International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

# Efficient Integrity Protection for Android Smartphone

**Pranjali N. Watkar[1], S. P. Karmore[2]**

[1]Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India

[2] Professor, Department of Computer Science and Engineering, G. H. Raisoni College of Engineering, Nagpur, India

**Abstract***: In worldwide, the number of Android users increasing day by day as it is efficient and easy for usage. Android is a mobile operating system which is based on Linux kernel having a user interface based on direct manipulation. The problem is that so many viruses and malwares are detected on Android device during handling of device through Bluetooth and Multimedia message service. Another major issue is there are lack of procedures and requirements for identifying subjects as trusted. The main focus of the study is the prevention of application, whether the application contains malwares or not. User needs to take the permissions before using the applications because applications might contains malwares which are responsible for the damage of application and device integrity. In this paper, we proposed the Efficient Integrity Protection model for Android devices which checks the integrity of the device with the other Android device through the WAMP server which helps to remove the damaged application. All Android devices are protected with the OAuth protocol which provides authorization and protects the device from unauthorized access. We studied different integrity protection models for the smart phones which were implemented first on FreeBSD platform and Linux platform.*

**Keywords:** Android Smartphone, Mobile security, Integrity Protection, NFC, WAMP server, OAuth protocol

## 1. Introduction

The virus and malware detection has become the central component of a security system protecting mobile networks [1]. The rapid development of smart phones has facilitated the need for better protection against malware. Furthermore, malwares can jam wireless servers by sending thousands of spam messages or track user positions through GPS [2]. Modern mobile phones are not very different from desktop computers in terms of the range of applications they can support. Different development platforms allow anyone to develop an application for the mobile device. Mobile devices are rapidly becoming attractive targets for malicious attacks due to significant advances in both hardware and operating systems.

Nowadays, the development in mobile technology and services has greatly increased the possibility of serious malware attacks on a mobile network [4]. While transactions some malicious programs may damage the software and hardware or even steal important private information. The main focus of the study is the prevention of application, whether the application contains malwares or not. For this purpose, we studied the existing attacks on operating system. All the applications in Android device will be scanned. The size of the particular application will also been display in the list of application. It shows the size of the application like small, medium or large. The data contains in the application denotes its size i.e. it is small, medium or large. It shows an Alert Dialog to report the application. At last the total number of Activities and Services will be count.

On the other side, existing exploits in mobile phones have shown that user downloaded and installed applications are major threats to mobile services. Nearly more than 370 malwares are detected on Android devices since last two years. Most of the infections are due to user downloaded applications like Dampig, Fontal, Locknut and Skulls [1]. Also many other major virus mechanisms include Bluetooth and multimedia message services like Cabir, CommWarrior and Mabir [1]. Many exploits compromise with the integrity of a mobile platform by maliciously modifying data or code on the device. So, we implemented the boundary for trusted and un-trusted domain. User always asked for the Permissions before accessing the applications in threat of malwares [12]. Different colour codes are provided to detect the malwares and spywares. Android has the security system of permission mechanism; the user has to allow the permissions of all requests and cannot limit the permission based on specific conditions in the installation of Android applications. We allowed the applications to broadcast the intents.

In this paper, we consider the attacks from the application level and implemented the efficient integrity protection model on Android device which checks the integrity of the device with the other Android device through the WAMP server which helps to remove the damaged application. All Android devices are protected with the OAuth protocol which provides authorization and protects the device from unauthorized access [13].

The rest of the paper is organized as follows. In Section 2 we explained some preliminaries for the Android security, while we discuss the related work in Section 3. Privacy issues are presented in Section 4, while existing techniques and its limitation are discussed in Section 5. Section 6 covers the Database Connectivity in details. In Section 7 we described the OAuth (Open Authentication) protocol for privacy issues. Concluding section discusses limitations of the current implementation, and envisages possible future work.

## 2. Android Security

Android is a Linux-based mobile platform which is developed by the Open Handset Alliance (OHA). Most of the Android applications are mainly programmed in Java and compiled into a custom byte-code that is run by the Dalvik virtual machine (DVM). On a Linux, file access permissions are set for three types of users: the owner of the file, the users who are in the same group with the owner of the file and all other users. For each type a tuple of read, write and execute permissions is assigned. In Android, by default, the files in the user's home directory can be read, written and executed by the owner and the users from the same group as the owner. All other users cannot work with these files. So as different applications by default have different user identifiers files created by one application cannot be accessed by another [12].

For example, consider an application that needs to monitor incoming SMS messages, AndroidManifest.xml included in the application's package would specify:

<uses - permission android:name = " android.permission. RECEIVE_SMS" />. Permissions declared in the package manifest are granted at the installation time and cannot be modified later [12].

## 3. Related Work

The main focus of study belongs to the integrity protection of the android devices. Before accessing any application in the android phone, permissions should be taken by user to avoid the unnecessary abuse of application, as it is a security mechanism of android phone.

### 3.1 Permission based Mechanism

In the latest version of Android, there are total 130 permissions [12]. Different permissions in Android phone are shown in Figure1, with its categories.

| Category | Android Permissions |
|---|---|
| Contacts | WRITE_SYNC_SETTINGS, WRITE_CONTACTS, READ_CONTACTS, MANAGE_ACCOUNTS, GET_ACCOUNTS, ACCOUNT_MANAGER |
| SMS-MMS messages | WRITE_SYNC_SETTINGS, WRITE_SMS, VIBRATE, SET_ORIENTATION, SEND_SMS, RECEIVE_SMS, RECEIVE_MMS, READ_SMS, FLASHLIGHT, BROADCAST_SMS |
| Call Log | VIBRATE, PROCESS_OUTGOING_CALLS, FLASHLIGHT, CALL_PHONE, CALL_PRIVILEGED |
| Audio & Video | WRITE_EXTERNAL_STORAGE, SET_ORIENTATION, RECORD_AUDIO, MODIFY_AUDIO_SETTINGS, GLOBAL_SEARCH, FLASHLIGHT, BLUETOOTH, BLUETOOTH_ADMIN, CAMERA |
| Tasks & Calendar | WRITE_CALENDAR, REORDER_TASKS, READ_CALENDAR, GLOBAL_SEARCH, GET_TASKS, BLUETOOTH_ADMIN |
| Browser History | WRITE_SYNC_SETTINGS, WRITE_HISTORY_BOOKMARKS, READ_HISTORY_BOOKMARKS |
| Images | WRITE_EXTERNAL_STORAGE, SET_WALLPAPER_HINTS, SET_WALLPAPER, SET_ORIENTATION, READ_FRAME_BUFFER, GLOBAL_SEARCH, FLASHLIGHT, BLUETOOTH, BLUETOOTH_ADMIN, CAMERA |

**Figure 1:** Permissions in Android phone

This study collected 535 Android applications, there are 202 normal applications and 60 malwares; the testing set consists of 213 normal applications and 60 malwares. Normal applications are hot recommendations of Google Play, most of which are downloaded by users most with high scores.

Figure 2, shows the application info which asked the user certain permissions like storage, network communication and phone call before accessing the application.


**Figure 2:** Permission request in the installation of an application

### 3.2 Filter based Mechanism

To avoid the entries of the malicious program through the trusted and non-trusted domains, filters are added between the trusted and non-trusted domain. Filters such as firewall, authentication process or program interfaces need to be added before accessing the applications. Certain rules are applied to control the information flow between high and low entities shown in Figure3.

The integrity based rules define the flow control based integrity from the high and low attribute or entities. The set of information defined in the attributes it contain types of subjects. These subject mainly consumed with create, read and write operation. The integrity rules are defined under the different subjects or object suppose 'x' and integrity level is defined under L(x).Table 1 shows the integrity rules.

Process is created by using Rule 1. When Process is created its integrity level is checked by using Rule 2 with comparing two attributes or subjects. Rule 3 defines the read process of different attribute with lower and upper bound at the same level i.e. reading process is done by using read command. Rule 4 defines for the write operation on different attributes; the low integrity process is applicable for the write process. Rule 5 both read and write operation is done by using this rule which is applicable for the subjects in the process. The last rule i.e. Rule 6 defines the change of level i.e. the high integrity process can be change to low integrity process.

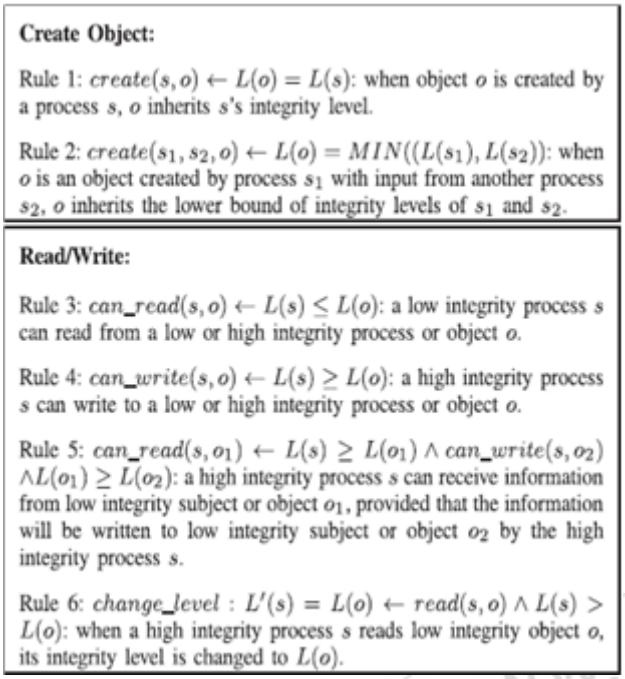


**Figure 3:** Integrity rules applied on Android phone

### 3.3 Resources responsible for compromising the integrity

Three main goals of integrity model are as follows:
a) Preventing the unauthorized users from making modifications to the data or program.
b) Preventing the authorized users from making unauthorized or offensive modifications.
c) Protect the external and internal consistency of programs and data.

Mainly, there are four different resources can easily compromise the integrity of mobile platform described as follows:

1) *Operating system resources:* The operating system resources mainly consist of the trusted computing base of operation, system configuration and various types of services. Attacks to these resources can easily compromise the integrity of the whole platform [1].
2) *Resources of network service provider:* A mobile device usually consists of sensitive data from network service provider. Data stored in device may get unauthorized access from unknown user can compromise the running behaviour of the device and communications between the device and wireless network [1].
3) *Device data and status settings:* The latest mobile device provides many status setting functions. It Manipulate these sensors without permission from the user can cause unexpected behaviour of a device [1].
4) *Resources of mobile user:* Many online service providers store user data on mobile device side like online banking and entertainment services which are the main targets of the Malwares. It send text messages and made hidden phone calls to premium phone numbers, additional cost generated in user account by different malwares [1].

### 3.4 Online operations

It will ask the user, whether he/she wants to provide the integrity to the application or uninstall the application or report to database via integrity violated option.

### 3.5 Database Connectivity

We are checking our phones integrity with the other phones application through DBF android integrity. First go to the online checking then see the score of the application. For example, the score may be 0 or 1. Then go to the WAMP server, check the score. If the score is positive then the application is integrity protected. Otherwise, it will ask for the un-installation of the application if the score will deduct by -5.

### 3.6 Color codes

Different color codes are provided to check the malwares in the application. Color code indicates the malwares found in un-trusted application like Gmail, Browser and many more by filter mechanism. To access these applications, android requested for the permissions. Suppose at a time an application contain two viruses it will add both the colors and form the combination of color. It will indicate both the viruses are found while scanning the applications.

## 4. Privacy Issues

According to the android security, the app stores email attachments making them accessible to the third party user that has access to the phone. There are three parties in the OAuth eco-system: Provider, User and App. In order to read or write a User's object, the third-party App should get the corresponding authorization from User. Users should be aware that the cases like they often expect their phones emails are protected while using the mobile application.

To preserve the privacy of the applications open authentication is provided by the OAuth protocol. OAuth protocol is secure, easy to use, scalable and general purpose. OAuth 2.0 protocol is used for delegated authorization which enables the user to grant the third party application access to the web resources without sharing the password. It allows the third party application to access the HTTP service. The authorization server gives the access token to the client after successful authentication by user.

Only one user can sign up with the Kerberos server as IMEI number of the device taken as user id because it is unique so that no other user can make account in same device. In case the unauthorized user will try to make new account it shows the report that database is existing in the server. Password, Contact and Profile will be entered by user while signing up for the first time. Until the entries of the default report not get deleted from the server, it would not give access to signing up with the new password.

In the WAMP server, the report of the authentication shows in the database with the IMEI no, Password will be shown in

hashed format as it is encrypted by SHA-1 algorithm so that no other user could see the password. Only after login user can access the device.

## 5. Problem Definition

We have done comparative study of different integrity protection model [6] [7] [8]. In BIBA model, certain limitations are found like lack of procedures and requests on recognizing subjects as trusted [6]. In CLARK-WILSON model there is lack of certification ensuring that programs are trusted [7]. In LOMAC model, it does not require any changes to pre-existing kernel or application configuration [8].

So considering the above limitations, we proposed efficient integrity protection model which deals with the Procedures, Requirements, Service, Authentication, Service, Providers and Receivers while handling the android devices. To fulfill the given requirements, Filters should be added between the trusted and non trusted domain while downloading the data from internet or receiving from Bluetooth. To avoid the unauthorized access to the application, Permissions are important and frequently needed before accessing the application as explained in Section 3.

## 6. Implementation of Efficient Integrity Protection model

In this research, there is a new approach for implementation of the "Efficient Integrity Protection model" which is explained in the given figure below. The design goals are adding the permissions and filters between trusted and untrusted domain, connectivity of the databases, providing open authentication for the user.

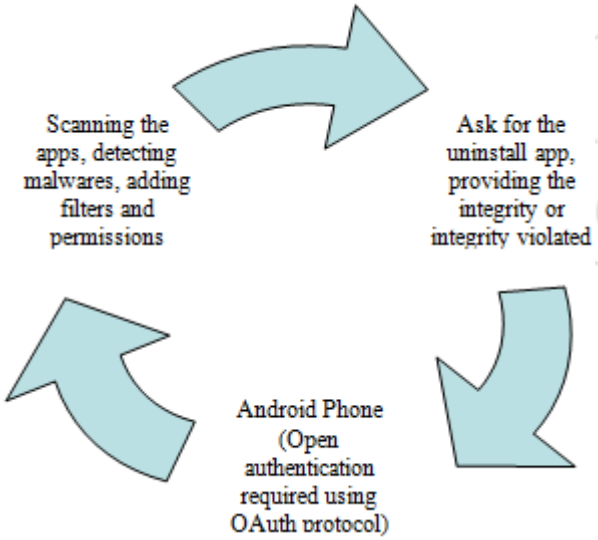


**Figure 4:** Efficient integrity protection model which deals with the permissions and filters

### 6.1 Scanning of the Applications

All the applications in Android device will be scanned first. The size of the particular application will also been display in the list of application. It shows an Alert Dialog to report the application. At last the total number of Activities and Services will be count.

### 6.2 Implementation of the boundary line between trusted and un-trusted application

There is an implementation of the boundary (filters here) for trusted and un-trusted domain. Then user asked for the Permissions before accessing the applications. Different color codes are provided to detect the malwares and spywares. We asked the user if it needs the permission. Check if confidence index is more than 0.5. So we can add this app to the filter category. Filters such as read, write and read/write added to the database. Check if the app was found in any category. We allowed the application to broadcast the intents.

### 6.3 Database Connectivity

We have checked our device integrity with the other phones application through Dbf android integrity. First go to the online checking then see the score of application. For example, the score may be 0 or 1. Then go to the WAMP server, check the score. If the score is positive then the application is integrity protected. Otherwise it will ask for the uninstallation of the application if the score will deduct by -5.

### 6.4 Open Authentication to avoid Privacy issues

User needs to sign up with IMEI number which will be taken automatically from the device manufacturer, also needs to enter the password, phone no. and profile before login. Only one user can sign up to the device, to avoid the unnecessary authorization. The entries will be displayed the database. The password will be showed in hashed format which is encrypted with SHA-1 algorithm.

To avoid the privacy issues while communication between the android devices, Open authentication protocol is been provided. This protocol used for the authentication process like Login into the devices. Same authentication process is provided to the android device before accessing the application.

IMEI number taken as a user id and password will be entered by the user during login. Android user need not to sign up every time while using device because only one user can handle the device. Once the sign up is done, the report automatically goes to the WAMP server and displayed the dialog that the report is already submitted. The user login details will be showed in the database of WAMP server. The password is encrypted by SHA-1 algorithm.

## 7. Experimental Results

In this research, there is an implementation of the integrity model which gives the stepwise following results:

### 7.1 List of scanned applications

All the applications in Android device will be scanned first. The size of the particular application will also been display in the list of application.

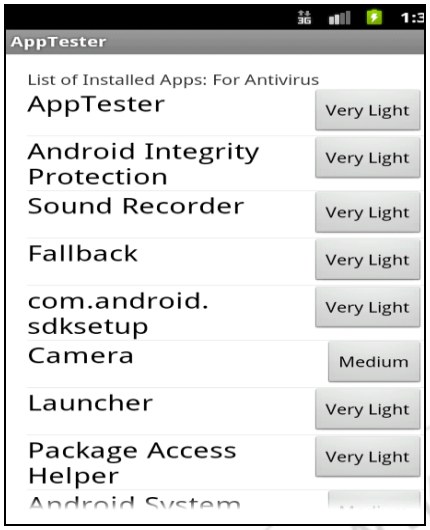


**Figure 5:** Snapshot of the Scanned applications with its size

The activities in the Android device given in figure 6; are explained as follows:

- *Instrumentations:* It allows the application to monitor all of the interaction the system has with the application.
- *Configuration Preferences:* These store the application configurations on the device memory.
- *Provider:* Providers allow applications to share data between each others; more the number of providers, higher the exposed data of the application to the other application.

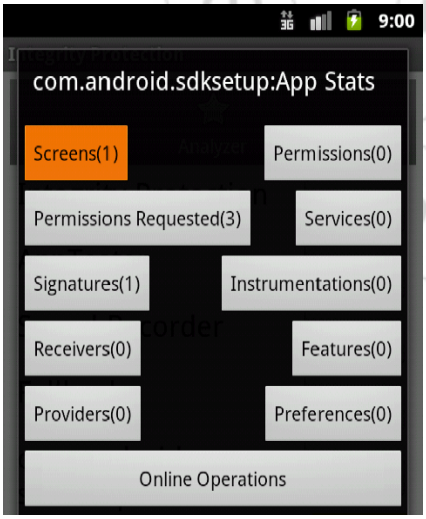


**Figure 6:** Total number of Activities provided to the application

- *Receivers:* It allows applications to receive system wide events like incoming calls, messages and more.
- *Requested feature:* This feature allows the applications to use features on the device.
- *Services:* These allow applications to run services in the background of the device; which can be used to send/receive background data.
- *Activities:* These are number of screens/layouts installed by the application.
- *Permissions created:* These permissions are created by the application.
- *Permissions requested:* These permissions are requested by the application.
- *Signatures:* The number of authentic certificate/signatures used by an application.

### 7.2 Implemented the boundary line between trusted and un-trusted application

We implemented the boundary line (filters here) for trusted and un-trusted domain. The user asked for the Permissions before accessing the applications. Different color codes are provided to detect the malwares and spywares. The user will ask for the permission if it needs the before accessing the application.

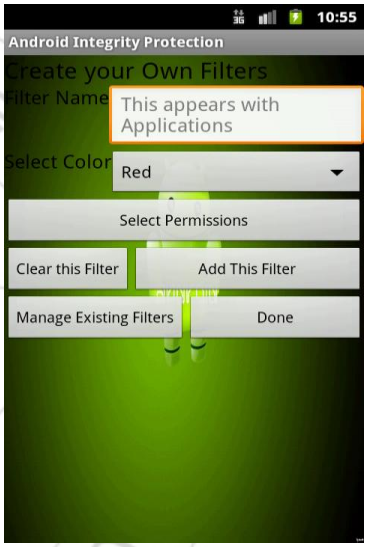


**Figure 7:** Snapshot for creating the filters

### 7.3 Database Connectivity

We have checked our device integrity with the other phones application through Dbf android integrity.

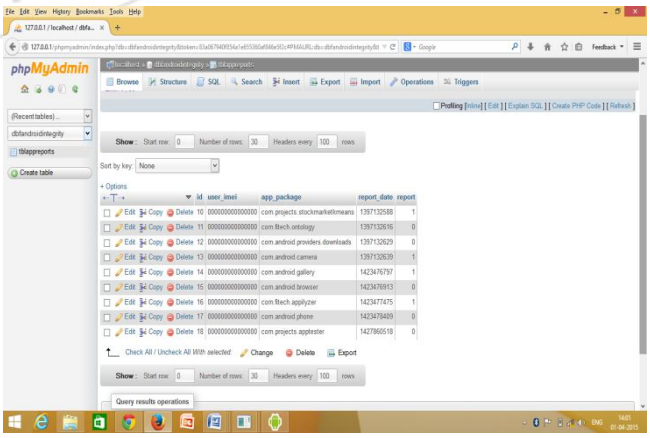


**Figure 8:** Snapshot of Database connectivity with the WAMP server

### 7.4 Open Authentication to avoid Privacy issues

To avoid the privacy issues while communication between the android devices, Open authentication protocol is been provided by Kerberos server. Only one user can signed up with the device as IMEI number kept as a user id, so that no other user access the device by making the sign up again.

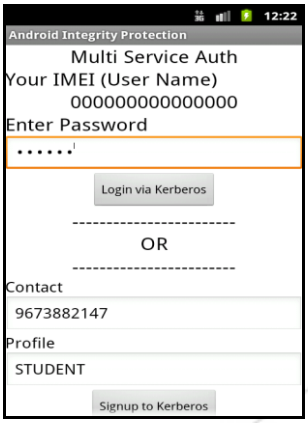


**Figure 9:** Open authentication by OAuth protocol

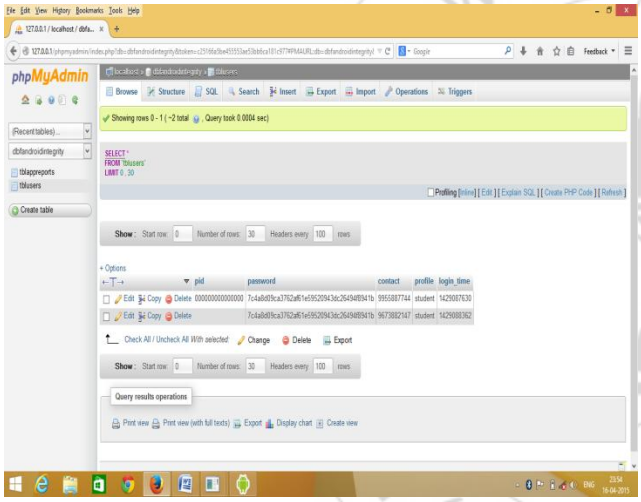


**Figure 10:** Snapshot of Database entries of Open Authentication Protocol in the WAMP server

Once the sign up is done, the report automatically goes to the WAMP server and displayed the dialog that the report is already submitted as shown in figure 10.

## 8. Conclusion and Future Work

Integrity protection is a major issue in the android phones as so many malwares are detected while using the android devices. Various types of malwares are responsible for the violating the applications in android. Moreover, lack of procedures, requirements and authentication are needed for avoiding the unauthorized access to the application. The certification rules and policies required for the easy access to the application. In this paper, we have presented our work on the integrity protection. Also presents the methodology used for this research. We proposed a new design of integrity protection model; we named our work as "Efficient Integrity Protection model". The design goals like adding the permissions and filters between trusted and untrusted domain, connectivity of the databases, provide open authentication for the user are achieved.

While we believe that our work has several claims of achievements, there would also be some weaknesses. In addition, several ideas have occurred during work on this research, and opened up several further avenues for exploration. Still we can implement the same integrity protection model and integrity rules on IOS, Windows platform and many other mobile operating systems in future by overcoming the issues in our work.

## References

[1] Xinwen Zhang, Jean-Pierre Seifert and Onur Aciic¸mez, "Design and Implementation of Efficient Integrity Protection for Open Mobile Platforms", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 13, NO. 1, JANUARY 2014
[2] Chao Gao and Jiming Liu, "Modeling and Restraining Mobile Virus Propagation", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 12, NO. 3, MARCH 2013
[3] Sana Nseir, Nael Hirzallah, Musbah Aqel, "Issues with Various Security Threats on Mobile Phones", 2013 Palestinian International Conference on Information and Communication Technology
[4] Yury Zhauniarovich, Giovanni Russello, Mauro Conti,Bruno Crispo, and Earlence Fernandes, "MOSES: Supporting and Enforcing Security Profiles on Smartphones", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 3, MAY-JUNE 2014
[5] Ninghui Li, Ziqing Mao, Hong Chen, "Usable Mandatory Integrity Protection for Operating Systems",2007
[6] K.J. Biba,"Integrity Consideration for Secure Computer System," Technical Report TR-3153, Mitre Corp.,
[7] D.D.Clark and D.R. Wilson, "A Comparison of Commercial and Military Computer Security Policies,"Proc. IEEE Symp. Security and Privacy
[8] T. Fraser, "LOMAC: MAC You Can Live With," Proc. Usenix Ann.Technical Conf., 2001.
[9] Normi Sham Awang Abu Bakar, Iqram Mahmud, "Empirical Analysis of Android Apps Permissions", 2013 International Conference on Advanced Computer Science Applications and Technologies
[10] Naser Peiravian and Xingquan Zhu, "Machine Learning for Android Malware Detection Using Permission and API Calls", 2013 IEEE 25th International Conference on Tools with Artificial Intelligence
[11] Abdullahi Arabo and Bernardi Pranggono, "Mobile Malware and Smart Device Security: Trends, Challenges and Solutions", 2013 19th International Conference on Control Systems and Computer Science
[12] Ming-Yang Su and Wen-Chuan Chang, "Permission-based Malware Detection Mechanisms for Smart Phones", ICOIN 2014
[13] E. Hammer-Lahav, D. Recordon, and D. Hardt, "The OAuth 2.0 Authorization Protocol," Apr. 2011, IETF draft (work in progress), draft-ietf-oauth-v2-15.txt.