Droid Guard: An Approach to Make Android Secure

Tushar Chopra
B.Tech CSE
GTBIT, GGSIP University
New Delhi 110064, India

Vikash Kumar Sharma
Institute for Systems Studies and Analyses,
DRDO
Delhi 110054, India

ABSTRACT
Mobile technology has been through a rapid development during last few decades. Smart phones used nowadays have dominated the traditional phone sets in many ways and they have become a usual gadget in our day to day lives. In this paper authors explain the development and utilization of multitasking security app named "Droid Guard" for mobile as well as personal security. Four security features namely Amplifier, Emergency call, Location and Theft security have been discussed and explained along with graphical illustration. Such security apps can prove handy not only in solving regular problems faced by the mobile users and can also be a life saver in case of emergencies.

Keywords
Global Positioning System, Internet Protocol, Subscriber Identity Module, Geographical Location, Short Message Service.

1. INTRODUCTION

1.1 Introduction and Basic of Android
Android is an operating system basically for mobiles. It is being developed by Google. It is based on the Linux kernel and is widely used in mobile phones. It is absolutely free and open source software similar to Linux. Mobile Development India has worked on various projects such as media player, gaming software, picture editors and many more. Android provides a very convenient and easy to handle hardware platform for developers enabling them to realise their ideas effortlessly. A basic diagram of Android architecture is shown in figure 1[1][2].

1.1.1 Linux kernel:
Among the several layers Linux – Linux 2.6 with approximately 115 patches is at the bottom. The basic system functions such as memory management, device management like camera , keypad etc is facilitated by this layer. Also, the kernel smoothly manages all the other things such as networking and a vast array of device drivers.

1.1.2 Libraries:
Above the Linux kernel there is a set of libraries available including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful for storing and sharing of application data, to play and record audio and video, SSL libraries responsible for Internet security, surface manager, media framework etc.

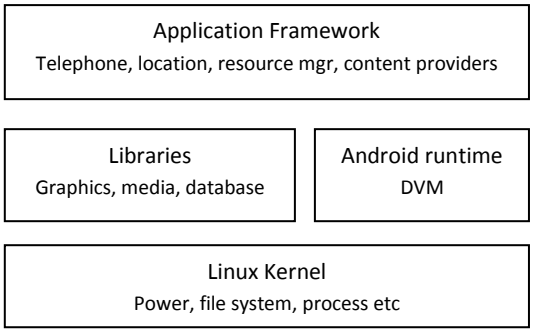

Fig.1 Android Architecture

1.1.3 Android Runtime:
This third section of the android architecture is available on the second layer from the bottom itself. It provides a key component (DVM) Dalvik Virtual Machine which is a type of Java Virtual Machine. The Dalvik Virtual Machine uses core features of Linux like memory management, process management and multi-threading etc. Android app can run in its own process by using the Dalvik Virtual Machine creating its own object. The Android runtime also has a set of core libraries which enable Android applications to be written using standard Java language.

1.1.4 Application Framework:
This layer utilizes Java classes to provide many higher-level services to android applications. Android application developers are allowed to make use of these services in their applications.

1.1.5 Application:
All the Android applications are placed at the topmost layer. Android developers write their applications in this layer only. Many examples of such applications include Books, Browser, home, Contacts, and Games etc.

1.2. Android activity Life Cycle
An activity stack is used to keep track of all the activities in the system. When a new activity created, it is placed at the top of the activity stack which will become the new running activity of the system. The older activity which had actually come before the running activity always keeps below the new running activity in the stack and won't come to the top until that new running activity exits.

The figure 2 displays the activity state paths [3][5][6]. The white rectangles represent callback methods when activity moves between states which can be implemented to perform various operations. The colored ovals are the major states in which an activity can be. The callback methods shown in figure 2 are discussed below:

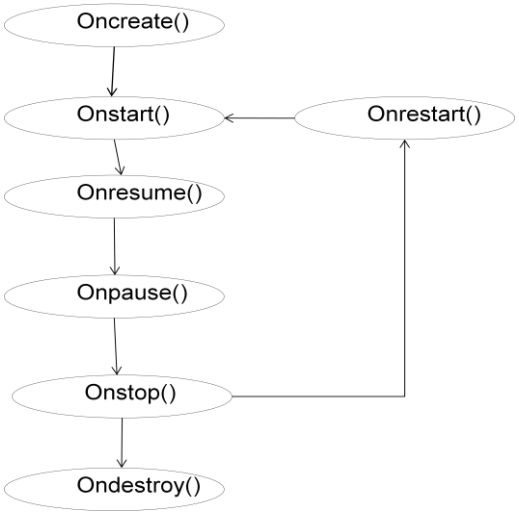

Fig.2 Activity Life Cycle

- onCreate() is called on activity creation. Here, all the static set up is done like creating views etc. It also stores the activity's previous frozen state, if present any. It is always followed by onStart().
- onRestart(): When an activity needs to be started again after it has been stopped, onRestart() is called before restarting it. It is always followed by onStart().
- onStart():When an activity got visible onStart() is called after on create().
- onResume(): When the user provides input to the activity and it is present at the top of the stack, onResume() is called. It is always followed by onPause().
- onPause (): When the system needs to resume a previous activity, onPause () is called. It is used to accomplish things which consume CPU like committing unsaved changes to the persistent data, animations stopping etc. It should be implemented very fast so that the resumption of the next activity can take place which will only happen after the return of onPause ().
- onStop(): When an activity is not visible to user, onStop() is called. Next, depending on the user activity, onRestart(), onDestroy(), or nothing is called. This method may not be called ever when the memory is too low because the system will not have sufficient memory to store activity's running process.
- onDestroy(): When an activity needs to be destroyed, onDestroy() is called. This can happen in two cases: first, if an activity is finishing and finish() is called and the second when the system is destroying the object of the activity in order to save space.

2. PROBLEM FORMULATION

Authors address firstly the regular problems faced by a mobile user. For example it is usual for a person to misplace his mobile in home, car or office, which is very frustrating. This situation becomes further annoying, if the mobile was in silent mode. The second problem addressed by the authors is to distinguish the important calls from the regular calls, especially during meetings, conferences etc. For example during a meeting, Steave has put his mobile in silent mode and may not be interested in any regular call other than the calls from a particular group (e.g. his parents, boss etc). This utility can also be helpful to attend emergency call during silent mode of mobile. The third feature of Droid Guard is the location security, which is a very important characteristic of Droid Guard for personal security. Lastly the theft security has been explained for the safety of stolen mobile. These four interesting problems are faced by most of the mobile users and have been addressed in the present paper using android platform. The authors used following mechanisms and mobile services to solve the above four problems and named them as:

Amplifier Security:

This type of security can be used to increase the volume of phone or remove it from silent mode via a simple message or keyword through SMS (Section 3.1). The users will now never waste their important time and effort in locating the misplaced mobile.

Emergency Call Security:

Choose the numbers and your phone will never be on silent mode for them (Section 3.2). The users will never miss the calls of their near and dear ones.

Location Security:

Choose the message or keyword you want others to SMS you so that your phone reverts back with your Latitude and longitude automatically (Section 3.3). Your loved ones will never be worried about your location even if you miss to inform them about your whereabouts.

Theft Security:

A way to locate the stolen or lost mobile (Section 3.4). The chances of locating the dropped or stolen mobile instrument increases with this mechanism.

3. APPROACH

3.1 Amplifier Security:

In this type of security, the authors used message service of android device. It is used to read new incoming message. First user enters the string or identifying sentence or word in the application which will act as a keyword and will be used later to identify that a request for increase in the volume of android device is made. Now every time the SMS is received, the application checks whether the content of SMS is same as that of the keyword. If the application found the same string or sentence then it will increase the volume of the android device.

3.1.1 Algorithm:

Step 1: User enters the keyword and mobile numbers which are stored in the android application database.

Step 2: Incoming message received by the android application.

Step 3: Broadcast receiver reads the content and mobile number of new incoming message.

Step 4: If the content and mobile number matches with the predefined keyword and mobile number

Step 4.1 TRUE

⇨ Increase Volume

Step 4.2 False

⇨ Do nothing, Exit

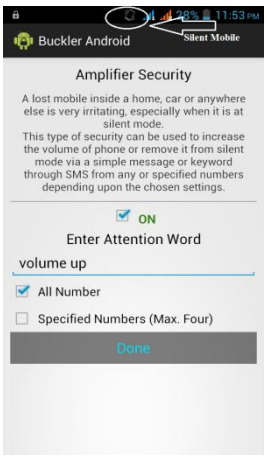

Fig.3 User saving keyword in application

Fig.4 Volume goes up on receiving keyword

3.1.2 Permission Used:

(i) Receive SMS[8] in android
(ii) Send SMS[8]in android

3.1.3 Broadcast Receiver Used:

(i) Receive SMS in android

Example: Alex is in hurry and he needs to make an urgent call or he is getting late for the party but he misplaced his mobile in home or office in silent mode. At this point he can send the keyword 'volume up' (set in figure 3) from any other device or mobile via SMS (see figure 4), the volume of the mobile will increase.

3.2 Emergency Call Security:

Here user enters the mobile numbers of a particular group of people (say his/her near and dear ones, boss, important client etc) in the application as shown in figure 5. This mechanism uses incoming call service of android device. Every time when an incoming call comes to the android device, the application reads the number of the incoming call and if it matches with the pre-set mobile numbers in the application then application will remove meeting mode or silent mode of the android device and the mobile will ring at its full volume as shown in figure 6. This mechanism can be of great use allowing the user not to miss the calls of his/her near and dear ones in case of urgency.

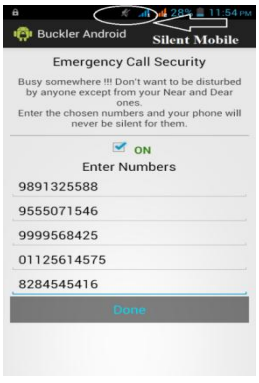

Fig.5 User saving mobile numbers in app

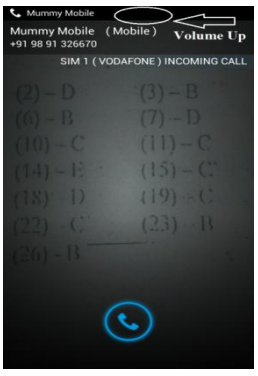

Fig.6 Volume goes up for same incoming number

3.2.1 Algorithm:

Step 1: User enters the mobile numbers which is stored in the android application database.
Step 2: Incoming call received by the android application.
Step 3: Broadcast receiver reads mobile number of new incoming call.
Step 4: If the mobile numbers matches with the predefined mobile numbers
Step 4.1 TRUE
⇨ Increase Volume
Step 4.2 False
⇨ Do Nothing, Exit

3.2.2 Permission Used:

(i) Read phone state [8] in Android

3.2.3 Broadcast Receiver Used:

(i) Access phone state in Android

3.3 Location Security:

As discussed earlier, this is a personal security feature of Droid Guard based on message service of the android device. It is used to read new incoming message. First user enters the string or identifying sentence or word in the application which will act as a keyword and will be used later to identify that a request is received. Now the application will read every incoming message. If the content message matches with the pre-saved keyword in the device which is entered by the user, the device will automatically start using global positioning system (GPS) service and internet (IP address) service which are used to access current location (latitude and longitude) of the device.
Actually GPS is a space-based satellite navigation system which is used to provide location and time information anywhere on or near the Earth. It is freely accessible to anyone having a GPS receiver and is maintained by the US government. Android device nowadays has GPS receiver that can be used to access their location. The IP address uses the geolocation systems, which works using WHOIS service and receives the registrant's physical address. IP address location data may have other information such as country, city, postal/zip code, latitude, longitude and time zone. From the above two methods, the location of the android device can be accessed. After successfully accessing the location of the device this location is sent via SMS to the same number from which the request was made.

3.3.1 Algorithm:

Step 1: User enters the keyword which is stored in the android application database.
Step 2: Incoming message received by the android application.
Step 3: Broadcast receiver reads the content of new incoming message.
Step 4: If the content matches with the predefined keyword and mobile numbers
Step 4.1 TRUE
⇨ Location of the device (lati, longi) accessed by using global positioning system[9] and internet service of the android mobile and is sent via SMS to the same number from which the request was made[10].
Step 4.2 False
⇨ Do Nothing, Exit

3.3.2 Permission Used:

(i) Receive SMS

(ii) Send SMS
(iii) Access coarse location
(iv) Access Fine location
(v) Access network state
(vi) Internet

3.3.3 Broadcast Receiver Used:

(i) Receive SMS in Android

Fig.7 User saving keyword

Fig.8 Location tracked

Fig.9 Location on Google Maps

Example: Tara's son went to coaching. He usually returns at home around 6 pm but he didn't come home. It is 8pm now and he isn't picking the phone up as well. At this point Tara can get the location of her son by just sending a SMS (say 'location') in figure 7, to his son's android device and device will revert back his location via SMS to her mobile automatically (figure 8). Now she can use Google maps (figure 9) to locate and find the exact location of her son.

3.4 Theft Security:

It's a way to locate your stolen or lost mobile. Every mobile uses a Subscriber Identity Module (usually called SIM) card, which has a unique serial number. This serial number can be used as a key feature to identify when the new SIM is inserted in the android device.

In this security, the user enters the mobile numbers in application to whom he/she wants to send the details of the stolen mobile shown in figure 10. Application will save all these numbers along with the present unique SIM card serial number. Now each time when the mobile or android device is rebooted, the application will match the current SIM card serial number with the previously saved one. If both of the serial numbers are found to be different then it will start GPS and IP address service to access the current location and application will send the location of the device along with the new number via SMS to all those numbers which were previously saved by the user in the application as shown in figure 11.

Fig.10 User saving the mobile numbers in application

Fig.11 Location and new mobile number tracked

3.4.1 Algorithm:

Step 1: User enters the mobile numbers which are stored in the android application database along with unique SIM id.

Step 2: When the device is rebooted, the application matches the unique SIM id from the database with the current inserted SIM id.

Step 3: If the new SIM id didn't match with the pre stored SIM id from the database.

Step 3.1 TRUE

⇨ The location of the device (lati, longi) is accessed by using global positioning system[9] and internet service of the android mobile and is sent along with the current active mobile number via SMS to all the pre defined mobile numbers in the application as previously saved by the user[10].

Step 3.2 False

⇨ Do Nothing, Exit

3.4.2 Permission Used:

(i) Send SMS
(ii) Access coarse location
(iii) Access fine location
(iv) Access network state
(v) Internet

3.4.3 Broadcast Receiver Used:

(i) Boot completion in Android

4. CONCLUSIONS

In the present paper authors explain the development and utilization of multitasking security app named "Droid Guard" for mobile as well as personal security. Authors provide with a mechanism to find the misplaced mobile (in silent mode) by sending a predefined keyword via other devices. In the second problem, authors distinguished the important calls from the regular calls which can be helpful to attend emergency calls during meetings, conferences etc. Next the Droid Guard's location security was discussed and shown with the help of an example. Lastly, the theft security has been explained for the safety of stolen mobile. These four interesting problems are faced by most of the mobile users and have been addressed in the present paper using an android platform. These security apps can prove handy not only in solving regular problems faced by a mobile user, but can also be a life saver in case of emergencies.

5. REFERENCES

[1] Tiwari Mohini. (2013, Nov) Review on Android and Smartphone Security, Research Journal of Computer and Information Technology Sciences, [online]. 1(6), pp.12-19. Available: www.isca.in/COM_IT_SCI/Archive/v1/.../3.ISCA-RJCITS-2013-030.pdf

[2] Kirandeep. (2013, Mar) Implementing Security on Android Application, The International Journal Of Engineering And Science, [online]. 2(3), pp.56-59. Available: www.theijes.com/papers/v2-i3/Part.Vol.%202.../I0232056059.pdf

[3] Vaibhav Kumar Sarkania. (2013, Jun) Android Internals, International Journal of Advanced Research in Computer Science and Software Engineering, [online]. 3(6), pp.143-147. Available: www.ijarcsse.com/docs/papers/Volume_3/6_June2013/V3I6-0134.pdf

[4] Yogita chittoria. (2014, May) Application Security in Android-OS VS IOS, International Journal of Advanced Research in Computer Science and Software Engineering, [online]. 4(5), pp.1432-1436. Available: www.ijarcsse.com/docs/papers/Volume_4/5_May2014/V4I5-0847.pdf

[5] Wei-Meng Lee, March 2012, Beginning Android 4 Application Development,Wrox,560 p

[6] Reto Meier, March 2010, Professional Android 2 Application Development, Wrox,576 p.

[7] James Steele, October 2010, The Android Developer's Cookbook, Addison-Wesley .Professional, 400 p.

[8] Android Open Source Project. Security and permissions. http://developer.android.com/guide/topics/security/permissions.html. (2013).

[9] Android GPS Location Manager. http://www.androidhive.info. (July 2012) .

[10] Android Sending SMS, http://www.tutorialspoint.com