ZPrivacy Suite: Android Based Permission Manager Incorporated with Firewall TechnologyTitle

Siti Zulaiha binti Mon Sharif
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
Kuala Lumpur, Malaysia
siti.zulaiha@outlook.com

Dalilah Abdullah
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
Kuala Lumpur, Malaysia
dalilah@miit.unikl.edu.my

***Abstract*—The user privacy security is one of the major concern, especially in IT world today. With user's information, nasty user especially hacker can do lots of malicious activity, for example, identity theft, spam, social engineering and lots more which will harm the user. The main purpose of this project is to monitor and protect the android user from privacy leak while solving the default android problem, Take-it-or-Leave-it policy on Android platform running version 5.0 and below. In addition, the powerful Linux firewall, IPtables, also has been embedded as an extra layer of security which functions to reject unwanted or unnecessary internet access requested by the application which in return benefits the user to reduce the usage of mobile data and at the same time increase the battery life.**

Keyword: Privacy security, Hacker, Malicious activity, Take-it-or-Leave-it Introduction

1. Introduction

In general, the information privacy is the right to have some control over own personal information which is collected and used. Privacy in this project context includes the user data and information such as media, contact, email etc. A lot of advertising company now day try to collect user data for their own purposes and the security of this information is undeniable important in order to protect it from falls into the wrong hand. The malicious user can do lots of shady activity if they gained the information either for advertising and marketing purposes, or their own malicious purposes.

Even though Android has implemented some control over user privacy, it still not enough. The Android Take-it-or-Leave-it policy, for example, is problematic as an Android user can choose to install the application and granting all permissions required or simply, not install it without any option to control them. Furthermore, even the application originates from legitimate sources; the trustworthiness of the developer still vague as they still can collect a great deal of data, either for advertising and marketing purposes, or their own purposes. Some developer also set the apps with too many permissions but failed to list the reasons for each permission requested thus will increase the risk of privacy leak.

The Android-based operating system has been chosen as the main platform for this project. This platform was chosen as Android currently was a dominant mobile operating system in the world now (Cerrato, 2015). The program will be written in Java programming languages and will run on a variant version of Android from 4.0.3 - 5.1.1 (Ice Cream Sandwich to Lollipop). Compared to an existing Android-based permission manager, this application was able to monitor almost all android application permission request and prompt the user either to allow, deny or feed it with empty, fake or bogus data. The user also can set the access permission manually by opening the application. In addition, the powerful Linux firewall, IPtables, also has been embedded as an extra layer of security which functions to reject unwanted or unnecessary internet access requested by the application which in return benefits the user to reduce the usage of mobile data and at the same time increase the battery life.

2. Android Permission and Permission Manager

Dawson & Omar (2015) in their book divides the permission into four categories which is normal, dangerous, signature and signature-or-system. Signature and signature-or-system permission is quite rare to be used compared to normal and dangerous permission. The normal permission usually was granted without explicit user approvals, but for the dangerous permission; the user will be acknowledged first before the installation process. This is because the permission itself poses a high-security risk as providing the application an access to user critical and private data. The dangerous permission includes the permissions of: 1) calendar; 2) camera; 3) contact; 4) location; 5) phone; 6) sensor; 7) SMS and; 8) Storage.

The permission manager, on the other hand, is an application which used to manage the permission of installed application within an android operating system. This application managed to do this functions by using the various technique, for example, system level permission and repackaging the application.

The system level permission uses a heavy modification on Android OS, thus requires Xposed Framework with ROOT

Privilege to achieve this objective as it modify Android OS at the kernel level. In Android, the Zygote is the first process started after the booting sequence of android device. Every application in Android is a copy or fork of Zygote main process. This process loads the needed classes and invokes the initialization method. Xposed Framework interferes this component by installing a jar file inside /system/bin and activates it before the main method of Zygote was called thus automatically becoming a part of zygote itself.

This system level permission consists of three components which are Privacy Policy Database, Privacy setting content provider, and Privacy Setting Manager. When an application tries to read data, the request will be intercepted by content aware component and direct it to the privacy setting content provider to be checked against the privacy policy database. The Privacy setting Manager will allow, deny or spoof data based on database result.

Meanwhile, the repackaging the application technique opposed the system level permission by modifies the application manifest and repackaging them with fewer permissions. This technique consist of four component including 1) Permission removal; 2) Installation and execution; 3) Automatic UI Exploration and; 4) Crash Detection.

3. Android Firewall

Firewall is defined by (Michael, 2015) as a software or hardware-based network security system used to filter the incoming and outgoing network traffic based on the predefined rules. There is various type of firewall which available on the market now including Packets firewall, Stateful firewall, Application-layer firewall and Proxy firewalls. In Android, the two most common firewall used is from the category of Stateful firewall and proxy firewall.

The IP Tables is a type of Stateful firewall and was a built-in command-line firewall utility for Linux operating system. Even Android is build based on Linux, Google has removed this utility to gain stability and performance. To re-enable this utility, the existence of Busybox application and Root access is necessary. This firewall uses the policy chain to permit or block the network traffic. If the rule for any connection is not available in its list, the IP Tables will resort the traffic to the default action (Korbin, 2014). The command used to include input, forward, output and others.

The Virtual Private Network (VPN) is another firewall method widely used by the existing application to monitor and control device network traffic and falls under the proxy firewall category. However, it doesn't create real VPN connection to a server even though it seems so. The user can filter any application or which traffic they want to allow or reject. They also will receive an alert certain application tries to access the internet (Mihir, 2014).

4. Proposed System

Based on the finding and research, the developers come out with a new application which is ZPrivacy Suite. This application requires a special privilege, which is “root” permission in order to access advanced Android features in addition of Xposed framework and Busybox application, which is necessary to execute system level command and features.

ZPrivacy suite uses the system level permission method in order to manage the Android permission. By using this technique, the application was able to effectively modify the permission requested by any installed application while at the same time feed the application with bogus, fake or an empty data to avoid the application from being crashed.

This application will also use the IPtables features which come with the installation of Busybox application. As being mention before, the IPtables is a Stateful firewall which able to permit or block the network traffic. This application which can be used by any Android platform running on version 5.0 and below. The details were shown in table 1.0.

Some of the user interface (UI) which designed into an information device with which a person may interact with the developed system were shown on bellows figure. The figure 1.0 shows the main activity for ZPrivacy Suite where the user will be directed when they launch the application. The user can access any of three ZPrivacy module from here by clicking a related button. While figure 2.0 illustrates the permission manager module. All the installed application will be displayed in this section sorted by the type of permission they may request.

The firewall module user interface is shown in figure 3.0. The orange text stands for system application while the black text stands for user application. From here, the user can choose to block the internet access either from Wi-Fi, LAN or mobile data. Figure 4.0 shows the interface for device spoofer module. From here, the user can choose either manually provide the fake data or automatically generate random system data. They also had the option to randomize the data for each device reboot by clicking the given checkbox.

5. Development Methodology

Lots of research has been conducted since the title selection of the project to solve the possible problem regarding the project. The Rapid Application Development (RAD) is chosen as the development methodology for this project with an aim to develop the high-quality product faster.

The first stages of this methodology are requirements planning phase as shown in figure 5.0. This stage is important for this project to identify the proposed business process and getting familiar with the project. In order to collect information and gain more knowledge, the developer did a lot of research that related to the project. The result from the research that has been done contributes to a better understanding the project requirement, scope and the lack of current Android-based permission manager which existed in the market nowadays.

Table 1.0: Comparison table between ZPrivacy Suite and an existing application

	APK Permission Remover	App Ops	Droidwall Firewall	NoRoot Firewall	ZPrivacy Suite
Require "root" access	No	No	Yes	No	Yes
Requires xposed framework	No	No	No	No	Yes
Required VPN Services	No	No	No	Yes	No
Uses Method	Repackaging application	System level permission	IP Tables	Fake VPN	System level permission and IP tables
Permission Manager Features					
Able to modify application permission	Yes, support any permission by modifying the manifest file and repackage the application	Yes, support all permission from dangerous permission group	No	No	Yes, support all permission from dangerous permission group
Generate fake or empty data	No, thus prone to application crash	Yes	No	No	Yes
Firewall Features					
Able to restrict internet access	Yes, but crashes the application in some cases	No	Yes	Yes	Yes, by using iptables firewall method
Can filter all type of connection	Yes, but crashes the application in some cases	No	Yes	Yes, except LTE	Yes
License	Paid, free version can only modify up to 10 apps only	Free, open source	Free	Free, open source	Free, open source
Requirement	Android 2.3 and above	Android 4.3 – 4.4 only	Android 1.5 and above	Android 4.0 and above	Android 4.0.3 and above

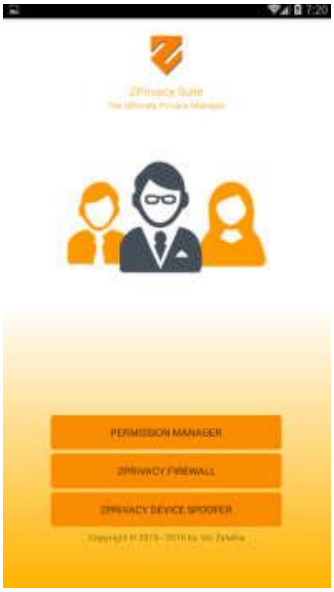

Figure 1.0:
ZPrivacy Suite main activity

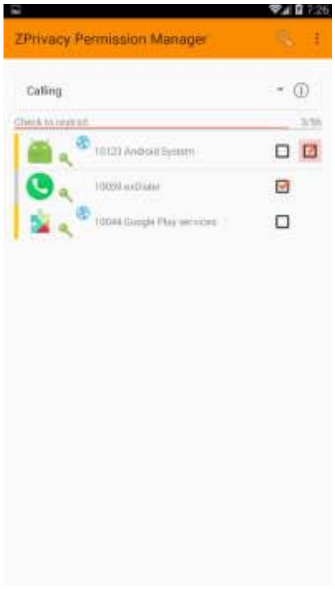

Figure 2.0:
Permission manager module

Figure 3.0:
Firewall Module

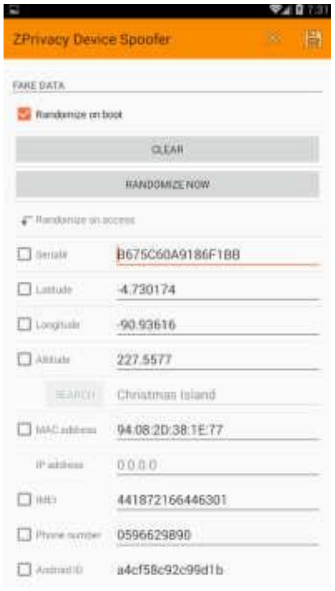

Figure 4.0:
Spoofer Module

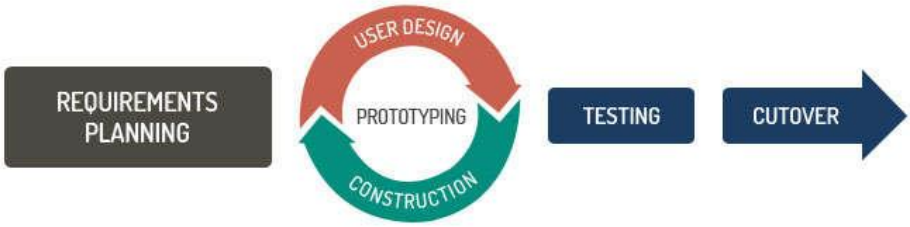

Figure 5.0: RAD Structure

The information which has been gained from the research was analyzed and the system requirement for the project is categorized into two which is a functional and non-functional requirement. The functional requirement for this project includes able to: 1) list selected permission of installed application; 2) Grant selected permission request; 3) Deny selected permission request; 4) Spoof data on selected permission; 5) On-demand selected permission restriction; 6) Allow internet access on selected application and; 7) Deny internet access on selected application. The non-functional requirement however for now only deals with the application response-time.

Figure 6.0 shows the Swimlane activity diagram for ZPrivacy Suite. This application can be used either by automatically or manually. If the application was planned to be used manually, the user itself can open the application, and manually set the permission for each installed application. Based on the permission that has been requested, the ZPrivacy suite will first check the permission against the available data within its policy database. If the request comes from a new application, the ZPrivacy suite will ask (On-Demand) whether to grant or deny the permission and store the response into the database for future reference. If the user grants the permission, the application will permit to access the requested data, while if the access id denied, the application will be fed by bogus, fake or empty data to avoid application from the crash. If on demand permission is enabled, ZPrivacy suite will automatically capture the permission request.

The actual test result for each module is recorded and associated in contrast to the functional requirement which previously specified previously as shown in table 2.0. This project can be considered as a success based on its reassuring achievement in the real test result where it was effectively meet the project objective. Regardless of this success, with the intention of ensuring the developed application able to function appropriately; the hardware and system also need to meet the specified project specification.

7. Conclusion

This project is finally done successfully and managed to achieve the proposed objective, but yet that are many changes can be added from time to time in order to make it better. This project development gives a lot of new input for the developer especially on how to plan and start a project, how to handle the process timeline, how to solve problem occurred during project development and the most important thing is to learn new programming language better than before. This project also helps to provide the developer to have a better insight on the famous Rapid Application Methodology (RAD) as it has been used as the main methodology for the development of this project.

References

Cerrato, I. (2015). Elements of domotic: Home automation, from a different point of view. Italy: Google Books.

Korbin, B. (2014, June 02). The Beginner's Guide to iptables, the Linux Firewall. Retrieved from How-to Geek: http://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall

Michael, C. (2015, December 07). Firewall definition. Retrieved from Tech Target: http://searchsecurity.techtarget.com/definition/firewall

Mihir, P. (2014, February 15). NoRoot Firewall Controls Internet Access for Apps. Retrieved from Lifehacker: http://lifehacker.com/noroot-firewall-controls-internet-access-for-apps-1523393751