International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 1, January 2015

Performance Analysis Of Policy Based Mobile Virtualization in Smartphones Using MOSES Algorithm

Ms.MALARVIZHI.M, Mrs.RAJESWARI.P

Abstract:

Now a day's most of the people used in smart phones. Smartphone's usage will be increased. So the security level will be implemented in smart phones are important one. MOSES algorithm improves the security in smart phones. Because it will provide separate security in data and application using virtualization techniques. We run a set of experiments using MOSES algorithm. In the simulation results the Smartphone's performance overhead is high. we proposed a modified MOSES algorithm in order to reduce the performance overhead.

Keywords— Android Security, virtualization, Performance measuring, Performance overhead.

1.INTRODUCTION:

Smartphones usage increasingly, and many users carry multiple phones to accommodate work, personal, and geographic mobility needs. Smartphones have become omnipresent devices. They combine the computing power previously known from desktop computers with the mobility and connectivity of cellular phones. Several device manufacturers are even following this trend by producing smart phones able to handle two subscriber identification modules (SIMs) at the same time. Here the third party application also installed. So security issues can arise any malicious person can open E-mail leakage of data.

One possible solution to this problem by isolating applications and data related to work separated from recreational applications and private/personal data. Within the same device, separate security environments might exist: one security environment could be only restricted to sensitive/corporate data and trusted applications; a second security environment could be used for entertainment where third-party games and popular applications could be installed. The second environment cannot access data from first environment . Such a solution could be implemented by means of virtualization technologies. The network, storage and server these type of virtualization are used in the IT network companies.

2. ANDROID SECURITY:

Android is open source platform, developers will work along to enhance it. Android platform is multitasking software; therefore no application will gain critical access to the components of OS. Android platform is UNIX based operating system that is the most secure operating system. The developers need a unique signature to publish their application on market. Users will report a possible security flaw through their Google account .All applications on android need permission from the user at the time of installation.

3. RELATED WORK:

3.1 A System for Enforcing Fine-Grained Context-Related Policies on Android: Current smartphone systems allow the user to use only marginally contextual information to specify the behavior of the applications: this hinders the wide adoption of this technology to its full potential. The concept of context-related access control is not new, this is the first work that brings this concept into the smartphone environment. In particular, a context can be defined by: the status of variables sensed by physical (lowlevel) sensors, like time and location; additional processing on these data via software (high level) sensors; or particular interactions with the users or third parties. CRêPE allows context-related policies to be set (even at runtime) by both the user and authorized third parties locally or remotely (via SMS, MMS, Bluetooth, and QR-code).

3.2 Bring Your Own Device Approaches:

Besides approaches to improve Android security in general, some solutions specifically aimed at supporting the BYOD have been proposed.

3.2.1 Secure Container:

Secure container used for separate the data and application.

3.2.2 Mobile Virtualization:

Virtualization provides isolated environment. It will create the virtual of one based on primary thing. The hypervisor is used for mobile virtualization to separate data and application in mobile. It have some advantage, (i) improve security, and (ii) reduce the cost of installed of applications.

4. MOSES OVERVIEW:

4.1 Existing System:

MOSES means MOde-of-uses Separation in Smartphones (MOSES). MOSES provides for separating data and apps decided different contexts that are installed in a single device. Here the corporate data and apps can be separated from personal data and apps within a single device. This technic provides compartments where data and apps are stored. MOSES enforcement mechanism guarantees data and apps within a compartment are isolated from others compartments' data and apps. These compartments are called Security Profiles in MOSES.

Security profile have set of policies. MOSES provide automatic activation of SP depending on the context, in which the device is being used. A context definition is a Boolean expression defined over any information that can be obtained from the smartphone's raw sensors (e.g., GPS sensor) and logical sensors.

In the system the owner have separate password. The group member have separate user id and all of other uses have separate user id. The group member and all other user cannot access data from owner profile.

The previous version of MOSES used on Taintdroid to split data between different profiles. In the current version of MOSES, the separation of application data is implemented on the Linux kernel level through file system virtualization approach.

4.2 Advantages:

Each security profile can be associated to one (or) more contexts that determine when the profile become active. Both contexts and profiles can be easily and dynamically specified by users. Switching between security profiles can require user interaction (or) Be automatic, efficient, transparent to the user. Virtualization used for file system. Attribute based policies are supported. security profile management will be improved. One of the features introduced in MOSES is the automatic activation of SP depending on the context, in which the device is being used. MOSES can be used for realising a Mobile Device Management solution to manage remotely the security settings of a fleet of mobile devices.

4.3 Disadvantage:

The currently working MOSES algorithm does not separate system data (e.g., system configuration files) and information on SD cards. Moreover, performance overheads are high. The energy, storage and security profile switching overhead is high.

5. IMPLEMENTATION:

5.1 Context Detection:

The context detection used for identifying which security profile will be activated. Here three types of security profiles are used. To separate different security profiles each security profile have own Context_id.

5.2 Filesystem Virtualization:

The file system virtualization used for separate different data in to different security profile. Virtualization means creation of virtual something. Here the file will be created virtually and data stored in different security profiles.

5.3 Dynamic Application Activation:

Each SP is assigned with a list of application UIDs that are allowed to be run when this profile is active. MOSES uses these identifiers to control which applications can be activated for each SP.

During the SP switching, the MosesAppManager selects from the MOSES database the list of UIDs, which are allowed in the activated profile, and stores it into the set of allowed UIDs. When a new SP is activated, only the of the allowed applications for this profile will be displayed.

5.4 Attribute-Based Policies:

Within each SP, MOSES enforces an attribute based access control (ABAC) model. In the concept the system want to installed the application otherwise it deny the application.

.

5.5 Security Profile Management:

The security profile management used to manage the different security profiles with in a single device.

6 .PROPOSED SYSTEM:

6.1 Modified MOSES algorithm :

In the existing system the Performance are measured energy, storage, security profile switching. These performance are measured using network simulator -2. In the simulation results the performance of energy, storage, security profile switch overhead is high. To overcome this problem this MOSES algorithm will be changed in small level. Energy consumption, Storage size and Security profile switching will be reduced by changing in the program.

6.2 Proposed system advantages:

The modified MOSES algorithm reduce the performance overhead. Energy, storage and security profile switch usage reduced. System Operate in a speed manner.

6.3 Proposed system disadvantage:

This algorithm not implemented in real time mobile phone because the cost in expensive.

7. PERFORMANCE EVALUATION:

7.1 Network Simulator – NS-2: NS-2 is an open-source simulation tool running on Unix-like operating systems. It is a discreet event simulator. It has many advantages that make it a useful tool, such as support for multiple protocols and the capability of graphically detailing network traffic.

7.2 Simulation setting:

We conducted the simulation experiments using network simulator-2 and considered the network with 10 mobile nodes. Here the base station transmit the packets randomly. The MOSES algorithm will be running on background, it will measure the smartphones energy, storage and security profile switching. The simulation setting shows in fig.2.

In the evaluation interval, packetsize, delay, control overhead, average energy consumption, residual energy consumption, normalized overhead.

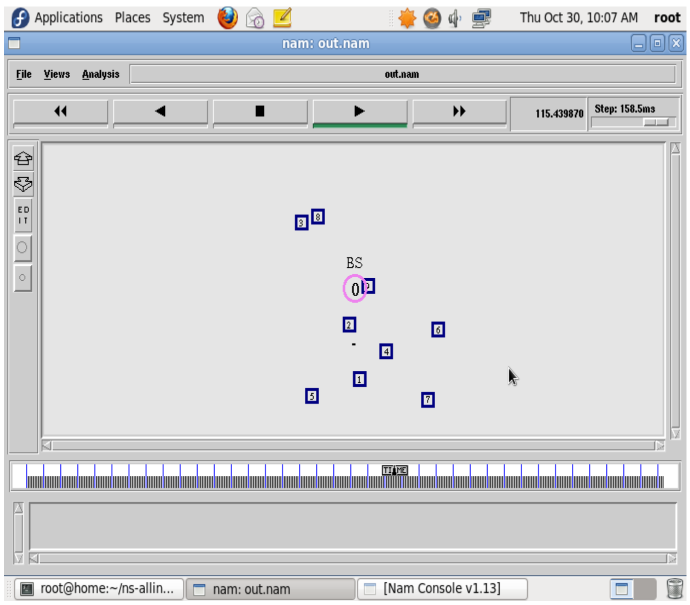

Fig.1 Simulation setting

7.3 Simulation results:

7.3.1 Energy Overhead:

To measure the energy overhead produced by MOSES, we performed the following tests. We charged the battery of our device to the 100 percent. Then, every 10 minutes we run in ten mobile application using moses algorithm. The different ten applications are running and moses the algorithm measured the total amount of usage energy. We executed this experiment for two types of systems: MOSES without SP changes, and MOSES with SP changes. The results of this experiment are reported in Fig. 2,3,4,. Here interval, average energy consumption are calculated. Also residual energy is calculated, residual energy means remaining energy in the phone. Average energy consumption with security profile is high compare to without security profile switching in smartphone.

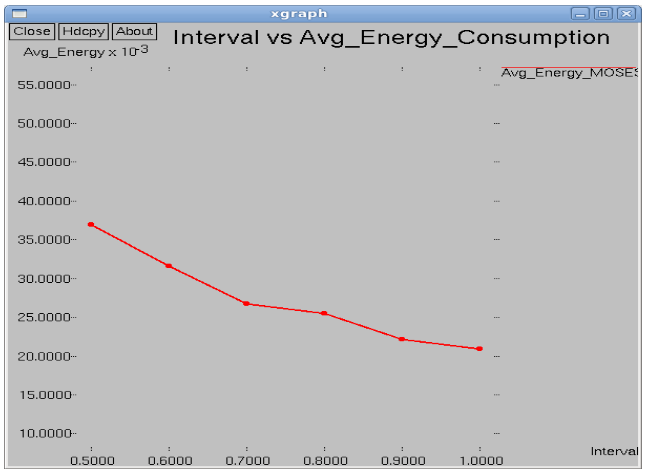

Fig. 2 Interval against delay

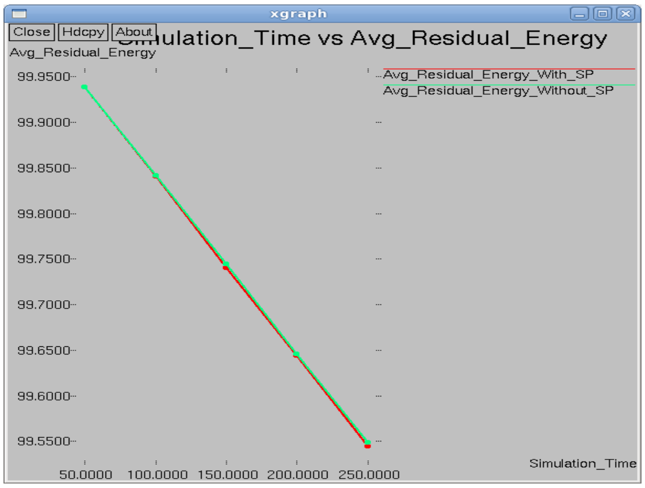

Fig. 4 Average residual energy consumption

7.3.2 Storage Overhead:

One of the most significant overheads produced by MOSES is the storage overhead. Because it have different security profile in each profile same application will be recreated so the storage size will be increase. In general, the storage size consumed where OS. MOSES measure the storage performance show fig. 5,6,

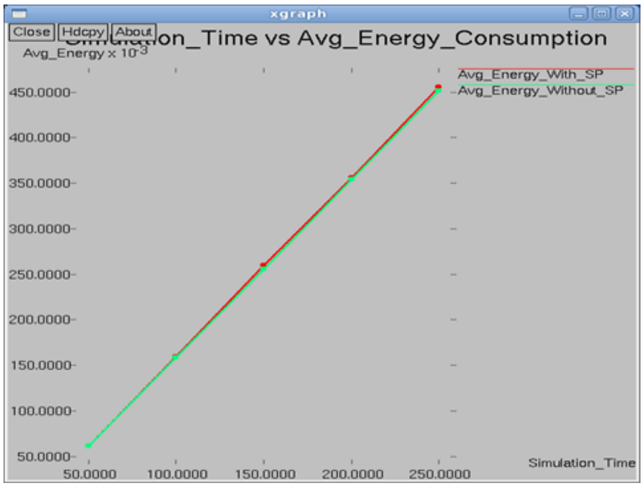

Fig.3 Average energy consumption

.

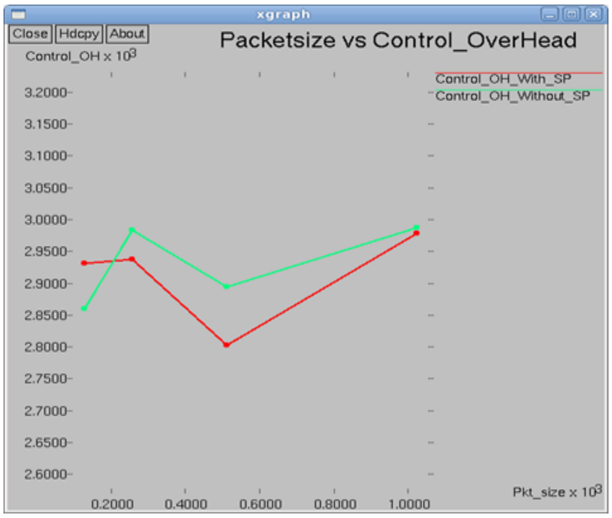

Fig.5 Packetsize against control overhead

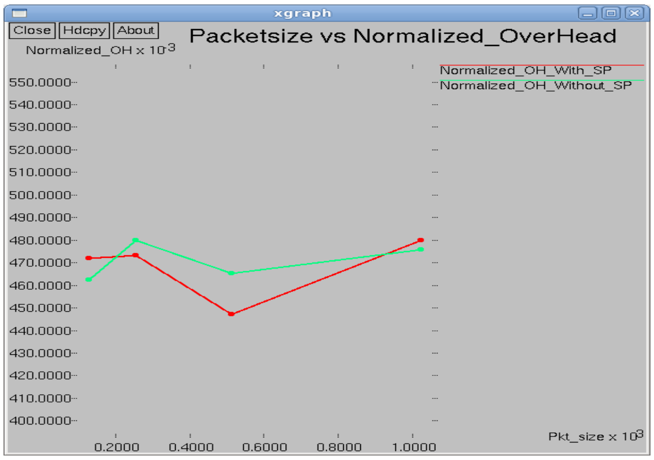

Fig.6 Packetsize against normalized overhead

7.3.3 Security Profile Switch Overhead:

In this section, we see how the performance by using with security profile without security profile how the energy consumption and storage performances are measured. MOSES performs the following security profile switching in fig7,.8.

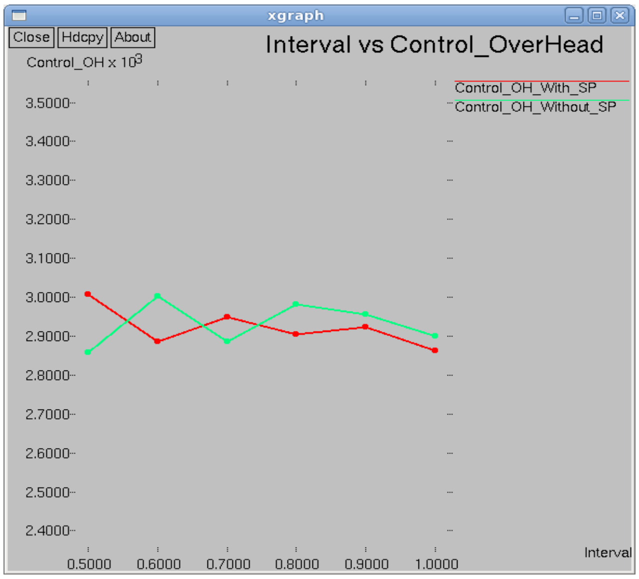

Fig.7 Interval against control overhead

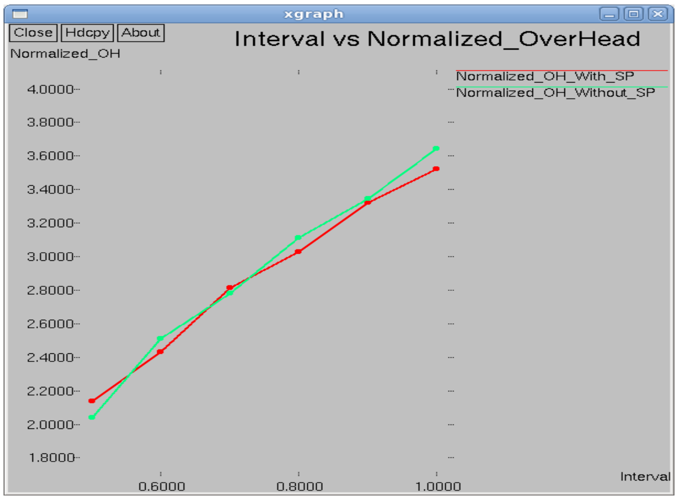

Fig. 8 Interval against normalized overhead

8. CONCLUSION AND FUTURE WORKS:

Moses provide improve security by isolating data and application by using software. In the project provide security and measuring performance using with security profile and without security profile. Here the energy, storage and security profile switching performances are measured. In the results the performance overhead is present. In future this moses algorithm is modified and reduce the performance overhead and implemented in real time smartphone.

REFERENCES

[1] Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013.
[2] Unisys Establishes a Bring Your Own Device (BYOD) Policy, 2014.

[3] T.U. Dresden, and U. of Technology Berlin, "L4Android," http:// l4android.org/ 2014.
[4] C. Gibler, J. Crussell, J. Erickson, and H. Chen, "AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications
[5] M. Lange, S. Liebergeld, A. Lackorzynski, A. Warg, and M.Peter, "L4Android: A Generic Operating System Framework for Secure Smartphones
[6] C. Gibler, J. Crussell, J. Erickson, and H. Chen, "AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications

Malarvizhi.M Gratuated with Bachelor's degree in Electronics and Communication Engineering from Dhanalakshmi Srinivasn Engineering college Perambalur,Tamil nadu,India. Currently doing Master of Engineering in Communication systems in Dhanalakshmi srinivasan Engineering college,Perambalur,Tamil nadu, India.

All Rights Reserved © 2015 IJARECE