Monitoring and Detecting Abnormal Behavior in Mobile Cloud Infrastructure

[1]Taehyun Kim, [1]Yeongrak Choi, [2]Seunghee Han, [3]Jae Yoon Chung,
[3]Jonghwan Hyun, [3]Jian Li, and [1]James Won-Ki Hong
[1]Division of IT Convergence Engineering, POSTECH, Pohang, Republic of Korea
[2]Network Technology Laboratory, KT, Daejeon, Republic of Korea
[3]Department of Computer Science and Engineering, POSTECH, Pohang, Republic of Korea
{[1]ggobugi3, [1]dkby, [3]dejavu94, [3]noraki, [3]gunine, [1]jwkhong}@postech.ac.kr, [2]seunghee.han@kt.com

***Abstract*—Recently, several mobile services are changing to cloud-based mobile services with richer communications and higher flexibility. We present a new mobile cloud infrastructure that combines mobile devices and cloud services. This new infrastructure provides virtual mobile instances through cloud computing. To commercialize new services with this infrastructure, service providers should be aware of security issues. In this paper, we first define new mobile cloud services through mobile cloud infrastructure and discuss possible security threats through the use of several service scenarios. Then, we propose a methodology and architecture for detecting abnormal behavior through the monitoring of both host and network data. To validate our methodology, we injected malicious programs into our mobile cloud test bed and used a machine learning algorithm to detect the abnormal behavior that arose from these programs.**

Keywords: Mobile cloud computing, mobile cloud infrastructure, mobile cloud service scenarios, abnormal behavior monitoring, machine learning

I. INTRODUCTION

In line with the numerous electronics manufacturers producing new mobile devices such as smart phones and smart tablets, various mobile services are being provided as applications for these devices. According to [1], there are more than 200,000 Android and 300,000 iPhone applications available as of March 2011 and these numbers are increasing rapidly. One recent trend for mobile services is their change to cloud-based mobile services. Cloud-based mobile services benefit users by richer communications and higher flexibility. Richer communications mean advanced techniques supporting such as enhanced phonebooks, messaging with push notification, and enriched call with multi-media content sharing. Massive computational processing is performed through cloud computing infrastructure instead of low-speed mobile devices. The data stored in cloud infrastructure can be accessed at any time and from anywhere through mobile devices. As a result, richer communications and higher flexibility can be provided to mobile device users through cloud computing.

Through the convergence of mobile devices and cloud services, we expect that new mobile cloud services will be provided with the virtualization of mobile devices in cloud infrastructure. Virtual smartphone over IP [2] is one example of provisioning virtual mobile instances to users. Each virtual instance in cloud infrastructure represents a mobile device, and users can connect to and use this instance. In this paper, we present a mobile cloud infrastructure as an infrastructure that provides virtual mobile instances, and those instances are managed in cloud computing architecture with massive computational processing power and storage.

However, service providers should be aware of security problems that may arise when they adopt and launch new cloud services. According to an IDC report [3], when questioned, 74.6% of service providers answered that the most important issue for cloud services is security. In addition, recent cloud computing attacks make it difficult to guarantee the trust and safety of cloud services [4][5]. For mobile cloud services, malicious mobile applications can be run on virtual mobile instances and therefore any security problems may be much severe if those applications target on the virtualization of mobile cloud infrastructure.

This paper focuses on the abnormal behavior detection in mobile cloud infrastructure. Although signature-based vaccine applications can target on virtual mobile instances to detect malware, it makes additional overhead on instances, and it is difficult for users to install vaccine software by force when those instances are provided as a service. Behavior-based abnormal detection can address those problems by observing activities in the cloud infrastructure. To achieve this, we design a monitoring architecture using both the host and network data. Using monitored data, abnormal behavior is detected by applying a machine learning algorithm. To validate our methodology, we built a test bed for mobile cloud infrastructure, intentionally installed malicious mobile programs onto several virtual mobile instances, and then successfully detected the abnormal behavior that arose from those malicious programs.

This paper is organized as follows. Section II describes related research on detecting abnormal behavior in mobile devices and cloud computing infrastructure. Section III defines mobile cloud services with mobile cloud infrastructure, and discusses security aspects with possible service scenarios. In Section IV, we present abnormal behavior monitoring methodology with the monitoring architecture of the mobile cloud infrastructure. Section V presents a validation of our proposed methodology and architecture. Finally, we summarize our work and discuss possible future work in Section VI.

This research was supported by World Class University program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea(R31-10100) and the KCC(Korea Communications Commission), Korea, under the "Novel Study on Highly Manageable Network and Service Architecture for New Generation" support program supervised by the KCA(Korea Communications Agency)"(KCA-2011-10921-05003).

978-1-4673-0269-2/12/$31.00 ©2012 IEEE

II. Related Work

A. Monitoring Abnormal Behavior in Mobile Devices

Some previous studies have focused on the detection of malware by monitoring behavior in mobile devices. *Shabtai et al.* [6] implemented a behavioral framework to detect malware for Android mobile devices. They extracted the features of CPU, memory, and network usages, monitored these using their mobile application, and then detected malware using several machine learning algorithms. *Damopoulos et al.* [7] focused on malware that are related to spamming, but their method cannot detect more general malware. They defined the behavior of mobile devices as web browsing, SMS, phone calls, and were able to detect abnormal behavior using machine learning algorithms available in Weka [15] with high accuracy.

There are other studies that also focus on abnormal behavior in mobile devices, but those studies defined the behavior of mobile devices differently. *Enck et al.* [8] related abnormal behavior of mobile devices to privacy information on mobile devices. Their framework monitors the privacy data by observing event lists in Android devices, and detected that several mobile applications can misuse users' private information. *Burguera et al.* [9] correlated behavior with the number of each system call counter, and focused on some important system calls that are related to normal applications and malware such as access(), chmod(), and chown(). However, their framework requires root permission in Android devices in order to monitor the number of system calls in mobile devices.

B. Abnormal Behavior in Cloud Computing Infrastructure

Several research groups have targeted intrusion detection for cloud computing infrastructure. *Roschke et al.* [19] discussed the requirements and proposed architecture that can detect malicious behaviors in cloud infrastructure. They identified Intrusion Detection System (IDS) management issues in the cloud considering both Host IDS (HIDS) and Network IDS (NIDS). However, their study does not focus on how those malicious behaviors are defined and detected in cloud infrastructure. *Vieira et al.* [20] proposed architecture for grid and cloud computing intrusion detection. In their architecture, they performed behavior analysis with the collaboration of each node, and also used knowledge-based analysis. However, their architecture does not reflect virtualization of each node when virtual instances are provided to users through cloud computing infrastructure. Moreover, their analysis is performed in service nodes, which can influence on the performance of cloud computing.

III. Mobile Cloud Service and Scenarios

This section defines a new mobile cloud service through the virtualization of mobile devices in cloud infrastructure. We describe two main service scenarios to explain how this mobile cloud service can be used. Service scenarios are useful to discuss security threats on mobile cloud infrastructure, because they include users, places, mobile devices, and network types, and user's interesting contents.

A. Defining Mobile Cloud Computing and the Concept of Mobile Cloud Service

Defining "mobile cloud computing" is important for the characterization and explanation of mobile cloud services. There are several definitions of mobile cloud computing that assign larger role to mobile devices for cloud computing. For example, *Warner et al.* defined mobile cloud computing as accessing the cloud through mobile devices and also mobile devices becoming part of a larger cloud construct [11]. *Marinelli* referred to mobile cloud computing as a term meaning that a number of mobile devices construct a cloud computing group, and jobs are allocated to various device nodes in order to execute computing jobs faster [10].

In this paper, we define mobile cloud computing as processing jobs for mobile devices in cloud computing infrastructure and delivering job results to mobile devices. This definition is also mentioned in other studies [17][21][22]. Based on this definition, we propose a new mobile cloud service as providing virtual mobile instances through mobile cloud computing. The proposed mobile cloud service provides virtual mobile instances through the combination of a mobile environment and cloud computing. Virtual mobile instances are available on mobile devices by accessing the mobile cloud infrastructure. This means that users connect to virtual mobile instances with their mobile devices and then use computing resources such as CPU, memory, and network resources on mobile cloud infrastructure. In this case, such mobile devices will have smaller roles to play than current mobile devices. Mobile cloud service providers can then distribute mobile applications which can connect to mobile cloud infrastructure, view, and interact to virtual mobile instances. Fig. 1 illustrates the concept of our defined mobile cloud service. By mobile cloud services, any mobile devices can be a super computer and they can support several rich services. So always keeping on computing life will be realized.

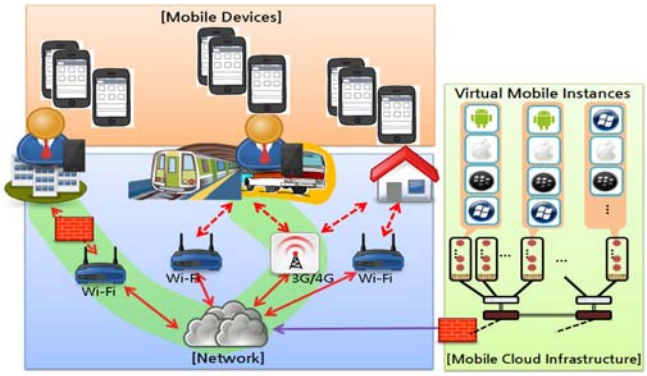


Fig. 1. The Concept behind our Defined Mobile Cloud Service

B. Service Scenarios for Mobile Cloud Services

Security threats to our mobile cloud service depend on how the service is prepared and delivered from the service providers to the actual users. We grouped together possible service scenarios on our mobile cloud service into two main categories. The first is for individual personal users and the other for office workers. Individual users use the mobile cloud service for entertainment and other individual purposes, and office workers mainly use it for smart work.

1) Individual users

Individual users are categorized as normal users, advanced users and developers according to usage types and their requirements. Following are descriptions of each individual user category and Table I provides a summary of possible service scenarios for each case in.

- Normal Users: These users are more interested in the services available from the mobile cloud environment than environment itself. The cloud environment required by these users is somewhat fixed with little change necessary.
- Advanced Users: These users are aware of overall mobile cloud services and more interested in cloud resources than normal users. They also require a cloud environment that varies more frequently.
- Developers: These users are aware of overall mobile cloud services and require a cloud environment that is specific, varied and which changes frequently.

TABLE I. SCENARIO ITEMS FOR INDIVIDUAL USERS

Normal User			
Place	***Device***	***Network***	***Consuming Content***
Subway	Tablet	3G/4G	Online game, movie
Home	Laptop	Home Wi-Fi	Movie
Advanced User			
Place	***Device***	***Network***	***Consuming Content***
Office	Smart phone	Office Wi-Fi	Smart work
Bus	Laptop	3G/4G	Simulation
Developer			
Place	***Device***	***Network***	***Consuming Content***
Subway	Tablet	Public Wi-Fi	Server management
Home	Tablet	3G/4G	Read big-size data

We show one example of service scenarios for individual users. This is a scenario for mobile application developers who develop mobile applications and run a server for the application in cloud.

- *Actor*: Mobile application developers
- *Interests*:
  - To build and test her or his mobile application in various mobile environments.
- *Preconditions*:
  - Mobile cloud service should support various mobile environments customizing function.
- *Main Scenario*
  - Actor requires several mobile environments with different display size and hardware resources.
  - Actor accesses each virtual mobile instance and test her or his application whether or not it works well in different environments with tablets through public Wi-Fi
  - Actor requires additional virtual mobile instances with strong hardware resources and uses it as a server for her or his application.

2) Office workers

We categorized office workers as staff in a main office, staff in overseas offices and subcontractors according to their office location and relationship to the company. Assuming that a mobile office system is installed in the main office, Table II provides a summary of possible service scenarios with different requirements for each case of office workers.

- Staff in a main office: These users work on mobile office systems installed in the main office.
- Staff in overseas offices: These users access mobile office systems from overseas offices to the main office. The network condition in foreign countries may be poorer than domestically accessed mobile offices. Therefore these users are interested in a stable mobile office environment.
- Subcontractors: These users contract as developers of the main office, co-work or run a project together with the company. If their contract is no longer valid, subcontractors should not use mobile office systems provided from the main office.

TABLE II. SCENARIO ITEMS FOR OFFICE WORKERS

Staff in a Main Office			
Place	***Device***	***Network***	***Consuming Content***
Subway	Smart phone	3G/4G	Messenger, schedule e-mail
Office	Laptop	Office Wi-Fi	Word, VoIP
Home	Tablet	Home Wi-Fi	Word, Approval
Staff in Overseas Offices			
Place	***Device***	***Network***	***Consuming Content***
Customer office	Smart phone	Public Wi-Fi	Document view
Office	Laptop	Office Wi-Fi	Payment, video conference
Subcontractors			
Place	***Device***	***Network***	***Consuming Content***
Office	Laptop	Office Wi-Fi	Program development, testing
Outdoor	Smart phone	Public Wi-Fi	Project management

The following is one example of service scenarios for office workers: staff in overseas offices.

- *Actor:* Staff in overseas offices
- *Interests*:
  - To access mobile office environment in mobile cloud with high speed and stability from overseas.
- *Preconditions*:
  - Special network channel such as VPN should be set to guarantee high speed connection between overseas office and mobile cloud infrastructure.
- *Main Scenario*
  - Actor accesses mobile office and shares multimedia files to co-workers in the main office with high speed connection.
  - Actor accesses mobile office via public Wi-Fi and check the payment. Although network connection is unstable because of poor network environment, he can continue his work by accessing mobile office again.

IV. Methodology and Architecture for Abnormal Behavior Detection

A. Our Abnormal Behavior Detection Methodology

Behavior means the actions of not only each virtual mobile instance in mobile cloud infrastructure itself but also mobile applications running virtual mobile instances. For example, a mobile application should use some virtual resources such as CPU or memory when it executes an action in the mobile cloud infrastructure. The application generates some network traffic data if it needs network connectivity that is internal or external to the mobile cloud infrastructure. These kinds of actions change the value of some features of virtual resources in the mobile cloud infrastructure. Thus, we assume that each mobile application and each user has a unique behavioral pattern. In this paper, we propose a monitoring and detecting methodology for abnormal behavior of virtual mobile instances and applications. If abnormal behavior is detected in one virtual mobile instance, it means that something is wrong or changed in this virtual mobile instance. At such a point a detection alarm would be notify the mobile cloud infrastructure or the actual user of this virtual mobile instance.

Virtual mobile instances in the infrastructure have the same role as normal mobile devices such as smart phones and tablet PCs. On such normal mobile devices, most current vaccine applications detect malware through a signature-based method. Signature-based methods can detect malware in a short space of time with high accuracy, but they cannot detect new malware whose signature is unknown or has been modified. If mobile cloud services are provided, much more malicious applications may appear including new and modified malware. Therefore vaccine applications cannot detect and prohibit them with only signature-based method in the future. Moreover, mobile cloud infrastructure supports a huge number of virtual mobile instances. When a malware is compromised on a virtual mobile instance, it can be delivered to other virtual mobile instances in the same mobile cloud infrastructure. Without monitoring the network behavior in mobile cloud infrastructure, the malware will spread over the entire infrastructure.

To address these security aspects, we propose a behavior-based abnormal detection methodology using both host data on virtual mobile instances and network data of mobile cloud infrastructure. We expect that we can prepare unknown security threats to mobile cloud infrastructure with our methodology. This methodology also benefits to detect already known abnormal behavior, and monitored data is also applicable to the calculation of the usage and analysis of each virtual mobile instance.

B. Architecture for Monitoring Mobile Cloud Infrastructure

Fig. 2 illustrates the architecture for mobile cloud infrastructure and the overall structure to detect abnormal behavior in the infrastructure. Mobile cloud nodes are divided into production service and non-production service nodes. Production service nodes are a group of service nodes providing services to customers directly, and non-production service nodes support mobile cloud services indirectly, such as through the processing of background jobs, managing production cluster nodes, and replacing nodes if production service nodes malfunction.

In production service nodes, each node is virtualized to provide virtual mobile instances. Hypervisors, which are installed on the physical nodes, divide privileged and non-privileged domain, and manage virtual instances (creation, modification, and deletion). Mobile virtual instances run on non-privileged domains. To monitor each virtual mobile instance, our proposed architecture installs an agent mobile application into virtual mobile instances. This agent application monitors the host data on each virtual mobile instance. Network data is monitored through port-mirroring that is provided by virtual routers.

We monitor host data using agent programs which are installed in each virtual mobile instance. It can monitor mobile host information in detail, including CPU and memory usage in a virtual mobile instance. Host data are sent to Analyzer through control network (peth1) not to influence on data network (peth0). The data network line is mainly used for the connection between virtual mobile instances and external servers or clients.

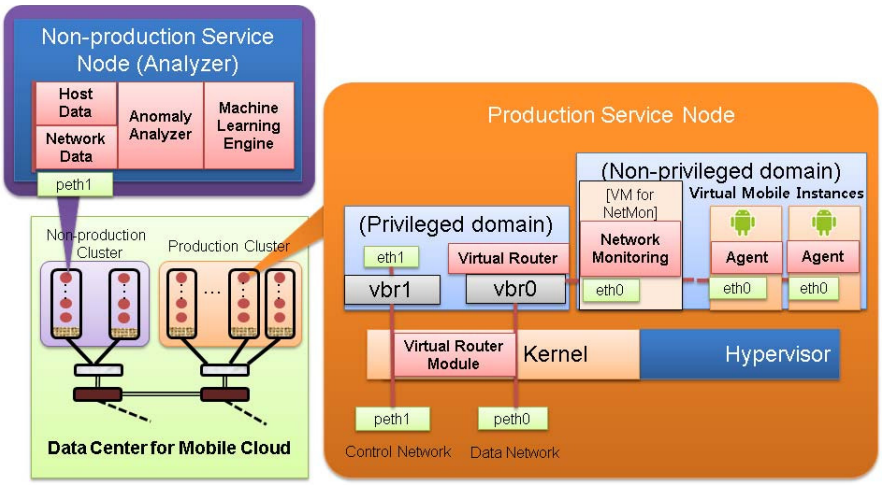


Fig. 2. Proposed Architecture for Abnormal Behavior Monitoring and Detection in Mobile Cloud Infrastructure

For network monitoring, we used port mirroring functionality via a virtual router. The virtual router installed on the hypervisor is designed to link and manage virtual network interfaces and to control the route of network traffic in those interfaces. To minimize the overhead of the hypervisor in each physical node, we installed one additional non-mobile virtual instance (VM for NetMon) on non-privileged domain, and configured the entire traffic in all virtual mobile instances so that it is mirrored to this instance. Fig. 3 illustrates a context diagram for network information monitoring. We can mirror network traffic from each virtual mobile instance to a VM for NetMon by applying the virtual routing configuration. Real network data is transmitted through eth0 and peth0, while control data including mirrored network data are transmitted through eth1 and peth1.

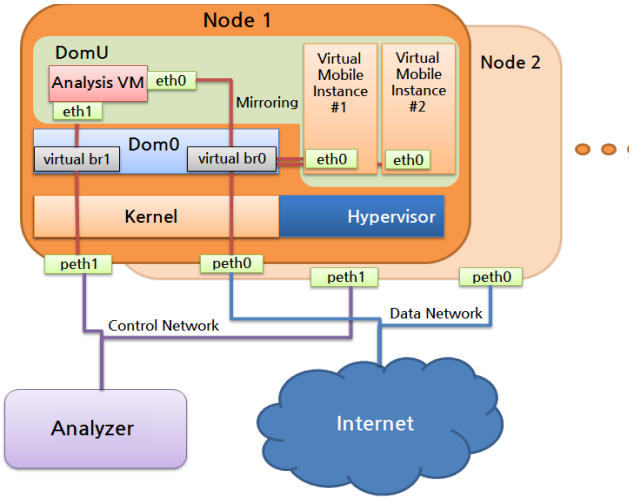


Fig. 3. Context Diagram for Network Information Monitoring

In VM for NetMon, there are two modules: the flow generator and the feature extractor. The flow generator converts copied packets to flows. It possesses all of the information on the network traffic including IP header, TCP/UDP header, and payload, the size of which is too large for use as features. The feature extractor extracts useful features such as number of flows, flow size and others from said information using network analysis tool, Tshark [18]. Then, it sends the feature information to the analyzer in the non-production service nodes. The detailed information from the flow generator can be used later for additional analysis of the detail network status in the mobile cloud infrastructure.

Finally, the analyzer analyzes the data of each virtual mobile instance and network data in each node in order to detect abnormal behavior. To detect abnormal behavior, the anomaly analyzer uses Random Forest (RF) machine learning algorithm. If abnormal behavior is detected, an alarm is delivered to network administrators or customers.

C. *Target Host and Network Features*

The analyzer uses collected data from agents in virtual mobile instances, and virtual routers to detect abnormal behavior.

1) *Host Features*

If we get the root permission in each virtual mobile instance, then we can monitor all events, including kernel-level data, and we can detect abnormal behavior more specifically and accurately. However, rooting means that it is more easily exposed to malware and becomes more dangerous than would normally be the case. Thus, in this paper, our agent mobile application does not require root permission; rather it gathers the information with general user permission, not root permission. *Shabtai et al.* analyzed the relationship between possible monitoring features and malware behavior [6]. However, the number of monitoring features was more than 80, with many of them showing low dependency. Therefore, we chose about 20 features from among the 80 which show high dependency on normal and abnormal behavior, and which are summarized in Table III.

TABLE III. MONITORING FEATURES USING AGENTS

CPU (%)	Memory (KB)	Process
CPU Usage (User)	Free Memory	CPU Usage
CPU Usage (System)	Active Memory	# of Thread
	Inactive Memory	Memory Usage
	Anonymous Memory	Context Switches
	Mapped Pages	Non-voluntarily Context Switches
Network	**OS**	Total Tx Bytes
3G Tx Packets	Running Processes	Total Rx Bytes
3G Tx Bytes	Context Switches	
3G Rx Packets	Process Created	
3G Rx Bytes	Process Blocked	
Wi-Fi Tx Packets		
Wi-Fi Tx Bytes		
Wi-Fi Rx Packets		
Wi-Fi Rx Bytes		

The data is collected using installed agent programs on virtual mobile instances. We can collect the data using mobile platform APIs (for running app processes and network information), basic user-mode commands (for CPU and OS information) and system files in the mobile platform (for memory and OS information).

2) *Network Features*

There are 3 types of malware that use network resources.

- Type 1 steals data in virtual mobile instances from behind the user and sends them to an external server
- Type 2 infects virtual mobile instances as zombie and uses them to build botnet and DDoS attack
- Type 3 increases network usage so that communication fare is over-charged

To detect these malware, not only simple network data from virtual mobile instances but also more specific and accurate data at the network level is needed. We monitored detail network data using port-mirroring functionality through

a virtual router. Flow generator converts them to flows as 5-tuple format, and feature extractor extracts useful features in Table IV from the flows in every minute. If a mobile agent is infected by one of the malware, some features among these are suddenly increased suddenly or the pattern is changed, and we can detect those network-related malware.

TABLE IV. MONITORING FEATURES AT THE NETWORK LEVEL

Network Features
Number of Hosts
Number of Packets
Number of Flows
Size (Bytes)
Number of DNS Packets
Number of HTTP Packets
Number of HTTPS Packets
Number of Well-known port Packets
Number of Other port Packets

D. Abnormal Behavior Detection

The analyzer performs machine learning algorithms using a tool, Weka [15]. We used the Random Forest (RF) machine learning algorithm to train abnormal behavior with our collected data set. The RF algorithm is a combination of decision trees that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [16]. We represented the collected features as a vector with the data subsequently used to train our collected data set.

We defined three states of behavior: inactive, active, and abnormal. An inactive state means that the instance is not used and few applications are running in the background. If the user uses the instance and runs a web browser, game, or other application, then the host moves into an active state. If one or more running applications are detected as malware by repeatedly requiring root privilege or delivering local information to certain remote servers, then the host is deemed to be in an abnormal state. We gathered total 257 sample data from the virtual mobile host; inactive data for 116 minutes, active data for 36 minutes, and abnormal data for 105 minutes.

V. EVALUATION

A. Collecting Behavior Data

1) Test Environment

We implemented our test bed for mobile cloud services to validate our methodology and architecture. Table V and Table VI illustrate the hardware and software specification of our test bed, respectively. We used two nodes and ran a total of 10 virtual mobile instances from them. To virtualize mobile instances, we installed a Xen hypervisor on each physical node. For virtual mobile instances, we created virtual images by compiling Android x86 [23] kernel sources. In our test bed, Gingerbread 2.3.6 is used with Linux kernel version 2.6.39. Open vSwitch [24] is installed to use the port-mirroring functionality in each physical node.

TABLE V. HARDWARE SPECIFICATION OF OUR TEST BED

Features		Contents
Node Model		Dell PowerEdge R610
CPU	Model	Intel Xeon X5650
	Clock	2.67GHz
	Cache Size	12MB
	# of Core	6
	# of Thread	12
Memory	Type	DDR3 1333MHz
	Size	24GB (8GB x 3)
Storage	Size	2TB
Network Controller		Two dual port embedded NetXtreme IITM 5709c Gigabit Ethernet NIC

TABLE VI. SOFTWARE SPECIFICATION OF OUR TEST BED

Features	Contents
OS (Privileged Domain)	CentOS 5.6 (x86_64)
Kernel	2.6.18-194.el5
Hypervisor	Xen 4.0.0
Libvert	0.8.1
Android x86 Version	Gingerbread 2.3.6 (Kernel 2.6.39)
Virtual Router	Open vSwitch 1.2.2

2) Result of Host Monitoring

We implemented an agent mobile application for virtual mobile instances. This agent program runs as a service mode in Android x86 to collect host data. Fig. 4 illustrates the result of host monitoring using agents installed in virtual mobile instances. The data are sent to the analyzer at a set interval of time (we set this interval to one minute).

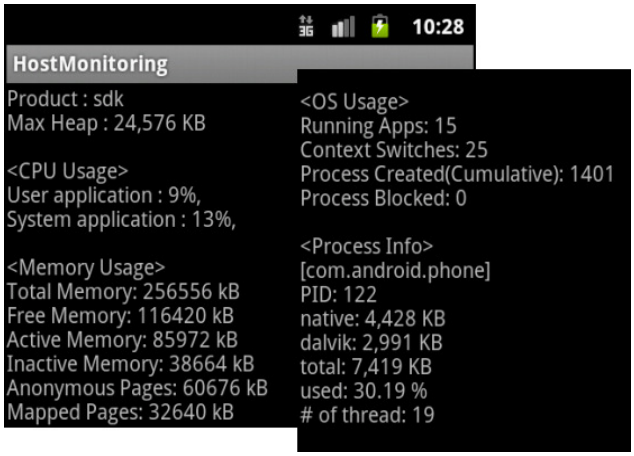


Fig. 4. Host Monitoring Result from Agents in Virtual Mobile Instances

3) Network Information

Network traffic data is mirrored to the VM for NetMon. Then, the flow generator in the VM for NetMon generates flows using the Tshark tool every minute. The feature extractor also extracts network behavior information every minute from the flows that are generated just one minute before. This network behavior information is then sent to the analyzer, also every minute. Fig. 5 shows an example of TCP and UDP flow traffic analysis using Tshark. Tshark is a terminal version of the Wireshark [18] tool. In Tshark, network traffic is displayed by host and counted for incoming, outgoing, and total frames and bytes.

```
================================================================
UDP Conversations
Filter:<No Filter>
                                      |       <-      ||      ->      ||    Total     |
                                      | Frames Bytes || Frames Bytes || Frames Bytes |
192.168.122.142:59879 <-> 192.168.1.2:53   1    125      1     74      2     199
192.168.122.142:65480 <-> 192.168.1.2:53   1    123      1     77      2     200
192.168.122.142:58865 <-> 192.168.1.2:53   1    124      1     78      2     202
192.168.122.142:61089 <-> 192.168.1.2:53   1    160      1     77      2     237
192.168.122.142:50272 <-> 192.168.1.2:53   1    161      1     78      2     239
192.168.122.142:49986 <-> 192.168.1.2:53   1    130      1     74      2     204
================================================================
TCP Conversations
Filter:<No Filter>
                                      |       <-      ||      ->      ||    Total     |
                                      | Frames Bytes || Frames Bytes || Frames Bytes |
192.168.122.142:49277 <-> 192.168.181.54:590  25  11204   13   822     38  12026
192.168.122.142:56541 <-> 192.168.82.74:80    4    953     5   1120     9   2073
192.168.122.142:56540 <-> 192.168.71.99:80    3    571     4   1183     7    175
192.168.122.142:56542 <-> 192.168.82.74:80    1     66     2    120     3    186
192.168.252.99:49962 <-> 192.168.122.181:910  0      0     2    132     2    132
================================================================
```

Fig. 5. TCP/UDP Flow Traffic Analysis using Tshark

4) Malware Data

We chose '*GoldMiner*' [12] malware applications to obtain abnormal data in our mobile cloud infrastructure. We installed the malware onto two hosts and ran it. It gathers location coordinate and device identifiers (IMEI and IMSI), and sends the information to its server. This also prompts the user to download and install an application. The application prompts user to uninstall other applications and sends a list of installed applications to the server. The malware target affecting each mobile instance as zombie, and there are many other malware which have the same purpose although their functionality and behavior are little different from each other [13][14]. This kind of malware is more threatening to mobile cloud infrastructure because there are lots of similar virtual mobile instances and they are closely connected to each other.

B. Result of Monitoring and Detecting Abnormal Behavior

Fig. 6 shows graphs of the monitoring results of five virtual hosts in a node. A malware, GoldMiner2, was installed on the 192.168.122.41 virtual machine, with the data of the machine represented in pink in the graphs. We activated the five virtual hosts by running normal applications and ran GoldMiner2 from 20:30 to 21:00 on the 192.168.122.41 host, and gathered 257 sample data. The number of context switches and the CPU usage ratio of this host were higher than other normal hosts during this time, see Fig. 6-(b) and (c). Also, the number of connecting remote hosts is sometimes higher than others during this time. According to these data, the analyzer detected abnormal behavior from the 192.168.122.41 virtual host and changed its state to abnormal, as shown in Fig. 6-(d). The 1st layer (yellow) means an inactive state, the 2nd layer (green) means an active state, and the 3rd layer (red) means an abnormal state in the graph. After running the malware, there are some periods of abnormal states because the malware was sometimes still running sometimes in the background even after stopping it from running.

With this result, we validated abnormal behavior detection using a 10-fold cross-validation to prove the accuracy of our methodology. The original sample is randomly partitioned into 10 subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times. Table VII shows the detection results of inactive, active, or abnormal states. There are 3 false labels in our results; one is detected as abnormal instead of inactive, and two are detected as abnormal instead of active. Table VIII shows the accuracy of our abnormal behavior detection results. TP Rate means True Positive rate, and FP Rate means False Positive rate. Precision means the fraction of retrieved documents that are relevant to the search, and recall is the fraction of the documents that are relevant to the query that are successfully retrieved. Weighted average shows average values of each accuracy feature according to the number of instances with each class label. Precision is high for all three states, and FP rate is almost zero. This result shows that our methodology can detect abnormal behavior of malware.

TABLE VII. ABNORMAL BEHAVIOR DETECTION RESULT: INACTIVE, ACTIVE AND ABNORMAL STATES

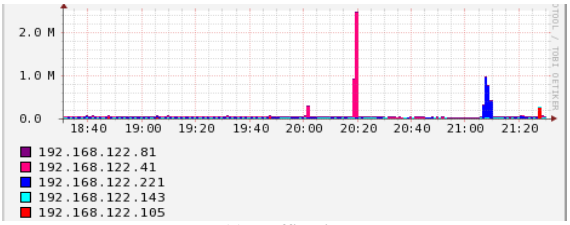
		Detection Result		
		Inactive	***Active***	***Abnormal***
Label	***Inactive***	115	0	1
	Active	0	34	2
	Abnormal	0	0	105

TABLE VIII. THE ACCURACY OF ABNORMAL BEHAVIOR DETECTION

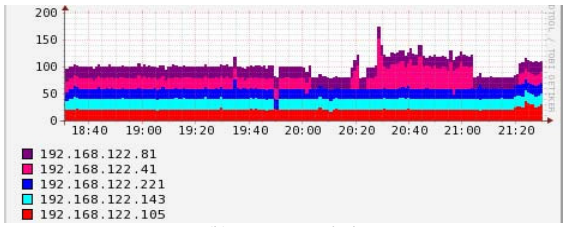
	TP Rate	**FP Rate**	**Precision**	**Recall**	**Class**
Accuracy	0.991	0.000	1.000	0.991	Inactive
	0.944	0.000	1.000	0.944	Active
	1.000	0.020	0.972	1.000	Abnormal
Weighted Avg.	0.988	0.008	0.989	0.988	

VI. CONCLUSION AND FUTURE WORK

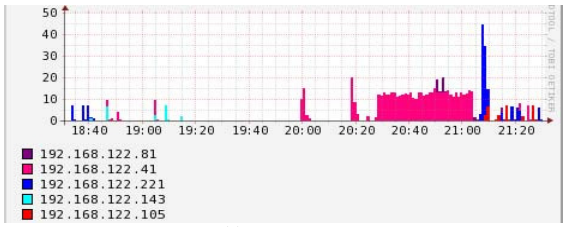
In this paper, we presented a new mobile cloud service with the virtualization of mobile devices and discussed some possible scenarios for individual users and office workers. To address security issues in mobile cloud infrastructure, we proposed abnormal behavior monitoring methodology and architecture to detect malware. These were then tested by deploying our mobile cloud test bed. Host and network data are used together to detect abnormal behavior. Our abnormal behavior detection using the RF machine learning algorithm



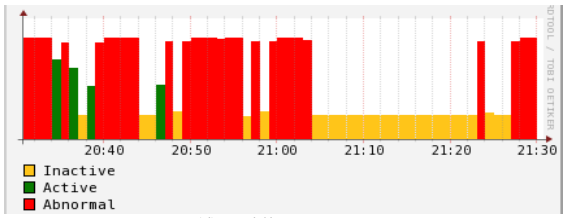
(a) Traffic Size



(b) Context Switch



(c) CPU Usage



(d) Mobile Host State

Fig. 6. Behavior Monitoring Results in Our Experiment Environment

shows that our proposed methodology and architecture successfully detect abnormal behavior.

For future work, we will investigate on the service feasibility of this new mobile cloud service. In addition to the monitoring of mobile cloud infrastructure focusing on security issues, other monitoring metrics should be considered for the provisioning and configuration, of services, and for the charging of users. We will also measure the performance of our proposed monitoring architecture. To deal with security aspects on this service, we will gather various additional types of sample malware for training in order to improve the accuracy of using various machine learning algorithms. Further, we will consider other monitoring features to improve the accuracy of detecting abnormal behavior. But there is an overhead issue such as time complexity and battery consumption if we gather lots of features. So we should also consider this aspect together.

REFERENCES

[1] Distimo, "The battle for the most content and the emerging tablet market", April, 2011, http://www.distimo.com/blog/2011_04_the-battle-for-the-most-content-and-the-emerging-tablet-market/.

[2] E. Y. Chen and M. Itoh, "Virtual Smartphone over IP", The next IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2010), Montrreal, Canada, June 2010, pp.1-6.

[3] F. Gens, "IT Cloud Services User Survey, pt.2: Top Benefits & Challenges", IDC eXchange(http://blogs.idc.com/ie/), August 14, 2008.

[4] Y. Chen, V. Paxson, and R. H. Katz, "What's New About Cloud Computing Security?," University of California Berkeley Report No. UCB/EECS-2010-5, January 2010.

[5] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of Network and Computer Applications 2010, Vol.34, No.1, July 2010, pp.1-11.

[6] A.Shabtai, U. Kanonov, and Y.Elovici, "Andromaly: a behavioral malware detection framework for android devices", Journal of Intelligent Information Systems, January 2011, pp 1-30.

[7] D.Damopoulos, S.A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke, and S. Grizali, "Evaluation of Anomaly-Based IDS for Mobile Devices Using Machine Learning Classifier", Security and Communication Networks, Vol.5, No.1, January 2011, pp.3-14.

[8] W. Enck, P. Gilbert, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones", In Proc. of the USENIX Symposium on Operating Systems Design and Implementation (OSDI), Vancouver, Canada, October. 4-6, 2010.

[9] I. Burguera, U. Zurutuza and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for android", Proceedings of the 1st workshop on Security and privacy in smartphones and mobile devices (SPSM'11), New York, NY, USA, October 17, 2011.

[10] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapRecuce", a Mater Thesis, CMU-CS-09-164, Carnegie Mellon University, September, 2009, available on http://reports-archive.adm.cs.cmu.edu/anon/2009/CMU-CS-09-164.pdf.

[11] S. A. Warner and A. F. Karman, "Defining the Mobile Cloud", NASA IT Summit 2010, August 16-18, 2010.

[12] GoldMiner, http://blog.mylookout.com/blog/2010/12/29/geinimi_trojan/, December 2010.

[13] HongTouTou, http://www.securityweek.com/multiple-variants-android-virus-hong-tou-tou-surface-china, Febrary 2011.

[14] PJApps.B – Girl Mahjong Android, http://contagiominidump.blogspot.com/2011/10/pjappsb-girl-mahjong-android.html, October 2011.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and H. Ian, "The WEKA Data Mining Software: An Update", SIGKDD Explorations Newsletter, Vol. 11, No. 1. New York, NY, USA, June, 2009, pp. 10-18.

[16] L. Breiman, "Random Forests", Machine Learinig, Vol. 45, No. 1, 2011, pp.5-32, DOI: 10.1023/A:1010933404324.

[17] A. Klein, C. Mannweiler, J. Schneider, and H. Schotten, "Access schemes for mobile cloud computing", Mobile Data Management (MDM), 2010 Eleventh International Conference on, Kansas City, MO, USA, May 2010, pp. 387-392.

[18] Wireshark, http://www.wireshark.org/.

[19] S. Roschke; F. Cheng; C. Meinel, "Intrusion Detection in the Cloud", Dependable, Autonomic and Secure Computing, 2009. DASC '09. Eighth IEEE International Conference on, Chengdu, China, December 12-14, 2009, pp.729-734.

[20] Vieira. K, Schulter. A, Westphall. C.B, and Westphall. C.M, "Intrusion Detection for Grid and Cloud Computing", IT Professional , vol.12, no.4, July-Aug. 2010, pp.38-43.

[21] P. Angin, B. Bhargava, and S. Helal, "A Mobile-Cloud Collaborative Traffic Lights Detector for Blind Navigation", Mobile Data Management (MDM), 2010 Eleventh International Conference on, Kansas City, MO, USA, May 2010, pp. 396–401.

[22] D. Kovachev, D. Renzel, R. Klamma, and Y. Cao, "Mobile Community Cloud Computing: Emerges and Evolves", Mobile Data Management (MDM), 2010 Eleventh International Conference on, Kansas City, MO, USA, May 2010 pp. 393–395.

[23] Android-x86 – Porting Android to x86, http://www.android-x86.org/.

[24] Open Virtual Switch, http://openvswitch.org/.