVICE_ADMIN
- android.permission.SEND_SMS
- android.intent.action.BOOT_COMPLETED

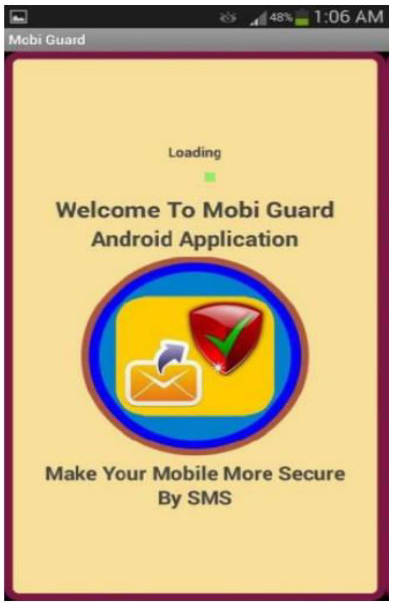

Fig. 3 Interface of Application

Fig. 4 Create Account

Fig. 5 Phone no. and commands

Fig. 6 Commands Details

6.5 Database Design

We rely on the built-in Android database to store the phone numbers of other mobiles and various action-specific (i.e., clear contact-list/inbox and SD-card and to clear the Facebook, twitter accounts), commands.

7. MobiGuard Testing

We test the implementation of MobiGuard using the well-documented techniques proposed in the literature. The basic purpose of the system testing is to check the functions of the system. This testing helps the developers in the identification of errors and debugs these errors to validate the functional requirements.

7.1 Regression Testing

We tested the MobiGuard by applying the Regression Testing (RT). In this testing, we checked whether or not the application responses correctly to the valid commands and its response to the invalid entry. In other words, we tested the modules of application in every aspect repeatedly to ensure the functional requirements of proposed system.

7.2 Black-Box Testing

Black box testing is also known as functional or behavior testing. This technique of software testing is applied when the user/developer does not know about the internal structure of the code or program. The testers know typically the input and the expected output; however, they do not know the inner side of the software, i.e., the actual processing of the input requests, etc.

7.3 System Evaluation

In system evaluation, we discuss us about the objectives of MobiGuard. The primary purpose of this testing is to confirm the achieved objectives. The proposed system is designed for mobile protection with the help of command in terms of messages sent by the other mobile. Further, it helps in switching the profile mode to normal if the phone is in silent mode, besides, locking the device if the smartphone is not locked. Furthermore, it helps in deleting the personal data to eliminate the chance of leaking it. By applying this system testing, we confirm that all the objectives are successfully achieved.

8. Conclusion and Future work

MobiGuard is a very useful system to protect the users' very sensitive data, remotely. On the lighter note, it is very common to forget the “Silent" profile enabled mode somewhere (in the office, or in the home). MobiGuard reverts back the smartphone to the normal mode and hence it can be found by calling to the number of that smartphone. Further, MobiGuard can do the same in the case that an adversary has turned on the “Silent" mode. Furthermore, MobiGuard can help the smartphone owner, in deleting remotely all the sensitive data, i.e., inbox, email/Facebook accounts, SD-card data, and/or other media Files in the case that the phone is lost. Similarly, it can also lock the device if the phone had no locking mechanism enabled. MobiGuard clearly is in the initial stages and we will keep on adding upcoming features and functionalities associated with the technological advancements. Additionally, we will also try to achieve SIM free communication between the two devices avoiding any involvement of cellular or Wi-Fi networks.

9. Acknowledgment

This paper work is supported by the National Natural Science Foundation of china under (No.61572454, 61572453, 61472382, 61520106007).

10. References

[1] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," in *Proceedings of the 4th USENIX conference on Hot topics in security*, 2009, pp. 9-9.

[2] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq, "Keystroke-based user identification on smart phones," in *International Workshop on Recent Advances in Intrusion Detection*, 2009, pp. 224-243.

[3] A. Buriro, B. Crispo, and Y. Zhauniarovich, "Please hold on: Unobtrusive user authentication using smartphone's built-in sensors," in *Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on*, 2017, pp. 1-8.

[4] A. Buriro, B. Crispo, F. DelFrari, and K. Wrona, "Hold and Sign: A Novel Behavioral Biometrics for Smartphone User Authentication," in *Security and Privacy Workshops (SPW), 2016 IEEE*, 2016, pp. 276-285.

[5] A. Buriro, B. Crispo, F. Del Frari, and K. Wrona, "Touchstroke: smartphone user authentication based on touch-typing biometrics," in *International Conference on Image Analysis and Processing*, 2015, pp. 27-34.

[6] A. Srinivasan and J. Wu, "SafeCode-Safeguarding Security and Privacy of User Data on Stolen iOS Devices," in *CSS*, 2012, pp. 11-20.

[7] A. De Luca, A. Hang, E. Von Zezschwitz, and H. Hussmann, "I Feel Like I'm Taking Selfies All Day!: Towards Understanding Biometric Authentication on Smartphones," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 1411-1414.

[8] I. Sommerville, *Software Engineering: Pearson New International Edition*: Pearson Education Limited, 2013.

[9] J. Kettenis, "Getting Started With Use Case Modeling: White Paper," *Oracle Corporation, USA,* 2007.

[10] I. Ion, N. Sachdeva, P. Kumaraguru, and S. Čapkun, "Home is safer than the cloud!:

privacy concerns for consumer cloud storage," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, 2011, p. 13.

[11] I. Joe and Y. Lee, "Design of remote control system for data protection and backup in mobile devices," in *Interaction Sciences (ICIS), 2011 4th International Conference on*, 2011, pp. 189-193.

[12] K. Kuppusamy and G. Aghila, "A model for remote access and protection of smartphones using short message service," *arXiv preprint arXiv:1203.3431,* 2012.