Lockme – Android Security Application

[1,] Sumaiya Patel, [2,] Darshana Thakur, [3,] Sujit Sherkar,
[4,] Priyanka Dhamane,
Information Technology Department
[1,2,3,4,]Padmabhushan Vasantdada Patil Pratishthan's College of Engineering, Mumbai, India

Abstract: This paper presents an Android based App which is a security application. The idea behind this project is to develop an application which will help user of android to create Admin and Guest accounts like other computer based operating system. Security holes in Android operating system occur due to the permission based security model which is not properly enforced during system design. Permission based security model has central role hence it creates security holes in Android OS [1]. Google's android security model erased by its openness

Keywords: Android OS, Permission based security model, central role

I. Introduction

Android is a new, next-gen mobile operating system that runs on the Linux Kernel. Android Mobile Application Development is based on Java language codes, as it allows developers to write codes in the Java language. These codes can control mobile devices via Google-enabled Java libraries. It is an important platform to develop mobile applications using the software stack provided in the Google Android SDK. Android mobile OS provides a flexible environment for Android Mobile Application Development.With the increased pace of life and shrinking time people need everything in the palm of their hand, everything happening with a touch. Hence it all boils down to one point, which system provides the best applications? Android is winning the race globally, accounting for more than 50 percent of the market as of 2012. A tablet may be shared by many people in a organization or at home by family members. So there is a need for the owner to allow a guest account login, the way it is in Desktop PC. This feature is needed, in organizations where individuals are given company handsets.There are many situations where in a foreign (not the owner of handset but works within the same org as the owner) user may face the necessity of using the actual owner's phone (in case the owner is not present in the vicinity) in placing an internal call (call within the organization) or send emails or simply send a group sms. This functionality is precisely the one available on Desktop OS.

2. Existing System

By default the android phone allows only a single user sign on [2]. This person is the owner or the administrator for the phone. Owner has complete privileges. The current android Operating System [3] does not have a facility to create user account.

3. Proposed System

The device will facilitate user of device to create two user accounts viz.

1) Administrator
2) Guest

The Administrator has all the rights and permission to access all apps. The Guest has call and camera blocked. The guest has no permission to access the camera. She/he cannot make a call.In this application, we ensure that screen lock password of sufficient length is set up before displaying the secure content, as well as the screen lock timeout is also set.It also has an option to disable the camera. It stores all the policy that we set up, e.g. minimum password length, password complexity, screen timeout, etc. Once users enable the device admin application, they are subject to its policies. Complying with those policies typically confers benefits, such as access to sensitive systems and data.If users do not enable the device admin app, it remains on the device, but in an inactive state. Users will not be subject to its policies, and they will conversely not get any of the application's benefits—for example, they may not be able to sync data.

3. Technology Used

For developers:

OS : Windows XP or above
IDE : Eclipse.
SDK : android SDK.
Emulator : any android emulator.
System architecture : 32/64 bit.

For users:

OS : Android
Hardware : any android phone
Size on disk : minimum 1 MB.

4. Classes Used

BroadcastReciever

android.content.BroadcastReceiver When a matching event is generated in the system, Android delivers the event to that broadcast receiver. Applications with Broadcast Receivers registered in the manifest don't have to be running when the Intent is broadcast for the receivers to execute. They will be started automatically when a matching.This is excellent for resource management as it lets you create event-driven applications that will still respond to broadcast events even after they've been closed or killed.

4.1 DevicePolicyManager

This service is provided through the **android.app.admin.DevicePolicyManager** class. The device administration API provides device administration [4] features at the system level. It allows development of security-aware applications that are useful in enterprise settings

Design

Android GUI is single-threaded, event-driven and built on a library of nest-able components. The Android UI framework is organized around the common Model-View-Controller pattern.

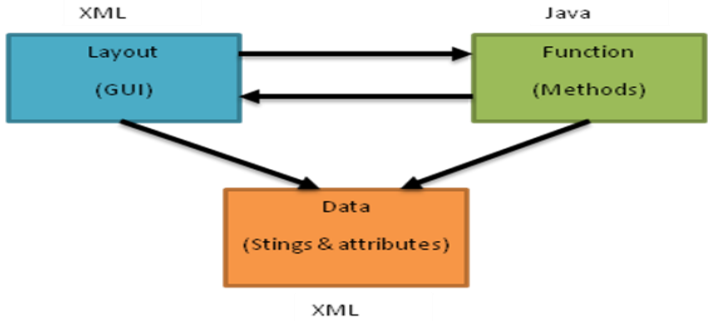

Fig Structure of the Application Design

Basically there are two ways to create a User Interface in Android, either through XML or by creating the UI Programmatically. The diagram shows the different tools used in developing the application for different purposes.The first step is GUI i.e. graphical user interface that is developed using the XML tags. Each form in Android is an Activity.To code for the function and for interconnection of the Activities we use Java. Java provides methods which will help to achieve all the task.

Block Diagram

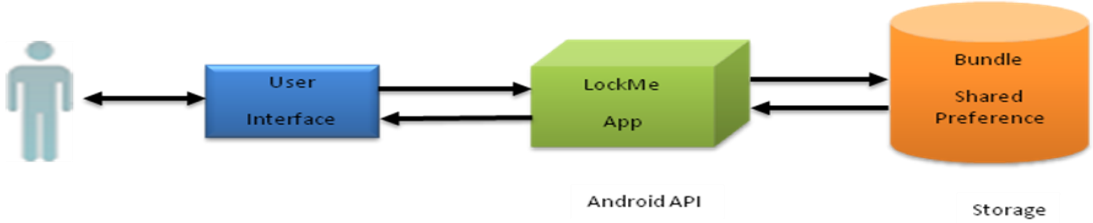

Fig: Architectural Block Diagram

The block diagram shows the architecture of the Application. The user will interact through User Interface which will consist of:

- Setting up Password policies
- Logging on to device via Admin or guest account

The LockMe app which is built on Android API will setup the policies for the password such as Numeric, Alphanumeric or Mixed as shown.

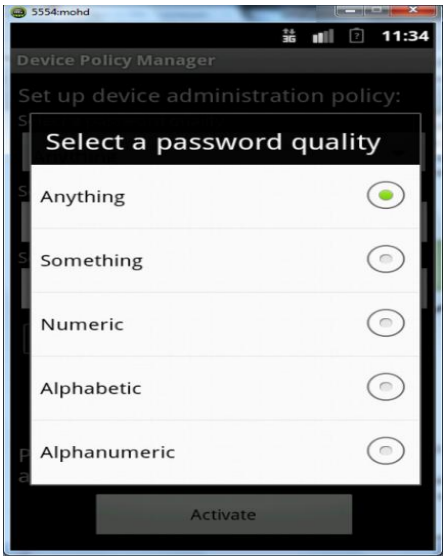

Fig: Selection of password quality

It will also block the calling facility and camera is user is logged on as Guest. For the storage of policies we use Shared Preferences where the policies are written to and read from while logging in. Finally, all the data that is used by the Java and XML is stored in form of values using XML.

4.2 Future Scope

The application has many options as its future scope. Development of mobile security application will be emphasized on following aspects:

1) Can be used to lock media folders for user's privacy.
2) Can be used to lock other apps.
3) Can be used to protect inbox so that no one can view messages.
4) Can be used enhance pattern changing.

5. Conclusion

Thus we have proposed an application targeted for android mobile and tablet users create a security-aware application that manages access to its content by enforcing device management policies. When device is protected using this application owner of device decides which content of device will be user see and access it. This application makes changes in OS through application this will help the easy to protect the device from unauthorized user. If other user gets access to device he/she will make change in the system or remove the security policies from device. This application protect user from doing system changes.

References

[1] Android.com," Available: http://www.android.com
[2] W. Enck, M. Ongtang, and P. McDaniel. Understanding Android security. IEEE Security & Privacy Magazine,7(1):10–17, 2009.
[3] Technical Blog of Sai Geetha dedicated to Android, http://saigeethamn.blogspot.com/
[4] Sayed Y Hashimi and Satya Komatineni, "Pro Android", Wiley India Pvt Ltd. (2009)
[5] Android Developers official website, http://developer.android.com/guide/topics/ui/index.html
[6] A visual interface editor for Android, http://www.droiddraw.org/
[7] Static detection of malicious code in executable programs by J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi.
[8] Android project. http://source.android.com/.