ISSN:2249-5789
Ahmad Talha Siddiqui et al , International Journal of Computer Science & Communication Networks,Vol 5(2),45-49

Proposed An Architecture:Android Based Hardware Abstraction Layer (HAL)

Ahmad Talha Siddiqui
Research Scholar
IFTM, University
ahmadtalha2007@gmail.com

Dr. Munesh Chandra Trivedi
Professor, ABS Engineering College
Ghaziabad
munsh.trivedi@gmail.com

Abstract—Android has become the most popular mobile OS in the market. Nowadays, Smartphone are common in the population and easily available in the market. The newly developing application viz Near Field Communication (NFC) requires security features. Which are not available in current Smartphone OS features. In this context, we propose a secure architecture to prevent the Smartphone OS against cyber crimes. In this secure architecture, we used generic hardware abstraction layer and also device drivers are already implemented in this layer, no need to install manually during extended the hardware. Our proposed secure architecture allows highly secured applications to run in the virtual Machine.

Keywords—Smartphone, Secure Android OS, System Virtualization, Near Field Communication

General Term-Security

I. INTRODUCTION

Smartphone are the electronic devices which are present almost everywhere. They have a combination of computing power and connectivity. Google's Android Market [1] is the official online mechanism for delivering software to an Android application based Smartphone. Unfortunately Android application developers can upload their applications without any check of their trustworthiness. The applications are self signed by developers themselves, without the intervention of any certification authority. Unofficial repositories also exit, where developers can upload applications, including cracked application or Trojan horses. According to exhaustive International Data Corporation and Smartphone vendors will expert more than 450 million devices in 2011, and 303.4 million units 2010 [2]. Moreover, the Smartphone market has grown four times faster than mobile phone market and the demand of Smartphone has rise exponentially Table 1 depicts Worldwide Smartphone OS Market Share 2011 and 2015.

OS	2011 Market Share	2015 Market Share	2015/2011 Evolution
Android	39.5%	45.5%	23.8%
Blackberry OS	14.9%	13.7%	17.1%
iOS	15.7%	15.3%	18.8%
Symbian	20.9%	0.2%	-65.0%
Windows Mobile Phone 7	5.5%	20.9%	67.9%
Others	3.5%	4.6%	28.0%
Total	100%	100%	19.6%

Table 1: Worldwide Smartphone OS Market Share 2011 and 2015.

Apart from it, they are also very easy to carry. They are always able to connect to internet at all times with the help of Bluetooth, wifi, etc. because most of the people have become dependent on Smartphone vizAndroid, Apple, iOS through Smartphone user can do easily online ticketing, mobile banking, e-recharge, e-commerce, etc [3] These applications have high demands of security. It is the work of the OS to take care of the security of the system. But unfortunately, Android phones are having certain security problems such as delayed security updates, sufficient access control model, etc [4] to improve upon these problems, the research community proposed some solutions, i.e. extremely patent tracking, behavior analysis [5], mocking interface [6], application of mandatory access control [7], [8], analysis of remote duplicates [9], label based tracking [10] and to implement a custom privacy mode [11]. But in our proposed work, we are trying to improve an Android Security to an isolated environment application with high demands of security. The Android OS is securely encapsulated inside a Virtual Machine (VM). In this work we proposed a generic framework to secure Smartphone's and implemented a proof of concept setup that runs on a real phone. Our proposed work Secure Architecture based generic hardware abstraction layer has three important parts:

(i) Microkernel.
(ii) Virtualization.
(iii) Android Architecture.

We briefly discuss the Microkernel, virtualization, and Android Architecture

i. Microkernel

The OS kernel runs at the most privileged mode of the CPU (kernel or supervisor mode). The application runs with fewer privileges in user mode. In the contrast to monolithic kernels a microkernel implements only the essential mechanics. File system, device drivers, protocol stacks are implemented as user mode task [12]. Because of the kernel running with the highest privileges makes the reasonability critical to maintain system security has to be implements in the kernel are protection domains, schedules and means of communication between protected domains.

ii. Virtualization

A formal definition of virtualization was made by Popak and Goldberg [13]. A special piece of software, the virtual machine monitor (VMM) establishes the virtual machine. The model requires a CPU with two-mode kernel mode and user mode. According to the definition x86[14] and ARM [15] ISA were found to be non-virtualizable. To implement a virtualization on non-virtualizable architecture, several well-known work around exit

(i) Emulation

(ii) Binary Translation

(iii) Re-hosting

To enable more efficient virtualization solutions Intel and AMD added hardware virtualization capabilities to the CPU [16].

iii. Android Architecture

In general, Android is a software, stack for Smartphone's and tablets. It consists of the kernel, the Android runtimes, and libraries an application framework and the applications.

*i. Kernel:*Android is based on a specially crafted Linux kernel. We know the Google has improved Linux to better address the needs of mobile platform with improved power management between handling of limited system resources and a special IPC mechanism.

ii. Libraries: Android provides a set of native libraries that are used by various components in the system. The functionality of these libraries is exposed to application by the application framework. Examples are WebKit and SQLite

*iii. Android Runtime*is mainly made of the Dalvik Virtual Machine, a register-based Java Virtual Machine. Dalvik run java codes compiled to a special format (.dex) Android application are written in Java. Native code is integrated with the Java code through JNI. It does not benefit from theJava abstraction (automated memory management, garbage collection).

II. RELATED WORK

Related work comprises microkernel research, mobile virtualization and effort to improve Android Security. Table-1 shows a summarized view of related works.

Author	Detection Work	Platform	Description
Enck et al [17]	Anomaly	Android OS	Implement an online information flow tracking system on Android
Enck et al [18]	Anomaly	Android OS	Train Droid is a real time monitoring system for Android. Traint Droid monitors Android application and alert the user whenever a sensitive data of the user is compromised. Uses "Taint Tracking" analysis to monitor privacy sensitive information
Beresford et al [19]	Anomaly	Android OS	Mocked hardware resources to revoke application access to particular resources at run time
Zhang et al [20]	Anomaly	Android OS	Applied mandatory access control to Android with SELinux
Zhou et al[21]	Anomaly	Android OS	Implement a custom privacy mode to enable fine grained control over application's access on private information
Hwang et al [22]	Anomaly	Android OS	Report on a part of XEN to ARM platform. In XEN all VMs depends not only on the hypervisor, but on a full OS and its application that act as demo.
Schmidt et al [23]	Anomaly	Android OS	Show how a mobile trusted module (MTM) can be implementing on a microkernel based system. This work cover the SIM only
Klein et al [24]	Anomaly	Android OS	Design and implemented SeL4. SeL4 is the first microkernel formally verified to implement its specification
Schmidt et al [25]	Anomaly	Android OS	Analyzes the security on Android Smartphone from Linux-Kernel view. Uses network traffic, kernel system calls, filer system logs and event detection modules to detect anomalies in the system.

Enck et al [17] implemented an online information flow tracking system on Android. The same author presented Traint Droid in [18]. Their system used dynamic taint analysis techniques to monitor sensitive information on Smartphone's. Thus, they can track a suspicious 3rd party application that uses sensitive data as GPS location information as address book information. An application using sensitive data does not necessarily correspond to malware.

VMware introduced a mobile virtualization solution. It is designed to run an Android VM on top of Android. The setup implements a type-2 hypervisor [26] and the integrity of the VM relies on the integrity of the host OS kernel. VMware's emphasis is on manageability and it does not improve on security. Hwang et al [22] reported on a part of XEN to ARM platform. In XEN all VMs depend, not only on the hypervisor, but on a full OS and its application that act as demo with a full OS in their TCB, VMs on top of XEN are not an option for highly secure applications.

Schmidt et al [23] show how a mobile trusted module (MTM) can be implemented on a microkernel based system. The same authors also show how virtual SIMs can be implemented in such a system [27]. This work covers the SIM only. The same author proposed a solution based on monitor events occurring on Linux Kernel level [28]. They reviewed Linux based tools for enhancing security, and extracting features such as system

calls, modified files, etc, from the Linux kernel. These features were then used to create a normal model for the Smartphone behavior. At that time there were still no real Android devices available, so they could not test their system properly.

Klein et al [24] designed and implemented SeL4. SeL4 is the first microkernel formally verified to implement its specification. This microkernel is turned for verification and does not implement multiprocessor support.

OK labs implemented the microkernel [29], which is used in the Motorola Evoke. It is used to consolidate application and baseband processors. In contrast to our solution OKL4 is proprietary and closed source. In contrast to L4Re, which was specifically developed as the user mode counterpart to the Fiasco? OC Kernel, Genode [30] is a framework designed to support a variety of kernel interfaces. Fiasco. OC is supported as one of 8 different kernels. Even though Genode supports the use of Para virtualized Linux, its primary vision is a small component-based general purpose OS rather than a virtualization platform. All these solution improve on Android security, but fall short if the kernel is compromised.

III. DETECTION

There are four main issues which makes Android exposed to attacks.

i. Slow System Update.
ii. Linux Kernel.
iii. Rooted Phones.
iv. Android Permission System.

i. Slow System Update

Android is an open source software project. In software security the time span from the discovering an exposed until the deployment of the security is critical. During this time span the system is exposed to attacks and attackers race to create exhibit.Manufactures also augment Android with custom user interfaces. This add-on requires a deep modification of the Android source code. Android does not allow for selective updates, but on full system images that have to be provided by the device manufactures. These factors tell us about an important delay in the implementation of updates which make millions of device exposed to the problem of slow system updates security features like full disk encryption are introduce with new Android releases, but they are not existing versions. A recent study by Google [31] shows that distributed of different Android version revealed that 90% of the Android devices are still using.

ii. Linux Kernel

Android is based on Linux kernel, Linux implements monolithic architecture. All kernel parts including drivers, run in kernel, where no isolation of the part is provided. Any kernel bug that can be exploited enables an attacker to modify kernel memory and hereby mitigate updates and deep testing and validation of kernel code are vital to Android security. A recent study on the stock Android Froyo kernel points 88 security critical bugs [32]. This gives a rough impression of the security of Android kernels. However, due to the bad driver code supplied by vendors, we suspect the error rate of deployed kernel to be higher.

iii. Rooted phones

Rooting is the process that overcomes the kernel's integrity barrier. It can happen in two ways:

i. Voluntarily by the user who wants to be able to install additional, potentially unauthorized application. This two type of rooting can be done by installing a modify firmware including a kernel image.
ii. By malware such as Droid Dream [33] in order to gain maximum privileges on the infected system. This type of rooting is achieved by exploiting known security flows in the respective Smartphone OS.

This problem becomes even more pronounced, since two major Android device vendors announced root ability as a marketing features [34] [35] for their devices.

iv. Android Permission System

Android implements mandatory access control (MAC) in the form of a permission to access system resources at any location. The user is then presented with a screen allowing him together grant all the permission or cancel the installation. It is not possible to selectively accept or deny access to some particular users.

Another problem is that the permission is too coarse grained [36]. Using live taint tracking, Enck et al [17] found that 2/3rd of the application analyzed, exhibited suspicious handling of private data. Static code analysis revealed potential privacy leaks in even more applications [37] with many applications being distributed via the Android market; its acceptance process has the potential to filter malicious applications. However, the Android market was found to distributed malware [38] [39].

IV. PROPOSED FRAMEWORK

We consider the Android Secure Architecture based on Hardware Abstraction Layer as the main reason for its security problems. This proposed framework has many subsystems; the design of framework is based on the principle divide and conquers technique. The framework hosts multiple smaller subcomponents. Each parts implements one basic service and has the permission needed for its correct operation. Apart from it our proposed framework provided Hardware Abstraction Layer is the library which provided the links by kernel-space

drivers to the Android Service and Android manager.

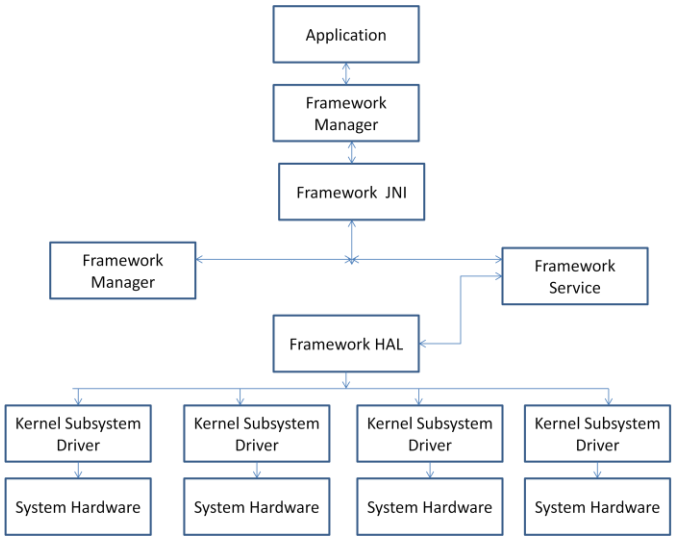

Figure-1: Android Based Hardware Abstraction Layer (HAL)

The Architecture of the Android Framework is shown in figure-1.

A. Application Framework: applications that make use of thedata from the devices. The communication starts in the manager class to then pass to the lower through the Java Native Interface.

B. Kernel: In this layer we find device drivers created using the input subsytem, a framework for all input devices. The dataare exposed to the user space through the virtual file system.

In this paragraph we examine the Hardware Abstraction Layer libraries files and its architecture model. There are two configuration files in the libraries: *configuration.h* and *sensor.h.*

The configuration.h file is used to set some parameters such as the name of the sensors, name of the sysfs files, enable or bisable information.

The sensors.h file is used to set the convert value of the data, the event type of the drivers.

V. CONCLUSION

In this paper we "proposed an architecture containing a generic hardware abstraction layer. Device specific drivers are implemented in its own layer. The Android kernel implements a generic driver interface that is the same on all devices"
The framework consists of these components. A microkernel assumes the secure foundation and is accompanied by a user friendly. Another's component is VM's envelops existing Smartphone OSes.
The security demands of future Smartphone applications security problems of those OSes are infeasible. Instead we try to show that our framework is able to create a secure execution environment for applications with high demands on security. Thereby it enables future use cases of Smartphone.

REFERENCES

[1] Google Inc. Android Market. http://market.android.com/.
[2] Ramon T. Llamas, William Stofega, Stephen D. Drake, and Stacy K. Crook. Worldwide Smartphone 2011-2015 Forecast and Analysis. Technical Report, International Data Corporation, 2011.
[3] Google Inc. Wallet. http://www.google.com/wallet, June 2011
[4] D. Barrera, H.G.Kayacik, P.C.VanOorschot and A. Somayaji. A Methodology for empirical analysis of permission-based security models and its application to android. In proceeding of the 17th ACM Conference on computer and Communications security (New York), NY, USA, 2010), CCS'10 ACM, pp. 73-84.
[5] L.Xie, X. Zhang, J.P.Seifert and S.Zhu. pBMDS: A Behaviour-based malware detection system for cellphone devices. In proceedings of the 3rd ACM conference on wireless network security (New Yourk, NY, USA, 2010), WiSec'10, ACM, pp. 37-48
[6] A.R.Beresford, A.Rice, N.Skehin and R.Sohan. MockDroid: Trading Privacy for Application Functionality on Smartphone. In 12th Workshop on Mobile Computing Systems and Applications (March, 2011).
[7] X. Zhang, J.P. Seifert and O. Acicmez, SEIP: Simple and Efficient Integrity Protection for Open Mobile Platforms. In Information and Communications Security (2010), vol. 6476 of Lecture Notes in Computer Science, Springer Berlin/ Heidelberg, pp. 107-125.
[8] D. Muthukumaran, A. Sawani, J. Schiffman, B. M. Jung and T. Jaeger measuring integrity on Mobile phone Systems. In Proceedings of the 13th ACM Symposium on Access Control Models and Technologies (New York, NY, USA, 2008), SACMAT'08, ACM, pp.155-164.
[9] G. Portokalidis, P. Homburg, K. Anagnostakis and H. Bos. Poranoid Android: Versatile Protection for Smartphones. In Proceeding of the 26th Annual Computer Security Application Conference (New York, NY, USA, 2010), ACSAC'10, ACM, pp. 347-356.
[10] Mulliner, C., Vigna, G., Dagon, D., and Lee, W. using Labelling to Prevent Cross-Service ATTACKS Against Smartphones. In Proceedings of the Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA) (Berlin, Germany, July 2006), vol. 4046 of LNCS, Springer, pp. 91-108.
[11] Y.Zhou, X.Zhang, X. Jiang and V.M.Freeh. Software Creates Privacy Mode to Help Secure Android Smartphones. http://news.ncsu.edu/release/wms-jiang-tissa/2011.
[12] J.Liedtke. on micro-kernel construction. In Proceedings of the 15th ACM Symposium on Operating Systems principles (New York, NY, USA, 1995), SOSP'95, ACM, pp. 237-250.
[13] G.J.Popek and R.P.Golberg. Formal requiremnets for Virtualization 3rd Generation Architectures. Commun. ACM 17 (July 1974), pp 412-421.
[14] J.Y.Hwang, S.B.Suh, S.K. Heo, C.J. Park, J. M. Ryu, S.Y.Park and C.R.Kim. Xen on ARM: System Virtualization using Xen hypervisor for ARM-based Secure Mobile Phones. In Consumer Communications and Networking Conference Conference, 2008. CCNC 2008. 5th IEEE (2008), IEEE, pp. 257-261.
[15] R. Bhardwaj, J.Reames, R. Greenspan, V.S. Nori and E.Ucan. A Choices hypervisor on the ARM Architecture. Department of Computer Science, University of Illinois at Urbana-Champaign 11 (2006).
[16] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M.Martins, A.V.Andersons, S.M. Bennett, A.Kagi, F.H. Leung and L. Smith. Intel Viryualization Technology. Computer 38 (May 2005), 48-56.
[17] W.Enck, P.Gilbert, B.G.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 Proceeding of the 9th USENIX Conference on Operating Systems Design and Implementation (Berkely, CA, USA, 2010), OSDI'10, USENIX Association, pp. 1-6.
[18] D.Barrera, H.G.Kayacik, P.C.VanOorschot and A.Somayaji. A Methodology for Emperical Analysis of Permission-based Security Models and its Application to Android. In Proceeding of the 17th ACM Conference on Computer and Communications Security (New York, NY, USA, 2010), CCS'10, ACM, pp. 73-84.
[19] A.R.Beresford, A.Rice, N.Skehin and R.Sohan. MockDroid: Trading Privacy for Application Functionality on Smartphones. In 12th Workshop on Mobile Computing Systems and Applications (March 2011).

[20] X. Zhang, J.P. Seifert and O. Acicmez, SEIP: Simple and Efficient Integrity Protection for Open Mobile Platforms. In Information and Communications Security (2010), vol. 6476 of Lecture Notes in Computer Science, Springer Berlin/ Heidelberg, pp. 107-125.

[21] Y.Zhou, X.Zhang, X. Jiang and V.M.Freeh. Software Creates Privacy Mode to Help Secure Android Smartphones. http://news.ncsu.edu/release/wms-jiang-tissa/2011.

[22] J.Y.Hwang, S.B.Suh, S.K. Heo, C.J. Park, J. M. Ryu, S.Y.Park and C.R.Kim. Xen on ARM: System Virtualization using Xen hypervisor for ARM-based Secure Mobile Phones. In Consumer Communications and Networking Conference Conference, 2008. CCNC 2008. 5th IEEE (2008), IEEE, pp. 257-261.

[23] A.U. Schmidt, N.Kuntze and M.Kasper. on the Deployment of Mobile Trusted Modules. In Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE (31 2008-April 3 2008), pp. 3169-3174.

[24] G.Klein, K.Elphinstone, G. Heiser, J. Andronick, D.Cock, P. Derrin, D.Elkaduwe, K. Engelhardt, R.Kolanski, M. Noorish, T. Sewell, H. Tuch and H. Winwood. SeL4: Formal Verification of an OS Kernel. In ACM Symposium on Operating System Principles (2009), ACM, pp. 207-220.

[25] L.Xie, X. Zhang, J.P.Seifert and S.Zhu. pBMDS: A Behaviour-based malware detection system for cellphone devices. In proceedings of the 3rd ACM conference on wireless network security (New Yourk, NY, USA, 2010), WiSec'10, ACM, pp. 37-48

[26] Barr, K., Bungale, P., Deasy, S., Gyuris, V., Hung, P., Newell, C., Tuch, H., and Zoppis, B. The vmware Mobile Virtualization platform: is that a hypervisor in your pocket? SIGOPS Oper. Syst. Rev. 44 (December 2010), pp 124-135.

[27] M.Kasper, N. Kuntze and A.U.Schmidt. Subscriber Authentication in Celluar Networks with Trusted Virtual SIMs. In Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on (Feb 2008), vol. 2, pp. 903-908.

[28] M.Kasper, N. Kuntze and A.U.Schmidt. Subscriber Authentication in Cellular Networks with Trusted Virtual SIMs. In Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on (Feb 2008), vol. 2, pp. 903-908.

[29] C.Walker and M.Konstant. OK Labs Enables World's First Virtualization Smartphone, with Mobile Virtualization Solution. http://www.ok-labs.com/releases/releases/ok-labs-enables-worlds-first-virtualized-smartphone-with-mobile-virtualizat, 2009.

[30] Genode OS Framework. http://genode.org

[31] Google Inc. Distributed of Android Versions. http://developers.android.com/resources/dashboard/platform-versions.html.

[32] Coverity Inc. Coverity Scan 2010 Open Source Integrity Report. http://www.coverity.com/html/press/coverity-scan-2010-report-reveals-high-risk-software-flaws-in-android.html.

[33] DroidDream. http://www.androidpolice.com/2011/03/01the-mother-of-all-android-malware-has-arrived-stolen-apps-released-to-the-market-that-root-your-phone-steal-your-data-and-open-backdoor.

[34] M.Bishop. Computer Security: Art and Science. Addison-Wesley, 2003.

[35] Droid Life: A Droid Community Blog. Motorola Eases up on locked bootloader stance, plans to unlock portfolio in 2011?

[36] D.Barrera, H.G.Kayacik, P.C.VanOorschot and A.Somayaji. A Methodology for Emperical Analysis of Permission-based Security Models and its Application to Android. In Proceeding of the 17th ACM Conference on Computer and Communications Security (New York, NY, USA, 2010), CCS'10, ACM, pp. 73-84.

[37] W.Enck, D.Octeau, P. Mcdaniel and S. Chaudhuri. A Study of Android Application Security. In Proceeding of the 20th USENIX Security Symposium (2011).

[38] D.Maslennikov. Malware in the Android Market: http://www.securelsit.com/en/blog/11267/malware_in_the_android_market_here_we_go_again.

[39] K. Mahaffey. Security Alert: DroidDream Malware Found in Official Android Market. http://blog-mylookout.com/2011/03/security-alert-malware-found-in-official-android-market-droiddream/