Mob-AIDS: An Intrusion Detection System for the Android Mobile Enterprise

Oyeleye Christopher A. (Ph.D)[1], Daramola Comfort Y.[2], Akinpelu James A. (Engr.)[3]

[1] Department of Computer Science & Engineering
Ladoke Akintola University of Technology, Ogbomoso
Oyo State, Nigeria

[2] Department of Computer Science
Federal University, Oye-Ekiti
Ekiti State, Nigeria

[3] Department of Computer Science & Information Technology
Bowen University, Iwo
Osun State, Nigeria

Abstract

Researches in the development of efficient security models, policies and standards for the Android platform has increased significantly in recent times. Android suffers from the widespread security threats and malware due to the lack of efficient security tools for protection. However, recent studies on android mobile platform security stressed that Mobile Botnets, Global System for Mobile Communication (GSM)-based Pivot Attacks and Malicious Applications are the major security challenges of the mobile operating environment. In this paper, a robust and efficient Android Intrusion detection system middleware Application is developed to address the major security vulnerabilities infiltrating the android Mobile platform. The application is developed using the Java 2 Mobile Enterprise (J2ME) platform. The proposed Android Intrusion Detection System (Mob-AIDS) was evaluated based on users' assessment to determine its efficiency in terms of graphical user interface, Mobile resource conservation, reliability and performance. Questionnaire and oral interview are tools used to gather relevant data to evaluate the performance of the system. Results of the analysis of the respondents' data revealed that Mob-AIDS has sufficient capability to prevent unauthorized or anomalous access into the android Mobile enterprise and help realize a more secured and reliable operating environment.

keywords—Android; IDPS; Mobile Security; Mob-AIDS.

1. INTRODUCTION

Mobile devices face a wide range of new security challenges including malicious threats and intrusion (Jon, 2010). These devices are increasingly being used to store sensitive personal information such as financial data used for Mobile banking, but also run applications that pose potential abuse for snooping on a Mobile user's voice, SMS, data and location services (Jon,Veeraraghavan, Cooke, Flinn and Jahanian, 2008). In an attempt to manage these security issues, firewalls and mobile antivirus softwares were developed.

Although these security mechanisms can protect against malicious software or services, such applications are only provided as a third-party and not as part of the operating system (OS). This makes the majority of the devices remains still unprotected. Furthermore, none of these security mechanisms are able to detect a new intrusion of threats, but they protect only against already known malware (Dagon and Stamer, 2004).

The android platform, an open-source Mobile operating system, suffers from major security vulnerabilities especially GSM based Pivot Attacks, Mobile botnets, Malicious Applications, Infection via Personal Computers, Device to device Infection and Infection via Rogue Wireless Networks (AISEC, 2012; Nohl & Melette, 2011).

In this paper, Mob-AIDS, a robust and efficient android intrusion detection system, is developed to address these security vulnerabilities so as to realize a more secured and reliable android operating environment.

2. LITERATURE REVIEW

This section introduces a brief literature on mobile device operating system for securing the android platform.

Mobile Device Operating System and Security Issues

Mobile environments differ significantly from traditional fixed computing environments. While some differences are straightforward, others may have subtle consequences that can have a significant impact on the security of a Mobile device. Most mobile device OS have two user profiles; the superuser and the user (Sven and Stephan, 2013). The first one is a special user account used for system administration in order to run low level system processes, to be able to view or make changes to the contents of the file system or change the permissions to specific files, processes or users. The second profile is the typical user profile having high level permission in order to read and execute specific commands and within the application sandbox.

In modern mobile device OSs, owner of a mobile device does not have the access to superuser privileges. In most cases, all encrypted user data are decrypted once the user is correctly authenticated by the OS using their PIN. Moreover, software exploits are used by hackers to modify the OS successfully to gain superuser privileges to system areas, bypass the sandoxed application or provide low level OS kernel modifications successfully, which were by default protected by each OS manufacturer. Such low-level modifications can have a variety of names dependi

OS they are applied to. They are either known as Rooting (Android, Windows Mobile), Jailbreak (iOS) or Capability Hack (Symbian) (Ekberg and Kyl, 2007).

Android antivirus software has to resort to two primary sources of information for malware detection (ADG, 2012):
1. Package database
2. Package files (APK files) of installed apps

The package database stores package names and the package file locations. However, Android antivirus software cannot list the contents of the directory with installed packages; it rather relies on the package database to provide it with the locations of installed packages. These package files can then be read to be checked using typical antivirus detection techniques. All files added after installation, however, remain invisible to antivirus software on the Android platform.

These indicate that retrospective detection rate tests of Android antivirus software do not reflect the real protection level offered by such antivirus software. The only threat it can protect against is known and not very advanced malware, using package names and package files of installed packages. Any activity after installation cannot be controlled or detected by Android antivirus software.

Security Challenges in Android Mobile Network

The major security vulnerabilities of the android enterprise addressed in this paper include Mobile botnets, GSM-based pivot attack and malicious applications.

A. Mobile Botnets: Compared to traditional fixed computing, the case for mass ownage of Mobile devices for creating a botnet is not as straightforward. Traditionally, attackers are able to monetize their botnets of compromised hosts through spam, denial of service extortion, sensitive data theft and phishing of confidential details (Dagon and Starner, 2004). Compromised hosts are often considered valuable to an attacker if they have high-throughput, low-latency, stable connectivity to the Internet and significant system resources; which are attributes that are not common with today's Mobile devices (Ekberg & Kyl, 2007). However, as more and more sensitive data such as login credentials are stored on Mobile devices, attackers may still wish to target them for harvesting data.

B. GSM based Pivot Attacks: The GSM implementation by many base transceiver stations is prone to various easily conductible attacks. Recent results presented at the 28th Chaos Communication Congress have demonstrated that most European GSM networks are not capable of prohibiting impersonation attacks (Brunner, Hofinger, Krauss, Roblee, Schoo and Todt, 2010). This occurs when an attacker fakes the identity of another GSM subscriber, thus receiving any communication addressed to the attacked person (Brunner et al, 2010). Hence, just one Mobile device under an attacker's control in a radio cell is sufficient to attack any other subscriber and to serve as a remote long range wiretap. This matter, though theoretically feasible, is of very high difficulty and no further research into it has been conducted yet (Brunner et al, 2010).

C. Malicious Applications: Malicious applications are software used or created to disrupt computer operation, gather sensitive information or gain access to private computer systems and Mobile devices (Nohl & Melette, 2011). Incidents have many causes, such as malware (worms & spyware), attackers gaining unauthorized access to systems from the Internet and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization (Bace & Rebecca, 2000). Some Intrusion Prevention System (IPS) technologies can remove or replace malicious portions of an attack to make it benign.

3. REVIEW OF RELATED WORKS

Sven and Stephan (2013) presented the design and implementation of FlaskDroid, a policy-driven generic two-layer MAC framework on Android-based platforms. The applicability of the design was prototyped on Android 4.0.4. Evaluation of the system shows that the clear API-oriented design of Android benefits the effective and efficient implementation of a generic mandatory access control framework like FlaskDroid.

Rafael, Julian and Marcel (2013) evaluated how well Android antivirus software performs under real world conditions, as opposed to retrospective detection rate tests. The authors conducted various tests on several antivirus apps for Android. The test setup considers the ability to cope with typical malware distribution channels, infection routines and privilege escalation techniques. It was concluded that it is easy for malware to evade detection by most antivirus apps with only trivial alterations to their package files.

Jon et al (2008) argued that while security vendors have marketed Mobile-specific versions of antivirus software to detect malware, these solutions are similar to their desktop variants and provide limited detection capability with significant power and resource overhead. Even with simple signature-based static analysis, the computational resources required to perform such analysis can be high. For example, the Clam AV antivirus engine available for the Nokia N800 Mobile device requires 57 seconds of processing just to initialize its signature database and consumes as much as 40 megabytes of memory.

Kruegel & Chris (2004) developed a malware detector that can identify reconnaissance activity in Android, which may indicate that an attack is imminent. The IDPS is able to block reconnaissance and notify security administrators, who can take actions if needed to alter other security controls to prevent related incidents. Because reconnaissance activity is so frequent on the Internet, reconnaissance detection is often performed primarily on protected internal networks.

Based on the existing research works on Mobile IDPS, little or nothing has been done on the development of IDS to mitigate the vulnerabilities of the Android platform which is the focus of this paper. The android vulnerabilities

addressed in this paper are GSM based Pivot Attacks, Mobile botnets and Malicious Applications.

4. MATERIALS AND METHOD

The detailed methodology and design approach adopted are described as follow:

a. The Android IDS Application Design Framework

The operational flow diagram of the android IDS application developed, which defines its mode of operation, is presented with tethering option selected in figure 1.

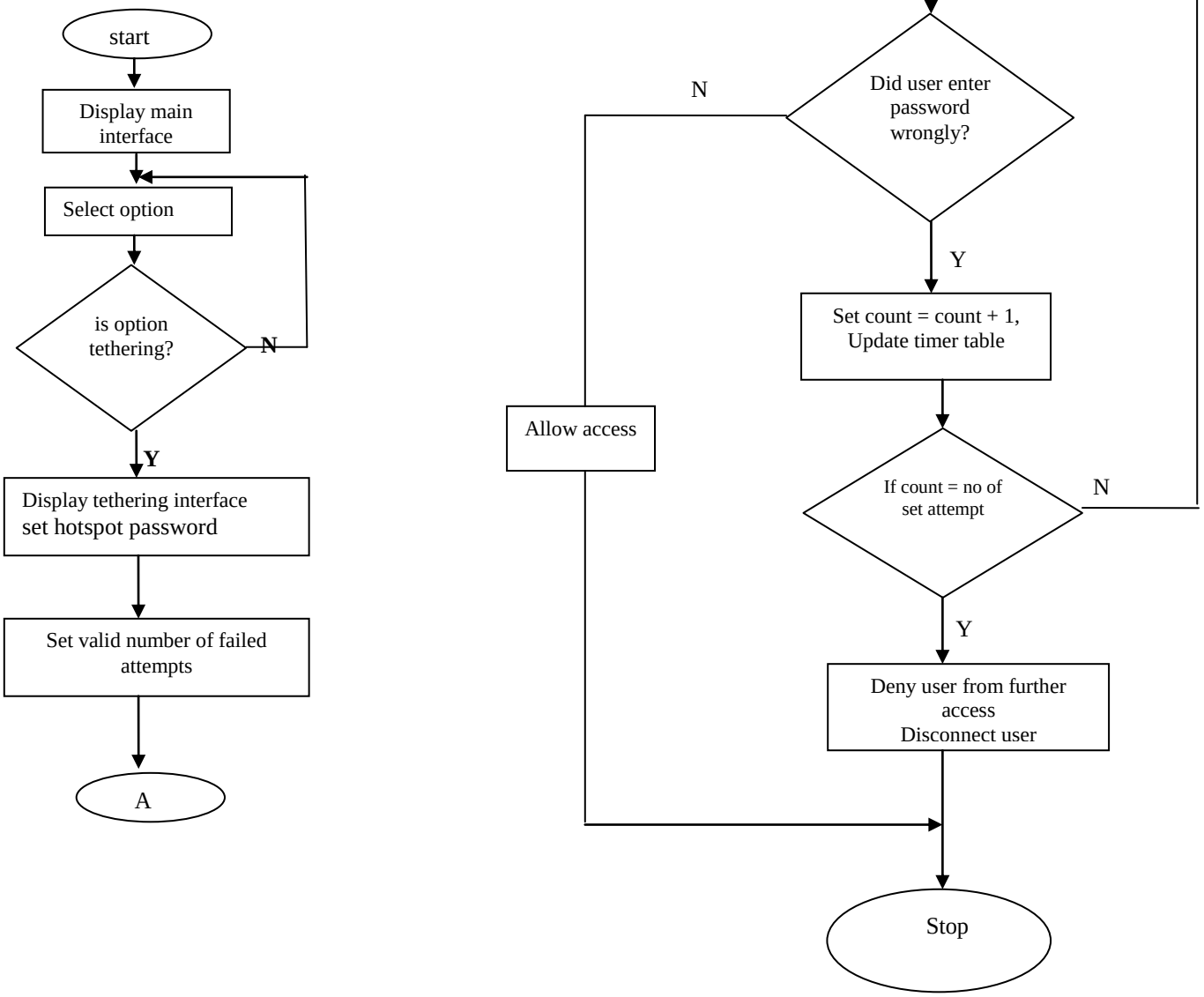

Fig. 1. The Operational Flow Diagram for Mob-AIDS Tethering Option

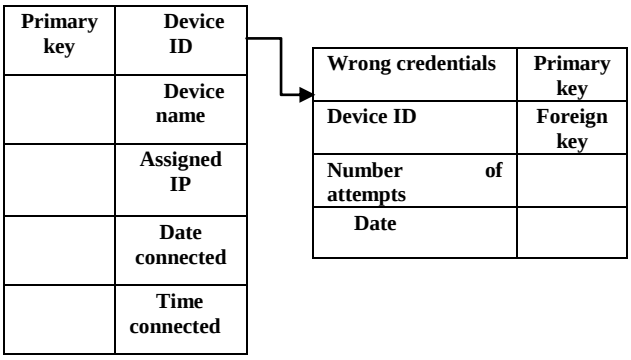

Fig. 2. The entity relationship between the Mob-AIDS tables

b. The Android IDS Application Database Design

The ultimate objective of database analysis and design is to establish an electronic data store, that is, a physical model of the relevant aspects of a user's conceptual world. Many factors considered during this process include, but not limited to, historical and future data perspectives, the diversity of the user community, organizational requirements, security, cost, ownership, performance, temporality, user interface issues and data integrity. These factors contribute to the success of a database application in either a quantitative or qualitative fashion, and in turn contribute to the overall quality of the database.

Mob-AIDS_Login_Attempt, Mob-IDS_Connected_Devices and Mob-AIDS_Wrong_Credentials_Log tables make up the Android IDS Backend as depicted in Tables 1, 2 & 3 respectively. The Mob-AIDS_Login_Attempt table holds data regarding the login attempt of the Mobile user, basically the identification number of each attempt and the number of allowed attempts. Mob-AIDS_Connected_Devices table holds information regarding the device name, the internet protocol (IP) address and the identification number of the connected device as well as the date and time when the device was connected. However, Mob-AIDS_Wrong_Credentials_Log table logs data regarding wrong credentials supplied by the android Mobile users.

Mob-AIDS_Login_Attempt Table Structure

Table 1: showing the attempt table design structure

Attempt ID	No of allowed attempts
Primary key	

Mob-AIDS_Connected_Devices Table Structure

Table 2: showing connected device data table design structure

Device ID	Device name	Assigned IP	Date connected	Time connected
Primary key				

Mob-AIDS_Wrong_Credentials_Log Table Structure

Table 3: showing wrong credentials table design structure

Wrong credentials	Device ID	Number of attempts	Date
Primary key	Foreign key		

Mob-AIDS_Connected_Device Table and Mob-AIDS_Wrong_Credentials_Log Table are linked or related by their common field (Device ID). Creating this one-to-many relationship established through the entity-relationship model provides an automatic layer of quality assurance to the database. This is depicted in figure 2.

5. IMPLEMENTATION AND DISCUSSION

This section presents and discusses the implementation of the Mob-AIDS developed:

The Android IDS Interface Design Implementation

Mob-AIDS is a Mobile application, developed using Java 2 Mobile Edition (J2ME) platform, for android Mobile devices. After the application has been deployed, a generalized password is set by the host or administrator as the main authentication method. The graphical user interface was designed with simplicity and easy to understand terms which will aid the user on correct usage of the application as every form of ambiguity is eliminated.

The startup interface comes up once the application is launched. This application runs on any version of the Android OS. The start-up page presents features for wireless connection management, timer policy, tethering management and other preferred policies based on the users' needs. The startup page of the Mob-AIDS is presented in figure 3.

Fig. 3. Mob-AIDS Start-up Page

The wireless manager manages the initiation and termination of wireless connections to the android device. It presents features to enable and disable wireless connectivity which intermittently the administrator can use to start or stop a wireless connection. This is presented in figures 4a and 4b below.

Fig. 4a. Mob-AIDS Wireless Manager

Fig. 4b. Mob-AIDS Wireless Startup

The wireless manager icon can be tapped to start or stop a wireless. The timer policy presents a policy mechanism that enforces the administrator or host to set the maximum limit

to password entry attempts. This will help to prevent unauthorized users access into the network or resource center from unlimited or endless login trials. Any user who reached the maximum set of failed attempts by the administrator will be logged out of the system and the details also documented down by the system.

Fig. 5. Mob-AIDS Timer Policy

The tethering management support feature allows the users to set and view the network, available connections and the user authentication mechanisms for devices attempting connections or already connected. Figure 6 below presents the tethering enabling interface.

Fig. 6. Mob-AIDS Tethering enabling interface

Tethering is a vital process of resource sharing mechanism which makes users to view and connect to available networks. Tethering is started by tapping on the "start tethering" button to enable users to connect to the host.

Fig. 7. MOB-AIDS Tethering enabled Features

Figure 8 shows the tethering enabled features. This page comes up once the tethering feature is activated. It provides various features for managing the connection of Mobile devices within range. With the "Tethering & portable hotspot" feature selected, "USB tethering", "Portable WLAN hotspot" and the "Portable WLAN hotspot setting" features become enabled.

With these features enabled, the external users can connect for resource sharing and other purposes.

Fig. 8. WLAN hotspot configuration page

Figure 8 shows the interface that allows the administrator to configure the WLAN hotspot basic settings. The password and the network ID to identify the network are set. "Fagbola's Mobile" as shown in the interface is the sample network name / ID. Password is also set for authentication purpose.

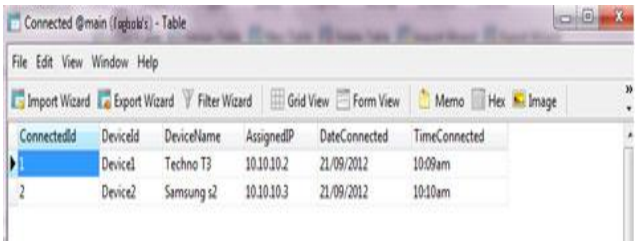

Fig. 9. Sample Data for Mob-AIDS_Connected_Devices Table

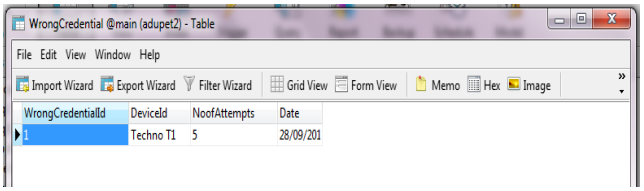

Fig. 10. Sample Data for Mob-AIDS_Wrong_Credentials Table

Figure 9 presents the record of all connected devices. Devices that make attempts more than the set policy can be traced for proper documentation and can be later traced. Arrest can be made for trying to intrude into the system if discovered that the attempts was from an unauthorized user. Logged Information about the devices is also maintained for auditing and further research purposes and can be viewed when needed. However, Figure 10 shows the record of all connection attempts made by each device.

Evaluation of the Developed Android IDS

The population used for evaluation comprises of some Android Mobile users in Oyo State, Nigeria. The users consist of students, Information Technology (IT) stakeholders, lecturers, teachers and artisans. Consequently, there was adoption of a purposive technique to determine those to be interviewed (sample size) because the population in the study area is large. Purposive sample is drawn to aid the ease of data collection or special features of the members of the sample. Therefore, the selection of the respondents was based on identification made by the researchers in the study area on those who can potentially serve the research purpose. A total of one hundred (100) copies of questionnaire were distributed to these respondents from diverse educational background while ninety-five (95) copies were returned, representing a response rate of 95% as follows:

i. IT Stakeholders = 22
ii. Students = 32
iii. Artisans = 9
iv. Lecturers = 11
v. Teachers = 21

The respondents were asked to comment on the efficiency of the proposed Android IDS in terms of graphical user interface, Mobile resource conservation, reliability and performance.

Data Collection Instrument

A well-structured questionnaire and oral interviews were used to gather primary data for the study. The questionnaire was validated and tested for reliability. A Cronbach alpha reliability co-efficient of $\alpha = 0.72$ was achieved.

Method and Tools for Data Analysis

Microsoft Excel was used to capture and analyze the data obtained from the duly-filled copies of questionnaire while percentage distribution was the descriptive technique used. The descriptive survey was adopted to obtain the opinion of a representative sample of the target population so as to be able to infer the perception of the entire population.

Result of the Evaluation of the Proposed Android IDS Application

The efficiency of the proposed Android IDS was evaluated by the users using performance, resource management, reliability and graphical user interface as evaluation metrics.

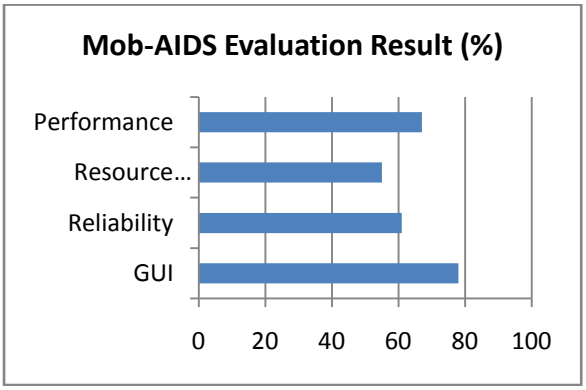

Fig. 11. Result of the evaluation of the Android IDS Application

The result of the analysis of the data extracted from the questionnaires is summarized in figure 11. The graphical user interface scored 78 %, the reliability scored 61%, the resource management scored 55% and the performance scored 67%.

6. CONCLUSION

Android platform, an open-source operating system, is susceptible to major vulnerabilities such as GSM based Pivot Attacks, Mobile Botnets and Malicious Applications. These challenges necessitate the development of a security support application for the Android enterprise. However, in this paper, an Android IDS Application (Mob-AIDS) is developed to address these vulnerabilities.

Mob-AIDS was tested on Android 2.2 and later versions and found to help detect and prevent intrusion based on set policies. With Mob-AIDS, a high level of network security integrity is guaranteed. Hackers, intruders, perpetrators and all other unauthorized users can be apprehended with use of the logged details about them. The technical approach adopted has been well illustrated and the Mob-AIDS system is expected to bridge the security vulnerability gaps of the current android platform. However, further research work can extend the deployment of the Mob-AIDS to a service-oriented application.

REFERENCES

[1] Android Developer's Guide (ADG), https://developer.android.com/guide/topics/manifest/permissionelement.html, 2012.

[2] Bace and Rebecca "*Intrusion Detection*," Macmillan Technical Publishing, 2000.

[3] T. Brunner, H. Hofinger, C. Krauss, C. Roblee, P. Schoo and S. Todt, "Infiltrating Critical Infrastructure with Next Generation Attacks; Stuxnet as a Showcase Threat", 2010.

[4] T. Dagon Martin and T. Starner, "Mobile phones as computing devices: The viruses are coming", IEEE Pervasive Computing". 2004.

[4] K. Ekberg and M. Kyl "Mobile Trusted Module (MTM), An Introduction" Nokia Research, 2007.

[5] AISEC "Android OS Security: Risks and Limitations", AISEC 2012.

[6] L. Garfinkel and M. Rosenblum "When virtual is harder than real: Security challenges in virtual machine based computing environments. In 10th Workshop on Hot Topics in Operating Systems", 2005

[7] Jon Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian "Virtualized In-Cloud Security Services for Mobile Devices in Workshop on Virtualization in Mobile Computing (MobiVirt '08)", Breckenridge, Colorado, 2008.

[8] Jon Oberheide "remote kill and install on google android,". http://jon.oberheide.org/blog/2010/06/25/remotekillandinstallongoogleandroid/, 2010.

[9] Karen Scarfone "Guide to Intrusion Detection and Prevention Systems (IDPS), 2009"

[10] K. Nohl and Melette L. "Defending Mobile phones," in *28th Chaos Communication Congress*, http://events.ccc.de/congress/2011/Fahrplan/attachments/1994_111217.SRLabs28C3Defending, 2011.

[11] Rafael Fedler, Julian Schütte, Marcel Kulicke , "On the Effectiveness of Malware Protection on Android, An evaluation of Android antivirus apps", Applied and Integrated Security (2013).

[12] SANS Institute "Intrusion Prevention Systems- Security",2002.

[13] Sven Bugiel and Stephan Heuser, "Flexible and Fine-Grained Mandatory Access Control on Android for Diverse Security and Privacy Policies", The Internet Society (2013).

[14] L. Wenjia and J. Anupam, "Security Issues in Mobile Ad Hoc Networks- A Survey" 2003.