IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 3, June-July, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

A Cloud Based Security System For Cellular Phones

Ehab M. Alkhafajy[1], Taha M. Mohamed[2], Mahmoud M. El-Khouly[3]

[1,2,3]Information Technology Department, Faculty of Computers and Information,
Helwan University, Egypt

Abstract

Smartphones have been widely used in recent years. They offer almost the same functionality as personal computers. Therefore, they are vulnerable to similar type of security risks of PCs. The Android-based mobile devices had appeared recently becoming an ideal target for attackers. The users of Android-based smartphones can download free applications from Android Market. These applications may contain malware that violates user privacy. In this paper, we propose a system for providing security services for smartphones. The proposed system works on a cloud environment. The proposed system can detect malware on android based devices. The system combines both signature-based and behavior-based techniques. The experimental results show that, the proposed system improves detection accuracy and scanning time for malware detection. It achieves higher detection rate up to 37% compared to a single antivirus engine. Moreover, it reduces false positive rate by 3.7%. The false negative rate is also reduced by 3%. The battery consumption is reduced by 7%. The CPU activity is also reduced by 23%.

Keywords: *Cloud computing, malware detection, mobile security, smartphones, Android OS.*

1. Introduction

Over the last decade, the popularity of mobile devices such as smartphones, have increased tremendously. According to Gartner [1], smartphone market is expected to grow about ten times from 9.8 million units in 2012 to 96.3 million units in 2017.Vendors sold 472 million smartphones worldwide in 2011. Estimations are talking about 982 million smartphones being sold in 2015 [2]. This tremendous growth in the numbers of smartphones is due to these phones are more capable when compared to traditional mobile phones. They offer many features such as the ability to run software applications, check e-mail, browse the internet, watch videos, play music and much more. In addition to these functions, they are also used for secured tasks such as online banking. Smartphones functions are no longer limited to simply browsing menus or dialing phone numbers. Smartphones also have a larger screen, increased storage and higher computational capacity compared to traditional mobile phones [3].

This large spreading of smartphones raises the need for security. The android based systems have larger popularity. The Android based systems are easier to be attacked due to its open source nature. So, its security issues are of major concern. The resource constraints of the smart phones such as battery, CPU, and memory are limiting factors in developing powerful security services.

During the last few years, a new kind of distributed computing raised. This is the Cloud Computing**.** Cloud Computing has many informal definitions. One of them that proposed by the National Institute of Standards and Technology (NIST) which defines the cloud computing as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [4].

Cloud Computing offers different service models, that allow customers to choose the appropriate service model that fits their environment needs. The cloud service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [5][6].

In SaaS, the consumer uses the provided applications which hosted in the cloud. In PaaS, the consumers deploy their own applications into the cloud infrastructure. Programming languages and application development tools used must be supported by the provider. For example, Google Apps. In IaaS, the consumers are able to monitor storage, network, processing, and other resources. They can also deploy and operate arbitrary software ranging from applications to operating systems.

Moving computationally intensive security services to the cloud could be extremely beneficial for smartphones users. The simplicity and scalability of cloud computing attract users and organizations. The proposed system concentrates on scanning mobile phones from viruses on the cloud side rather than the device side. The proposed system tries to benefits from the huge capabilities of the cloud and overcomes the limitations of smart phones. The proposed system targets android based systems.

This paper is organized as follows. In section 2, we introduce the previous work on mobile security. In section 3, we describe the design and implementation of the proposed system. In section 4, we show the experimental results. The paper is concluded in section 5.

2. Previous Work.

Chris et al. [7], introduced a system called ThinAV. It is an anti-malware system for Android that uses pre-existing web-based file scanning services for malware detection. ThinAV aims to assess the feasibility of providing real-time anti-malware scanning over a wide area network. ThinAV cannot be installed on any android based device. The running time is also large. Shabtai et al.[8], present a framework called Andromaly for Android smartphones, which realizes a Host-based Intrusion Detection System (HIDS). The detection system runs directly on the device and monitors various features and events on the smartphone and classifies them as benign or malicious. They evaluate their framework by testing game and tool applications, where the classification algorithm is able to distinguish between those two kinds of applications. The authors evaluate several combinations of classification algorithms and feature selections and conclude that the proposed anomaly detection is feasible on Android devices. Marco et al. [9], presented a novel system called CloudShield, to offload mobile computation to the cloud. The idea is to run a replica (the clones) of smartphone on the cloud. Are synchronized with the corresponding devices, and help alleviate the computational burden on the real smartphones. They used a peer-to-peer network to organize the clones, in order to facilitate content sharing among the mobile smartphones. Amir et al.[10], presented a cloud-based smartphone-specific intrusion detection system with a response engine. The system continuously performs an in-depth forensics analysis on the smartphone to detect any misbehavior. The disadvantage of this system is that it can be used on the device mobile only. So, it will consume the device resources. Lakshmi et al. [3], proposed a generic architecture for providing security services in the cloud for smartphones within a corporate environment. Their results support the idea of offloading the computationally expensive security functions from smartphones to the cloud environments. Georgios et al. [11], presented a system called Paranoid Android. The system offers versatile protection for smartphones.

The system views the security as another service at a higher level that can be hosted in the cloud. The basic idea is to run a synchronized replica of the smartphone in a security server in a cloud. The system focuses on detection of attack like Zero day attacks and memory resident attacks. However, the system cannot prevent attacks occurrences. Bo et al. [12], decreased the signature assigning cost by optimizing the signature library. They benefit from common characteristics of viruses such as self-replication and seasoning. Moreover, they decreased the number of unnecessary signature matching. They also raised the efficiency of the comparison procedure by rearrangement of signature library. Treadwell et al. [13], suggested analyzing the obfuscation pattern before unpacking. They provide the chance to prevent malware from further execution.

The advantage of their proposal that they used a heuristic detection approach to targets obfuscated binary files being loaded into memory prior to execution. Jon et al. [14], proposed a new model to move the mobile antivirus functionality to a network service. They employ a multiple virtualized malware detection engines. So, the anti-virus scanning is performed in cloud. The limitation is that, the data privacy is questionable and also suffers from the problem of disconnection.

3. The Proposed System

One advantage of using the cloud environment is to allow using multiple antivirus engines, in parallel, to determine the malicious files. Using of multiple engines increases the detection rate as proposed by [14]. In [14], they use multiple antivirus engines in parallel to detect malware. These engines are ClamAV (CM), Symantec (SM), McAfee (MA), Bit-Defender (BD), and F-Secure (FS) [14].

As an advantage of the proposed framework, the security service is located in the cloud. This is unlike the existing security services that suffer from technical constraints. In this paper, we proposed a new framework to provide a remote protection to the smartphones. We will use the cloud services to scan mobile phones from malware. The idea is to host replicas of smartphones applications on the cloud. So, we can apply various security functionality outside the mobile device. In this case, the detection algorithm of malicious applications is not done on the smartphone itself, but in a cloud service. The basic concept behind the proposed system is shown in Figure 1. The main motivation for this proposed system is the resources constraints in smartphones. These resources are battery consumption, storage capacity, and processing power. These constraints could be circumvented by using the security functionality as a service. We aim to provide an optimal solution to the virus detection and improve performance.

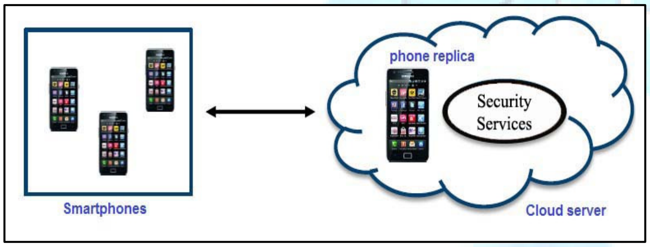

Figure1: Overview of the proposed system

The proposed system is divided into two parts, as shown in Figure 1. The first part is related to smartphone. In this part, a mobile agent is developed that responsible for collecting files and sends them to the cloud for analysis. In this part, we develop an android based application to transfer the user files, from the user's mobile, to be scanned on the cloud.

The second part is located into cloud which is a proxy server responsible for managing the communication between the smartphone and the cloud server. In this part, we used the software as service (SaaS), which is an information delivery model that utilizes existing technologies [15].

Table 1: Summary of the advantages and disadvantages of static and dynamic analysis

	Advantage	**Disadvantage**
Static analysis (Signature-based)	- Fast and safe. - Low false positives. - Good in analyzing multipath malware.	- Difficult in analyzing unknown malware. - Cannot deal with simple obfuscation.
Dynamic analysis (Behavior-based)	- Good in detecting unknown malware.	- Neither fast nor safe. - Difficult in analyzing multipath malware.

This part is divided into two modules. The first module is the registration module. In this module, the user information is registered such as the device type, operating system, and application. The security service module receives files from the agent and determines whether a file is infected or not. The second module is the security service module. In this module, we used multiple engines to detect the malware: the Static analysis technique (Signature-Based) and dynamic analysis technique (Behavior-Based). So, we combine both techniques to increase the advantages, decrease the disadvantages, and improve the detection accuracy. This process is detailed in Figure 2. The advantages and disadvantages of both static and dynamic technique are shown in Table 1.

The second module of the second part is the security service module. This module uses both static and dynamic scanning methods. Figure 3 illustrates the flowchart of malware detection by the proposed system.

In the static method, we used the Signature Optimizing Pattern Matching that depends on the virus signature [12]. The input files will be compared to the signatures which already stored in a database. The comparison aims

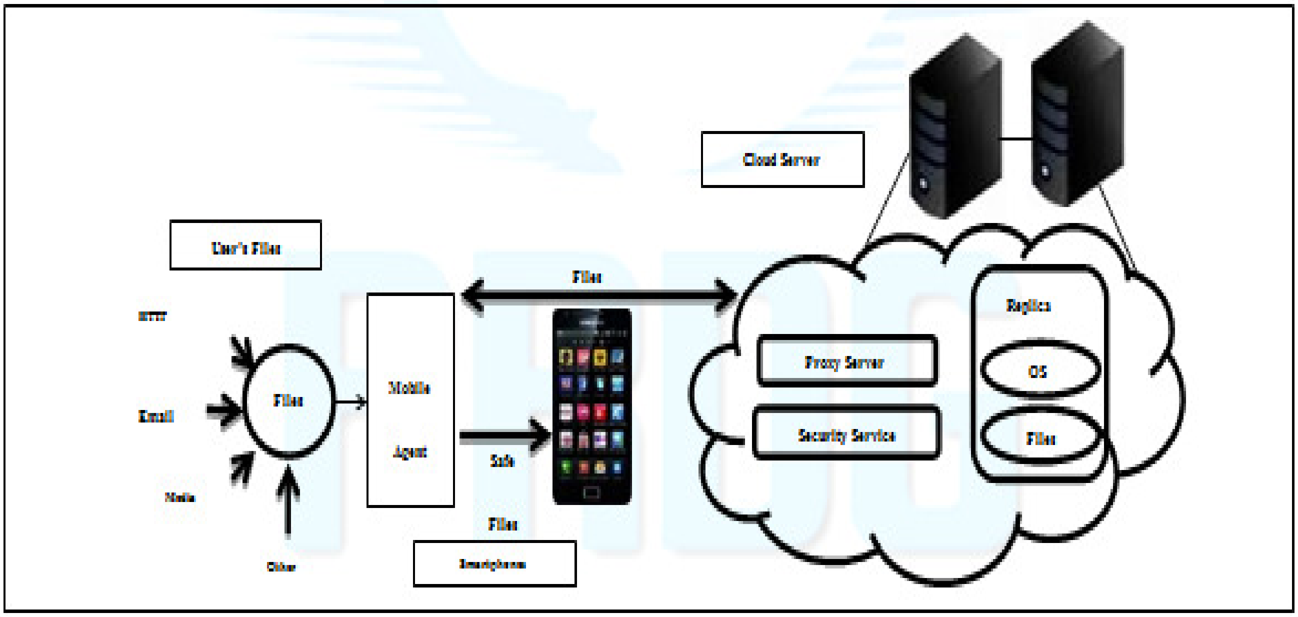

Figure 2: Details system description

to find code segments in a file. This is similar to DNA or protein sequences comparison. These files are scanned by using string matching algorithm to determine whether these files are malicious or not.

Based on the virus characteristic of self-replicating, this proposed system optimizing policy focus on signature database. One common feature of virus is that it will inject target files with malicious code into the normal files. So lots of replicas exist within one file. When any virus is detected by signature match, this virus signature is temporarily stored in the buffer.

So, the other replicas do not need to be matched against the other large amount of signatures in the actual signatures database. Therefore, these replicas will be matched with the buffer. So, this pre-comparison with already-detected viruses will reduce the time of signature matching [12].

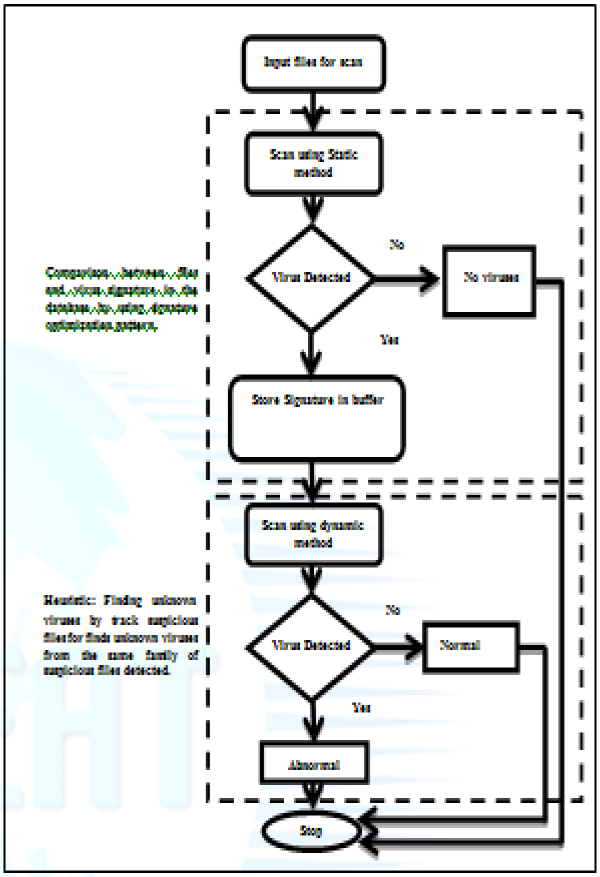
Input files for scan
Scan using Static method
Comparison between files and virus signature in the database by using signature optimization pattern.
Virus Detected
No
No viruses
Yes
Store Signature in buffer
Scan using dynamic method
Heuristic: Finding unknown viruses by track suspicious files for finds unknown viruses from the same family of suspicious files detected.
Virus Detected
No
Normal
Yes
Abnormal
Stop

Figure3: Simple outline of processes used in proposed framework

Then, we used the dynamic analysis technique (Heuristic-Based) [16]. Heuristic analysis methods are suitable for unknown malware. It is used to find previously unknown viruses or to detect new variants of known viruses. Instead of looking for specific signatures, heuristic scanning looks for certain instructions or commands within a program that are not exist in normal behavior files. However, heuristic analysis is depends on analyzing suspicious file's characteristics and behavior to determine whether it is a malware or not. For example, many malicious programs search for executable programs, and open the files that found and modify them. A heuristic method examines a file's code and increases its "suspicious counter" for that application if it encounters a suspicious command. If the value of the counter after examining the entire code of the file exceeds a predefined threshold, the object is considered to be infected. However, heuristic engine is able to detect malicious functionality in new and previously unexamined files. Such as the viruses replicates itself in the file.

Each time a new virus is discovered, the anti-virus vendors release a signature update for that virus. Those signatures are basically snippets of code extracted from the actual virus. So, enable vender's anti-virus scanners to detect malicious programs. If a file contains code that matches a signature, then there is a strong probability that the file contains a virus.

Malware Type	Infected Num	Correct Detected	False Positive	False Negative	Detection Rate
Geinimi	30	28	3.7%	3%	93.3%
DroidDream	30	27	5.4%	4.6%	90%
Plankton	30	27	4.3%	5.7	90%

Table4: Result of detection malware

The proposed signature-based detection (static detection) is an effective and computationally efficient method for virus detection. However, it has some shortcomings. Firstly, viruses are becoming increasingly sophisticated and use polymorphism (automatic mutation). It uses other concealment techniques, such as encryption, to evade detection. Secondly, there is always a gap between the time that a virus is discovered and the time that the vendors release a signature. This gap represents more risk. The systems remain vulnerable and useless in this case. Heuristic detection is intended to overcome these shortcomings. Heuristic scanners work by detecting the behavior of the virus.

4. Evaluations and Results.

The proposed system is an android application developed for a Samsung Galaxy S3 GT-19300, with OS Android version 4.1.2, Linux kernel version 3.0.31-1287119. We measured the detection accuracy of malware. We also measured the resource utilization and power consumption of the devices. In subsection A, we show the accuracy of malware detection. In subsection B, we show the measurements of resource consumption.

A. Malware Detection Evaluation.

In the experimental results, we use 90 files to be scanned by the proposed system. We also use the False Positive rate (FPR) and the False Negative rate (FNR) to measure the accuracy of malware detection. The False Positive Rate and False Negative Rate is defined by equations 1, 2 as follows:

$$FPR = \frac{NormalAsMal}{TotalDtect} * 100 \quad \ldots\ldots\ldots\ldots\ (1)$$

$$FNR = \frac{MalAsNormal}{TotalDtect} * 100 \quad \ldots\ldots\ldots\ldots\ldots\ (2)$$

In equations 1, 2 *NormalAsMal* means the number of files that is normal files, but the detection system wrongly classifies it as a malware class. *MalAsNormal* means the number of files that it is malware, but the detection system wrongly classifies it as a normal class. The *Total detect* means the number of files that being scanned. These results are registered and shown in Table 2.

Table 2 shows the results when applying the proposed system to three famous malware. We can see that the proposed system can detect most of these types of malware. The false positive rate and false negative rate are small as shown from the Figure.

B. CPU and Battery Consumption

Battery and CPU consumption in the smartphone are two key factors when dealing with security issues. This encourages the idea of offloading computation to the cloud environment having rich resources. Therefore, we performed some experiments to evaluate the smartphone's battery and CPU consumption when performing anti-virus scanning. The *battery monitor widget* and *system panel task manager* (available in the Android app market) were used for battery and CPU measurements.

The Kaspersky antivirus was chosen for comparison as it is one of the well-known and widely-used software in smartphones. The data size used in this experiment is 1024 MB. In our experiments, Kaspersky was started and a full scan of the smartphone's file system was carried out for a time period of 2 hours. Figure 4 shows that the CPU activity reached more than 80% during the scanning period in the smartphone. Also, Kaspersky is the top applications with respect to CPU usage. Figure 5 shows that the CPU activity reached to 23% during the scanning period in the cloud. Therefore, measurements of the proposed framework in cloud and the Kaspersky in the smartphone show that the CPU activity is much less when using the cloud.

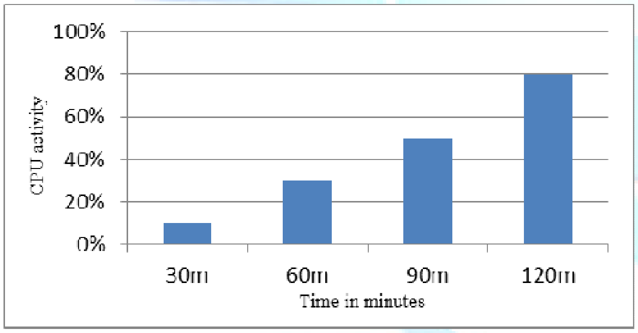

Figure 5: Kaspersky CPU consumption

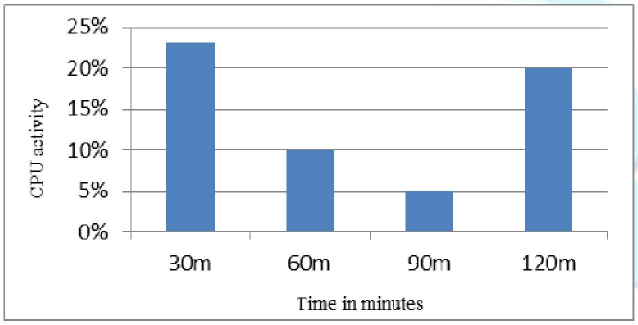

Figure 6: The proposed framework CPU consumption

Figures 6, 7 show the battery usage of the smartphone during the Kaspersky scan in smartphones and the proposed system. When Kaspersky and the proposed system were started, the smartphone was fully charged. It was in a stable state meaning that there were no fluctuations in the usage and a 100% charge was shown at the beginning of the battery monitor graph. Once we started scan virus, the battery began to discharge. After completion of the scan, the battery capacity was reduced by about only 7% in the proposed framework. However, the battery was reduced by 45% when using Kaspersky.

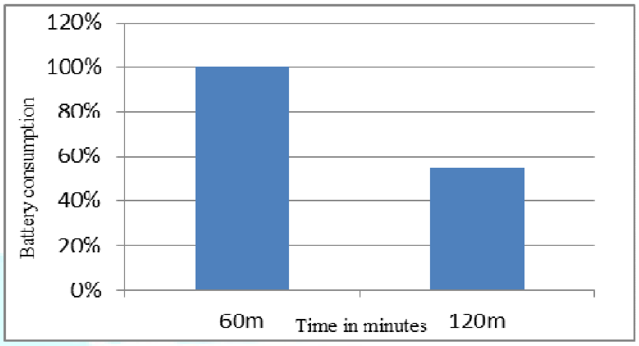

Figure 7: Battery consumption during the scanning process in smartphone.

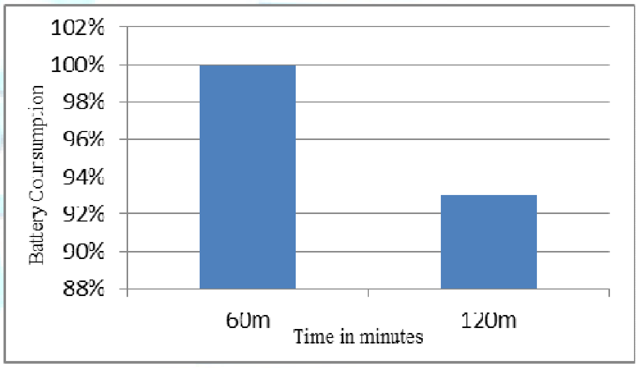

Figure 8: Battery consumption during the scanning process in cloud.

From the previous results, we can conclude that, we presented a new system based on cloud computing to deal with threats. The proposed system provides the security service for Android-based mobile devices. The proposed framework combines both Signature-Based with Behavior-Based technique to improve accuracy and scanning time for malware detection. The proposed framework achieves better detection of malicious software. Such enhancement is achieved by offloading files to a network service with rich resources. The proposed framework reduces resource consumption by transferring files to an in-cloud network service for analysis. Results show that the overall CPU use, memory use, and power is reduced compared to performing the detection analysis on the device itself. Moreover, by deploying a relatively simple agent on mobile devices, the complexity of mobile security software can be minimized. Finally, there is no posing risk to mobile hardware. If the device damaged, by a malicious application for example, files can be reconstructed from the cloud without higher cost.

5. Conclusions

In this paper, we proposed a malware detection system for mobile based on cloud computing. By moving the detection functionality to a network server, we gain many benefits. Such benefits include increased detection percentage, less complex mobile software, and reduced resources consumption. In the proposed system, we combined both the static detection and the dynamic detection techniques. Such combination ignores the drawbacks of other systems.

Results shows that, the proposed system takes less time and less resources consumption than other existing systems. Moreover, the detection rate is also high.

References

[1] Gartner Research, P. Swartz, "2.4 Billion Tablets, Smartphones and PCs Will Be Sold in 2013", 4 Apr. 2013 [Online]:http://www.dailypressdot.com/gartner-2-4-billion-tablets-smartphones-and-pcs-will-be-sold-in-2013/758741/, Retrieved at 31 May 2014.

[2] International Data Corporation (IDC), "Worldwide Smartphone Market Expected to Grow 55% in 2011 and Approach Shipments of One Billion in 2015", [Online] http://www.idc.com/getdoc.jsp?containerId= prUS22871611, Retrieved at 15 May 2013.

[3] L. Subramanian, Q. Maguire, and P. Stephanow, "An Architecture To Provide Cloud Based Security Services For Smartphones", the Wireless World Research Forum (WWRF), Düsseldorf, Germany, Oct. 2011, pp. 100-200.

[4] P. Mell, T. Grance, "The NIST definition of cloud computing", National Institute of Standards and Technology http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, retrieved at 31 May. 2014.

[5] S. Subashini, and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing", Elsevier, Journal of Network and Computer Applications, Volume 34, Issue 1, Jan. 2011, PP 1-11.

[6] D. Chaves, R. Uriarte, and C. Westphall, "Toward An Architecture For Monitoring Private Clouds", Communication Magazine, IEEE, Volume 49, Issue 12, Dec. 2011, PP 130 – 137.

[7] C. Jarabek, D. Barrera, and J. Aycock, "ThinAV: Truly Lightweight Mobile Cloud-based Anti-malware", in Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC), Orlando, Florida USA, 12 Dec. 2012, PP 3-7.

[8] A. Shabtai, and Y. Elovici, "Applying behavioral detection on android-based devices", in Proceedings of the 3th International Conference on mobile wireless middleware, operating system, and application, springer, Chicago, USA, 30 Jun. 2010, PP 235-244.

[9] M. Barbera, S. Kosta, J. Stefa, P. Hui, and A. Mei, "CloudShield: Efficient Anti-Malware Smartphone Patching with a P2P Network on the Cloud", in Proceedings of the 12th International Conference on Peer-to-Peer Computing (P2P), IEEE, Tarragona, Spain, 2012, PP 50–56.

[10] A. Houmansadr, S. Zonouz, and R. Berthier, "A Cloud-based Intrusion Detection and Response System for Mobile Phones", in Proceedings of the 4th International Conference on Dependable Systems and Networks Workshops (DSNW), IEEE/IFIP, Washington, USA, 2011, PP 31-32.

[11] G. Portokalidis, P. Homburg, K. Anagnostakis, "Paranoid Android: Versatile Protection for Smartphones", In Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), Austin, Texas USA, 10 Dec. 2010, PP 347-356.

[12] B. Li, E. Gyu, "A Signature Matching Optimization Policy for Anti-Virus Programs", In Proceedings of the 3th International conference on Computer Research and Development (ICCRD), IEEE, Shanghai, China, 11-13 Mar. 2011, PP 1-3.

[13] S. Treadwell, M. Zhou, "A Heuristic Approach for Detection of Obfuscated Malware", In Proceedings of the 09 International conference on Intelligence and Security Information (ISI), IEEE, Texas, USA, 8-11 Jun. 2009, PP 291-299.

[14] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, F. Jahanian, "Virtualized In-cloud Security Services For Mobile Devices", In Proceedings of the First Workshop on Virtualization in Mobile Computing(MobiVirt), New York, USA,17-20 Jun. 2008, PP 31-35.

[15] Kaspersky website, "Heuristic analysis in Kaspersky Internet Security" [Online]: http://support.kaspersky.com, ID: 8641, 2013, retrieved at 31 May. 2014.

[16] David Harley, "Heuristic Analysis Detecting Unknown Viruses", Technical report, http://www.eset.com/us/resources/white-papers/Heuristic_Analysis.pdf