IJCTA, 9(33), 2016, pp. 205-210
© International Science Press

Integrating Taintdroid with the Cloud In Order toEnhance Privacy

A. Deepak Raj* Varsha K.S* Theertha A.P* and Aswathy Asok*

Abstract : Today, with the advancement of technology and other services, securing of private data in Android devices has become a persistent and complex task. Third party applications within mobile devices can access private data and send it to advertisement servers as well as various other organizations to yield profit. Since Android uses a permissions based model [6], in which apps have to request for permissions to access certain resources, apps will be able to access to private data only if the required permissions have been granted by the user. But once permissions have been granted, the user is unaware about how much of data is being sent out and its destinations. Apps intentionally steal user's private data [16] like IMEI, Contact details, Personal images etc and hand it over to various agencies without the user's knowledge. To overcome this issue, a dynamic taint tracking system, Taintdroid [8] has been developed by Enck er all. It tracks the flow of private data through applications and issues notification to the user when it goes out through the network interface. This notification message contains the name of the application that sends data to network, its destination IP address, taint (IMEI, location) and short description of data. But this message alone is not sufficient for an average user to take decision regarding the safe usage of the app. Since, there are hundreds of notifications coming from various applications, an average user finds it difficult to judge whether an app is compromising on privacy. Hence, we propose a modification to Taintdroid architecture, which involves sending the notifications from thousands of mobile devices to a cloud server for analysis. This analysis will yield a wealth of information which can be communicated to the end user, empowering them to take decisions regarding the safe usage of the app.

Keywords : Mobile Cloud Computing, Privacy, Google Cloud Messaging, Taintdroid.

1. INTRODUCTION

Android devices namely, Smart phones and Tablets have become an integral part of our daily life. Modern Smart phones provide centralized services for downloading and installing third-party applications. Usage of such applications [14] are causing serious security concerns these days. If a user wants to install an application, he has to accept certain permissions [6] requested by the app, in order to access its private data. But unfortunately, there is no way for the user to verify the amount and type of data accessed by the apps and its actual destinations. So the private data in the android devices are not secured [8]. Therefore, additional tools are required to help the user to set the optimal permissions for each application. In the latest version of android OS (Marshmallow) the user don't have to explicitly accept all the permissions instead there is provision to individually grant or revoke each requested permission.

Our research focuses on developing a solution were private data leaks from Android devices are minimized by modifying Taintdroid, a dynamic taint tracking system [7][8]. We propose a new architecture to detect privacy leaks from Android devices using Taintdroid.

* Department of Computer Science & IT, Amrita School of Arts and Sciences, Kochi Amrita Vishwa Vidyapeetham (Amrita University), India, MCA Student,deepakraj23692@gmail.com, MCA Student, varshasidharthsv@gmail.com, MCA Student, theerthaap543@gmail.com, Asst. Professor, aswathy.is.in@gmail.com

2. RELATED WORK

[1][2] The paper introduces the working of Mobile Cloud Computing (MCC). It narrates various issues and threats in MCC like Data Ownership, Privacy, Data Security and other security issues. The paper also describes various existing framework such as host agent, network service and caching. It also conveys possible solution to security issues. It is from this journal we thought of concentrating more on the Privacy issue as it is one of the biggest challenge in mobile cloud computing environment. Since Third party application benefits more by selling private data to advertising agencies and various organizations.

[3] The paper provides the concept of cloud computing, the issues it tries to address, various research topic related to it, VCL technology used to implement cloud. Here they describe that cloud computing is build on decades of research in virtualization, computing in utility, distributed computing technology and more recently adopted networking systems, web and software services. It signifies a service-oriented architecture, minimized information technology for the end-user, has great flexibility, total cost of ownership is much minimal, on demand services and many other things.

Further focusing more on Privacy on mobile cloud computing, we came across a paper [5] which describes the issues relating to security. These issues mainly occur due to the constraint of resources in mobile devices especially in terms of storage, energy and processing. They claim that these issues can be resolved by establishing a framework for security and privacy.

[11][13] This paper represents a comprehensive survey of Mobile Cloud Computing (MCC). MCC refers to an infrastructure where both data storage and data processing happen outside of the mobile device. The paper provides a brief overview of MCC which includes its definition, architecture, and advantages. The architecture contains mobile users, base stations, central processors, servers, internet service providers (ISP), cloud controllers and data centers. The advantages of MCC are extending battery life, improving data storage capacity and processing power, and improving reliability. The paper also discusses the use of MCC in various applications like mobile commerce, mobile learning, mobile healthcare, mobile gaming and other practical applications. Beside that it shows several issues like low bandwidth, heterogeneity, computing offloading, security, enhancing the efficiency of data access. This paper also presents open issues and future research directions.

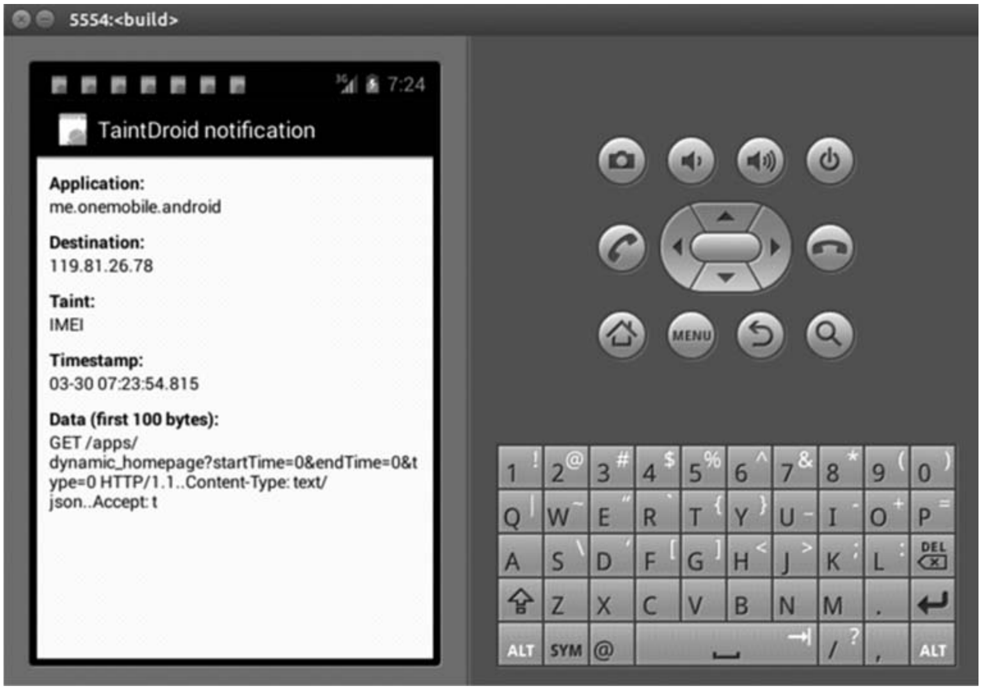
[12] This paper provides an overview of cloud computing, its constituting elements including the cloud platform and its application. A Comparison between EC2 and GAE cloud platforms based on several attributes were provided in the paper. It also discusses the challenges associated with implementing cloud computing such as the challenges regarding mobile device and network, the scarcity of resources, availability of high speed internet access, bandwidth Security issues etc associated with mobile devices along with their possible solutions.

3. STUDY OF EXISTING SYSTEM

Our research focuses on Taintdroid [8], a dynamic taint tracking system which tracks the flow of sensitive data within application. It labels or taints the data from sensitive sources, and warns the user when such tainted variables are leaked into the network. The taint tracking system performs certain tasks. Firstly, it assigns taint labels to private data. Secondly, it automatically propagates taint labels to dependent data and variables and finally, it takes some action when the tainted data reaches the taint sink.

A taint is a label given to the data. Taint assigns values such as IMEI number, Geographic Location, SIM number etc. The responsibility of Taintdroid is to assign taint labels at taint source, taint labels are propagated to dependent data and when this data reaching the taint sink or the network interface, Taintdroid issues a notification to warn the user. In Taintdroid, tracking occurs at four levels [8]: variable level, method level, message level and file level. In variable level tracking, the virtual machine assigns taint labels to variables holding private data. In message level tracking, messages sent across applications using IPC are tainted. In method level tracking, the returned values from native methods are tainted when tainted values have been passed. In file level tracking, persistent data stored within files are also tainted. This four level tracking approach is implemented to increase the performance efficiency of taint tracking.

Taintdroid was installed in our mobile emulator by following commands in [17]. After building and installing Taintdroid, as shown in Fig 2, the Taintdroid Notify Controller app got installed. Taintdroid Notify Controller app is a third party app developed by Enck et al [7]. It is used to start and stop the taint tracking functionality of Taintdroid. It also monitors the taint logs and converts them into Taintdroid notifications. As shown in Fig 1, notifications contain details about the name of the application sending out data, its destination IP, taint tag, timestamp and the actual contents of the data.



Application:
me.onemobile.android

Destination:
119.81.26.78

Taint:
IMEI

Timestamp:
03-30 07:23:54.815

Data (first 100 bytes):
GET /apps/dynamic_homepage?startTime=0&endTime=0&type=0 HTTP/1.1..Content-Type: text/json..Accept: t

Figure 1: Taintdroid Notification

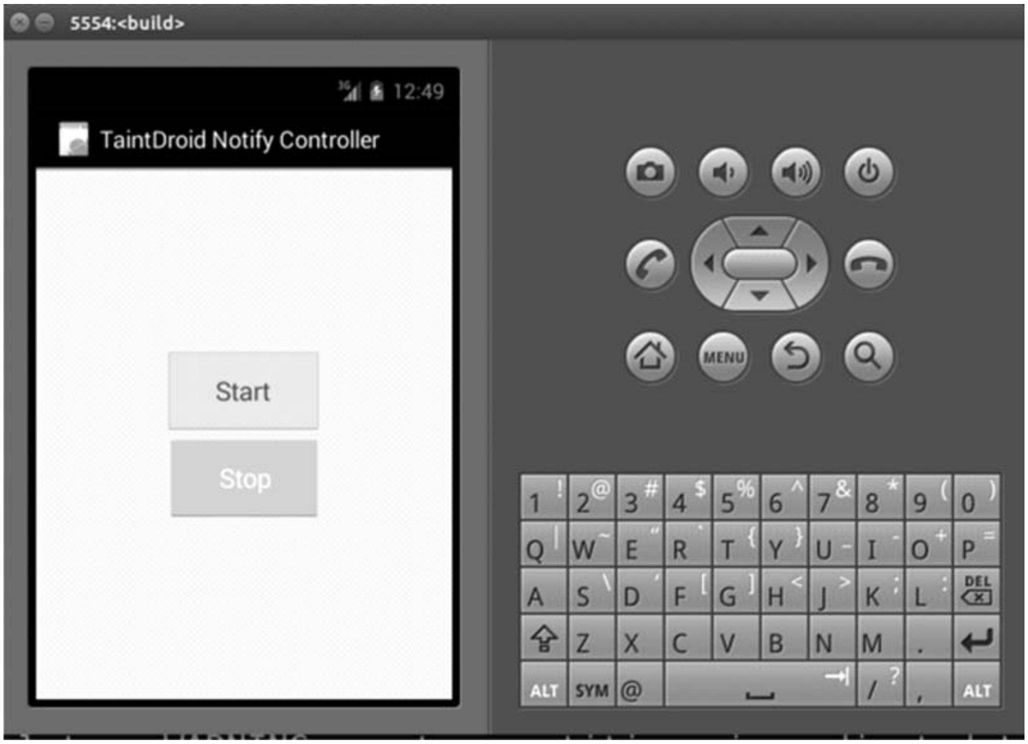


Figure 2: Taintdroid Notify Controller app

We analyzed the working of Taintdroid from a privacy point of view and came to the following conclusions. Even though Taintdroid shows user notifications, it does not serve the purpose of warning the user about privacy leaks. This is because an average user does not have the expertise required to judge whether the application is maliciously sending out private data, just by viewing the notifications. It has to be noted that notifications are produced for both legitimate and unwanted traffic. So in order to detect privacy leaks, we need to analyze the destination IP and also check whether the app has permission to access the data being send. The behavior of individual apps needs to be analyzed over a period of time before deciding whether it is safe or not. So we introduce an architecture to overcome these drawbacks of Taintdroid.

4. PROPOSED SYSTEM

It is clear from the above study that Taintdroid needs to be modified in order to perform some analysis on the outgoing data, in order to give appropriate warnings to the user about privacy leaks. Since a mobile phone does not have the memory or processing power to store and analyse this kind of information from all the installed apps, we propose the use of a cloud architecture [15] to perform the analysis. We need to integrate a Taintdroid Cloud Server whose responsibility is to store and analyze the details about the data being sent out by the various apps. As shown in Fig 3, an added advantage is that it can be used for the analysis of data from multiple mobiles running similar apps. While researching on suitable ways to integrate a Taintdroid client into the existing architecture, we decided that the Taintdroid Notify Controller app was a suitable one, mainly because it already has access to the taint logs. So all that is required to be done, is to add the functionality to connect to the Taintdroid server and to periodically send the taint logs to the server for analysis. The server can send back the results of the analysis and this will be made available to the user. The server can be programmed to do various types of analysis, for example, it can display a summary of the type of data accessed by these apps and the frequency at which it is being sent out, it can inspect the destination IP to detect malicious behavior and also give appropriate warnings to users about the second order privacy risks involved when the wrong combination of permissions are selected. The architecture of the proposed system is :

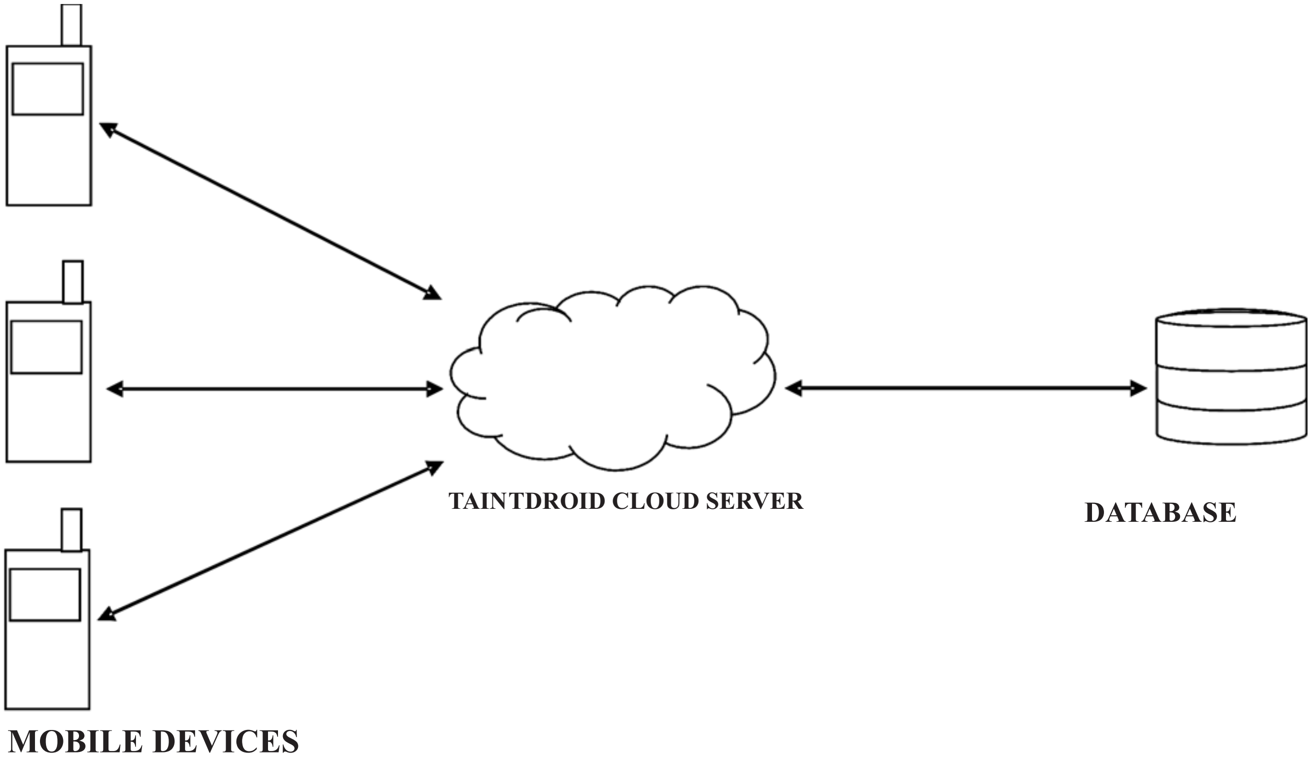


Figure 3: Taintdroid Cloud Server Architecture

4.1. Working of Taintdroid Cloud Server

Initially the mobile device needs to register with Taintdroid Cloud Server. After successful registration, Taintdroid Cloud Server starts communication with the Taintdroid Notify Controller app which acts as the Taintdroid Client. When tainted data goes out from the device through the network, Taintdroid tracks the tainted data and shows notifications. After displaying the notification, Taintdroid client sends application name, destination IP address, description and name of tainted data shown in notification message to Taintdroid cloud server and stores it in the cloud server database for further analysis. The analysis of data will enable us to perform a number of different tasks. For example, Taintdroid cloud server verifies whether the destination IP address shown in notification message is valid or malicious, checks whether tainted data in the notification message has been accessed with valid application permission or not, it can even rank applications on the basis of consistent behavior analysis. An analysis of second order privacy leaks can also be done with the available data. Summarization of the application behavior over a period of time can be made available to the user. On the basis of this analysis, warning messages are generated and send from the server to the various clients. From these warning messages, users can understand that applications are trying to gain access to private data and misuse it, which is leading to loss of privacy. So user can either uninstall the application from the mobile or change the permission settings and restore their privacy.

4.2. Detailed Description

The Taintdroid Notify app has to be modified to communicate with Taintdroid server using Google Cloud Messaging (GCM) [10]. GCM requires two servers: GCM server and App Engine server. GCM servers are provided by Google. GCM server receives alert messages from App Engine server and sends it to the mobile application. App Engine server is a third party application server. App Engine server sends alert messages to the mobile application via GCM server [4].

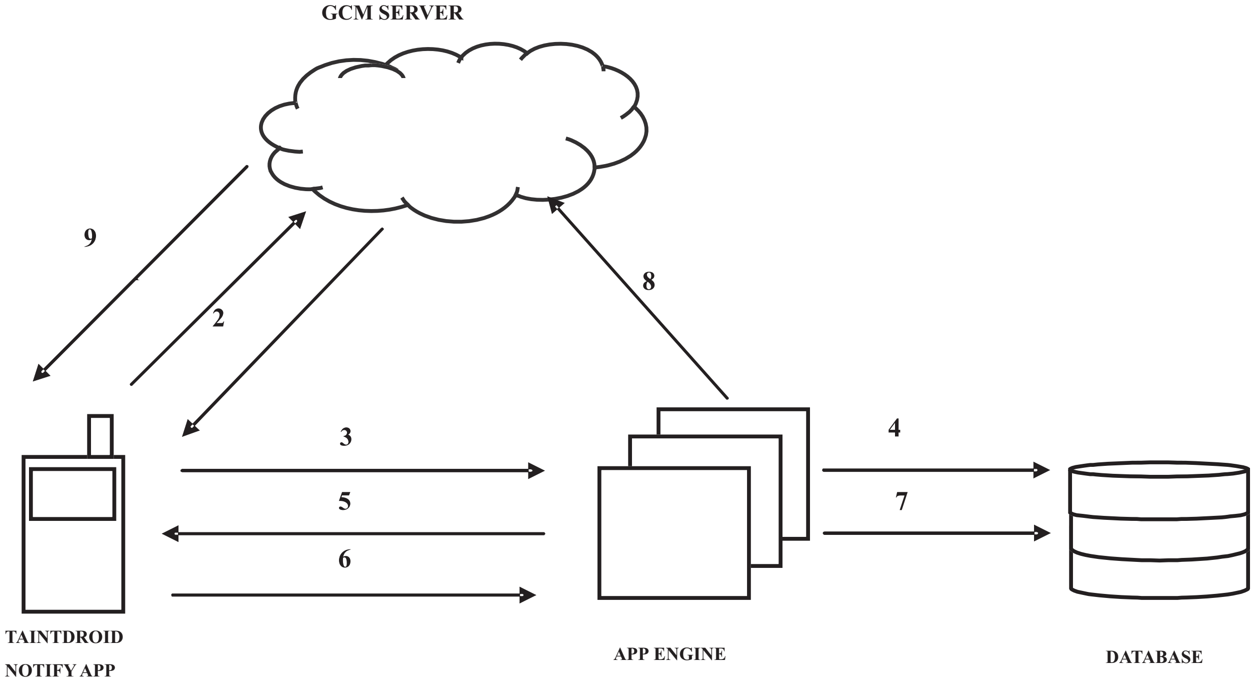


Fig. 4. Detailed Overview of Communication between Taintdroid Servers and Clients.

- Firstly, Taintdroid Notify App sends sender id, application id to GCM server for registration.
- After successful registration, GCM server sends registration token to Taintdroid Notify App.
- After receiving registration token, app will send token to our app engine server.
- App Engine stores registration token in the database for future usage.

- App engine server issues an acknowledgement message to Taintdroid Notify App.
- Taintdroid Notify app sends application name, destination IP address, description and name of tainted data shown in notification message to App Engine.
- The App Engine store received data in the database and performs various analysis operations.
- App engine server sends alert message to GCM server along with device registration token.
- GCM server delivers that alert message to Taintdroid Notify App using registration token.

5. CONCLUSION AND FUTURE WORK

In this paper, we discussed about the drawbacks of Taintdroid from a privacy point of view. Taintdroid displays notifications when tainted data leaves from the mobile devices. But this notification message is not sufficient for a user to analyze the behavior of applications installed in the device. To overcome the drawbacks of existing system, we introduced a new system which helps the user to understand whether the installed applications are malicious or not. Instead of notification message, Taintdroid cloud server will provide an alert message to the user after thorough analysis of the data. With the help of alert messages, user can either uninstall or can change the application permission settings and hence ensure privacy.

6. REFERENCES

1. Abhimanyu, M. A., Chetan, S., Dinesh, K., Kumar, G., & Mathew, K. (2010). Cloud computing for mobile world.
2. A. C., Donald, Arockiam, L., & S. A. Oli. (2013). Mobile cloud security issues and challenges: A perspective. International Journal of Electronics and Information Technology (IJEIT), ISSN, 2277-3754.
3. A Vouk, M. (2008). Cloud computing–issues, research and implementations.CIT. Journal of Computing and Information Technology, 16(4), 235-246.
4. Chan, N. W. H., Melton, R. W., Yang, S. J., & Zhao, L. (2015, August). Privacy Sensitive Resource Access Monitoring for Android Systems. InComputer Communication and Networks (ICCCN), 2015 24th International Conference on (pp. 1-6). IEEE.
5. Chaturvedi, M. M., & Malik, S. (2013). Privacy and Security in Mobile Cloud Computing: Review. International Journal of Computer Applications, 80(11).
6. Chin, E., Felt, A. P., Hanna, S., Song, D., & Wagner, D. (2011, October). Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (pp. 627-638). ACM.
7. Chun, B., Cox, L., Enck, W., Gilbert, P., Jung, J., McDaniel, P., & Sheth, A. Taintdroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones.OSDI'10. Proceedings of the 9th USENIX conference on Operating systems design and implementation, 2010.
8. Chun, B. G., Cox, L. P., Enck, W., Gilbert, P., Han, S., Sheth, A. N., & Tendulkar, V. (2014). TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS), 32(2), 5.
9. Chun, B. G., Ihm, S., Maniatis, P., Naik, M., & Patti, A. (2011, April). Clonecloud: elastic execution between mobile device and cloud. InProceedings of the sixth conference on Computer systems (pp. 301-314). ACM.
10. Developers, A. (2014). Google cloud messaging for Android. 2014-02-03]. http://developer. android. com/google/gcm/index.html.
11. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. Wireless communications and mobile computing, 13(18), 1587-1611.
12. Dubey, V., Sahu, D., Sharma, S., & Tripathi, A. (2012).Cloud computing in mobile applications. International Journal of Scientific and Research Publications, 2(8), 1-9.
13. Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29(1), 84-106.
14. Freeh, V. W., Jiang, X., Zhang, X., & Zhou, Y. (2011). Taming information-stealing smartphone applications (on android). In Trust and Trustworthy Computing (pp. 93-107). Springer Berlin Heidelberg.
15. Jadeja, Y., & Modi, K. (2012, March). Cloud computing-concepts, architecture and challenges. In Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on (pp. 877-880). IEEE.
16. Jiang, X., & Zhou, Y. (2012, May). Dissecting android malware: Characterization and evolution. In Security and Privacy (SP), 2012 IEEE Symposium on (pp. 95-109). IEEE.
17. Taintdriod. http://www.appanalysis.org.