Journal of The Korea Institute of Information Security & Cryptology
VOL.27, NO.1, Feb. 2017

ISSN 1598-3986(Print)
ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.1.49

리눅스 보안 모듈을 이용한 모바일 장치 통제 시스템

배 희 성,[1†] 김 소 연,[2] 박 태 규[2‡]
[1]공군교육사령부, [2]한서대학교

Mobile Devices Control System using LSM

Hee-sung Bae,[1†] So-yeon Kim,[2] Tae-kyou Park[2‡]
[1]Education Division of Korea Air Force, [2]Hanseo University

요 약

모바일 단말기의 확산과 더불어 많은 조직에서 직원과 방문자의 업무 효율과 보안을 위해 BYOD 개념을 MDM을 활용하여 구현하고 있다. 그러나 응용 수준에서의 단말기 장치 통제는 보안의 근본적 해결책이 될 수 없다는 문제점이 발생한다. 본 논문은 보다 근본적이고 유연한 보안 정책을 수립하는 방법으로서 모바일 단말기의 커널 수준에서 리눅스 보안 모듈(Linux Security Module)을 사용하여 강제적 접근 제어 방식으로 단말기 장치를 통제하는 방식과 절차를 제안한다.

ABSTRACT

With the prevalence of mobile devices, many organizations introduce MDM BYOD and try to increase the level of security with them. However, device control of mobile devices in application level cannot be a solution against the fundamental problems. In this paper, we propose a more flexible and more secure method to control the hardware devices using Linux Security Module in the kernel level with the mandatory access control.

Keywords: MDM Security, Linux Security Module, Kernel Level, Mandatory Access Control

I. 서 론

편리한 다기능 모바일 단말기의 확산으로 우리는 어느 때, 어느 장소, 어느 조직에 있더라도 거의 하루를 단말기와 함께하는 경향이 있다. 모바일 단말기는 유비쿼터스 실현에 있어서 우리에게 편리한 순기능적 역할을 담당하고 있다. 많은 조직에서는 이미 업무의 효율성을 위해 개인의 모바일 단말기를 공적 조직에서도 사용할 수 있도록 하는 BYOD(Bring Your Own Device) 개념에 따라 사적·공적으로 혼용하고 있다. 그러나 이러한 경향으로 인하여 가장 큰 걸림돌이 되는 문제로 많은 조직들이 보안을 꼽는다. 어디에서든 개인이 의도적 혹은 우연히 대화를 녹음하거나, 특정 장면을 촬영하거나, 중요한 자료를 저장하거나 함으로써, 단말기의 분실, 악성 코드에 의한 해킹, 정보 공유 등으로 인하여 조직의 중요 정보가 유출되거나, 개인 정보의 침해가 발생하여, 사회에 많은 혼란을 초래하고 있다. 따라서 이러한 보안 문제를 막기 위한 법·제도적, 기술적, 물리적, 윤리적 차원에서 다각적 대책이 마련되고 있다. 그러나 이러한 보안 문제는 기술적 대책 측면에서만 보아도 용이하지 않으며, 고비용, 고수준의 기술이 필요하다. 본 논문에서는 기술적 측면에서 비교적 저비용으로 높은 수준의 보안 정책을 유연하게 구현할 수 있는 모바일 단말기의 장치 통제 방안을 다룬다. 즉, 기존의 일반적인 물리적 혹은 응용 수준에서의 보안이 아닌 안드로이드 단말기의 운영체제인 임베디드 리눅스 커널 수준에서 LSM(Linux Security

Received(09. 30. 2016), Modified(01. 03. 2017), Accepted(01. 10. 2017)
† 주저자, hezo25@airforce.mil.kr
‡ 교신저자, tkpark@hanseo.ac.kr(Corresponding author)

Kernel)[1] 방식을 활용하여 단말기의 카메라, 마이크, USB 장치를 통제할 수 있는 방식과 절차를 제안한다. 본 논문은 2장에서 기존의 물리적·절차적, 기술적 보안 통제 방법과 문제점을 알아보고, 3장에서는 커널에서의 강제적 접근 통제 방식인 LSM과 적용 시나리오를 설명하고, 4장과 5장에서 본 논문에서 설계, 구현, 시험한 모바일 단말기 장치 통제 방안을 제시한다.

II. 기존의 물리적 · 기술적 보안 통제 방법

2.1 기존의 대응 방법

현재 많은 조직에서 사용하는 제한 구역이나 보안 구역 출입자(근무자, 방문객)의 단말기 보안 통제를 위한 대응 방법은 크게 3가지로 나눌 수 있다.

첫 번째 방법으로 출입구(혹은 경비실)에서 개인의 단말기를 회수하여 일시 보관하거나, 단말기의 카메라 렌즈 부위에 보안 스티커를 부착하여 지참을 허락하는 물리적·절차적 통제 방법이다.

두 번째 방법으로 비교적 큰 조직에서는 상주 직원을 대상으로 조직 차원에서 네트워크를 통한 보안·감시·응용 및 MDM(Mobile Device Management)을 도입하고 있는데, 대부분의 MDM은 직원의 개인기기에 MDM Agent를 응용 수준(user space)에서 설치하도록 하여 조직의 다양한 관리 정책에 따라 모바일 단말기를 통합관리하고 있다[10]. 이러한 MDM 솔루션에 대한 조직의 요구사항은 다양하나, 공통적으로 요구하는 기기 통제 기능은 카메라, 녹음기, USB를 통한 조직의 정보 유출 방지이다. 단말기 사용자는 외부에서는 자유롭게 모든 디바이스를 사용할 수 있지만, 조직 내 보안 구역에서는 정보 유출의 수단이 될 수 있는 각종 디바이스를 사용할 수 없도록 하겠다는 것이다. 조직 내의 단말기 입출에 대한 통제 방식은 응용 수준에서 조직을 커버하는 Wi-Fi 기지국이나 AP(Access Point) 기반 또는 출입통제시스템과의 연계 기반 등이 적용되고 있다.

세 번째 방법은 안드로이드 단말기기의 임베디드 커널 수준에서 BYOD를 구현한 기기를 구매하여 사용하는 것이다. 예로서 삼성전자의 KNOX 단말기[8]는 LSM 방식으로 발표한 미국 NSA의 SELinux[2]와 SEAndroid[4]를 채용하여 공적·사적 앱과 데이터를 엄격 분리하는 '컨테이너화'를 실현하였다. 이 분리 정책이 KNOX에 커널 수준에서 강제적 접근제어(MAC) 메커니즘이 적용된다. 그림 1에서 보는 것처럼 하드웨어와 운영체제 커널 수준에서 논리적으로 매우 엄격하게 개인 영역과 업무 영역을 파티셔닝(partitioning)해서 제어할 수 있다.

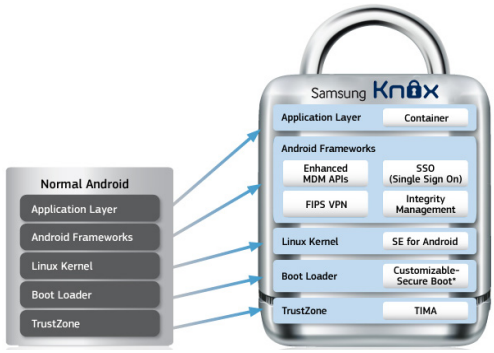

Fig. 1. Comparison of normal android and KNOX

2.2 기존 대응 방법의 문제점

물리적·절차적 통제 방법의 문제점은 첫째, 개인기기의 회수·보관으로 인한 개인의 통화권 박탈의 문제가 발생되며, 둘째, 상주 직원의 경우 개인의 단말기를 업무에 활용할 수 없기 때문에 불편하고 업무상 비효율적이어서 통제 대상을 단지 방문자에게만 국한하여 적용할 수밖에 없게 된다. 보안 스티커를 카메라에 부착하여 개인이 기기를 지참하도록 하는 방법은 출입자가 많을 경우 번잡하게 대기하게 되며, 물리적으로 보안 스티커를 일시적으로 제거하여 중요 장면을 촬영을 하는 경우를 예방할 수 없기 때문에 보안에 취약하여 정보 유출의 문제를 야기할 수 있다.

MDM 솔루션의 사용은 도입·운영에 따른 비교적 많은 비용과 관리상의 노력에도 불구하고 역시 한계점이 존재한다. 첫째, MDM 특성상 응용 수준에서 Agent 앱이 설치되기 때문에 비교적 용이하게 Agent 앱을 삭제하거나 중지될 수 있기 때문에, 이 경우 모바일 기기는 통제에서 벗어날 수 있어 보안에 취약할 수 있다. 둘째, Wi-Fi 네트워크를 통해 보안 구역 내의 디바이스를 기지국 Wi-Fi나, 조직 내 AP에서 관리하게 되는데, 전파의 특성상 음영 지역(Propagation Shadow Region)[9]이 발생할 수 있다. 이 음영 지역이 보안 구역 내에 존재하게 되면 장치 통제가 정상적으로 이루어지지 않을 수 있다.

또한 조직 내부의 보안 구역에 따라 보안 정책이 달라질 수 있는데, 이 경우 일반적으로는 MDM은 획일화된 공통 보안 정책만을 적용하기 때문에 각 보안 구역에 따라 다른 정책을 유연하게 적용하기가 어렵다. 그리고 MDM은 응용 수준에서 구현이 되기 때문에 악성 코드 대응에 근본적이지 못하다. 즉, 응용보다 저 수준에서 루팅(rooting) 등을 통하여 보안 정책을 우회할 가능성이 존재 하며, 기기 장치의 조작 실행의 상세 로그가 부족하여 보안사고 발생 시 정밀한 감사 추적에 부적합하다.

마지막으로 BYOD가 하드웨어와 운영체제 커널 수준에서 구현된 KNOX 단말기를 사용하는 방법은 조직에서의 공적·사적 단말기 장치 제어에 있어서 좋은 선택이 될 수 있을 것이다. 그러나 전용 단말기를 사용해야하므로 추가비용이 들어간다는 단점이 있어 특정 조직을 대상으로 제한적으로 사용되고 있다.

III. 모바일 장치 제어 시스템의 적용 시나리오

3.1 장치제어 시스템의 기본 요구사항

기존 대응 방법의 문제점을 분석한 결과를 바탕으로 조직의 근무자와 방문자의 단말기 장치를 통제하기 위한 요구사항을 최소한 만족시키기 위해서 본 논문에서는 본 시스템의 요구사항을 다음과 같이 정의하였다.

첫째, 비용 측면에서 효과적으로 적용할 수 있도록 구현 한다. 둘째, 기본적으로 다기능 모바일 단말기에 장착된 모든 카메라 장치(예로 전방·후방), 마이크 장치, USB 저장장치를 엄격하게 통제할 수 있어야 한다. 셋째, 조직의 차별화된 보안 정책에 따라 유연하게 선택적으로 적용할 수 있도록 해야 한다.

3.2 장치제어 시스템의 적용 시나리오

조직 내 출입을 원하는 출입자(근무자, 방문자)는 그림 2의 적용 시나리오와 같은 절차를 밟게 된다. ① 출입구(entrance)에서 관리자는 출입자로 하여금 조직의 관리용 Agent 앱을 스스로 자신의 개인 단말기에 설치하도록 한다. ② 설치가 끝나면, 출입구에서 관리자의 단말기 NFC와 출입자 NFC 간에 통신을 통하여 관리자가 출입자 기기에 인증(NFC 7-바이트 UID, 패스워드)을 한다. ③ 인증이 완료되면, 출입자가 방문할 조직 내 부서의 보안 정책에

Fig. 2. Applied scenario of device control system

따라 카메라(전방, 후방), 마이크, USB에 대하여 관리용 Agent로 전체 혹은 선택적으로 해당 장치에 잠금(lock)을 설정한다. ④ 출입자는 장치에 잠금 설정이 된 자신의 단말기를 지참하고, 방문을 원하는 해당부서(security zone)로 들어간다. ⑤ 해당부서에서 업무를 마치면 출구(exit)로 나온다. ⑥ 출구에서 관리자는 다시 NFC와 출입자 NFC 간에 통신을 통하여 관리자 인증을 한 후, 출입자 단말기의 장치 조작 행위가 기록된 로그 파일(log file)을 관리자 단말기로 전송하여 저장한다. ⑦ 관리자는 로그 파일의 장치에 대한 위반 여부를 검사하고, 이상이 있는 경우 필요한 조치를 수행한다. ⑧ 이상이 없는 경우, 관리자는 설정된 장치에 대한 보안 잠금을 해제(unlock)한다. ⑨ 설치한 Agent 앱은 출입자가 원하는 경우 설치한 상태로 유지/삭제(uninstall)하고 출입구를 떠난다.

IV. 모바일 장치 통제 시스템 설계

4.1 리눅스 보안 모듈(LSM) 소개

운영체제에서 사용하는 보안 메커니즘은 인증, 접근제어(임의적 접근제어, 강제적 접근제어)와 암호화 등이 일반적이다. 특히 강제적 접근제어를 리눅스 커널에 모듈 형태로 추가할 수 있도록 구현한 한 대표적인 프로젝트는 SELinux[2,5], TOMOYO Linux[3] 등이 있다.

SELinux는 미국 NSA의 지원을 받은 오픈 소스 보안 프로젝트로서, 커널 수준의 보안 정책 실현과 보안 정책을 독립적으로 동작하도록 하는 LSM 보안 정책 메커니즘을 지원한다. 아울러 최근 NSA의 SEAndroid[4]는 안드로이드 플랫폼의 보안 문제를 SELinux를 Android에 적용함으로써 보안을

향상시키고자 한 시도이다. TOMOYO Linux는 시스템 분석 도구로서 자원의 감시가 가능하고 그것을 이용한 강제적인 접근 제어를 지원하는 프로젝트이다.

리눅스 보안 모듈(LSM)은 리눅스에서 정의된 인터페이스를 이용하여 다양한 접근 제어 모델을 커널 수준에서 구현할 수 있도록 해 준다. 이로써 LSM을 통하여 커널 수준의 보안 모델 구현을 임베디드 리눅스 커널과 독립적으로 가능하게 한다.

4.2 장치 제어 LSM 구조 설계

그림 3은 본 논문에서 제안하는 방법에 따른 커널과 안드로이드 프레임워크 스택을 보여준다. 기본적인 프레임워크 스택과 다른 점은 커널 공간에 모듈을 추가 한 것이다. 이것은 기존 프레임워크의 수정 없이 커널 공간에 LSM 보안 모듈을 추가 하는 것만으로 강제적 접근 제어(MAC, mandatory access control) 기능이 가능하다. 응용 수준에서 요구하는

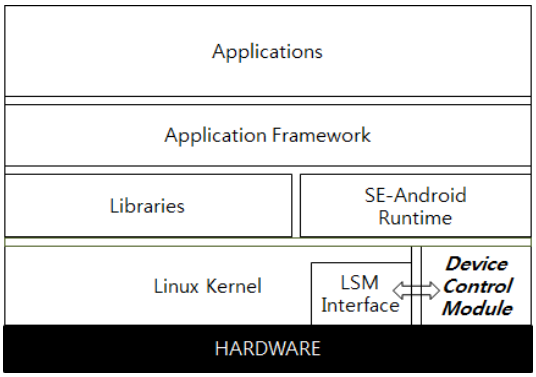

Fig. 3. Stack of Kernel, Android & LSM module

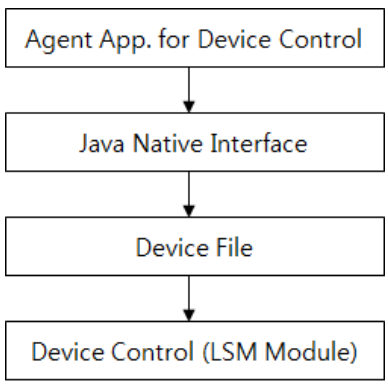

Fig. 4 Path from Agent App. to LSM Module

강제적 접근 제어를 위한 장치 통제 정보를 관리자 JNI(Java Native Interface)와 Device File을 거쳐 LSM이 적재된 커널 수준까지 보내기 위해서는 그림 4와 같은 접근 경로를 거친다.

커널에서 그 정보를 받기 위한 인터페이스로 장치 파일은 문자형 디바이스(character device)로 설계하였으며, 커널 모듈까지 전달된 정보는 그림 5와 같이 정책 테이블(device control table)에 저장된다. 또한 LSM 장치 통제 모듈은 정책 테이블에 기록된 장치에 대해 직접 관여한다. 어떤 응용이 장치에 접근하기 위해서는 그림 5와 같이 장치 파일에게 open시스템 콜을 요청할 때 최종적으로 접근 여부를 보안 모듈의 제어 엔진에게 질의하게 된다. 이

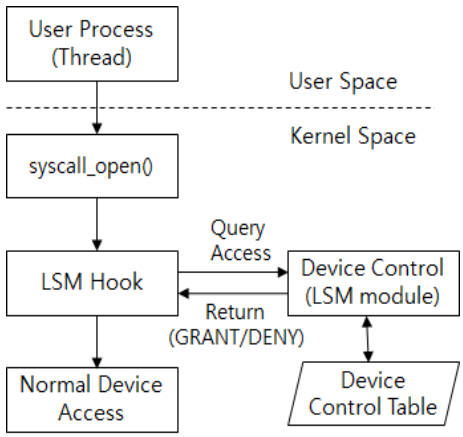

Fig. 5. MAC Procedure to device control table

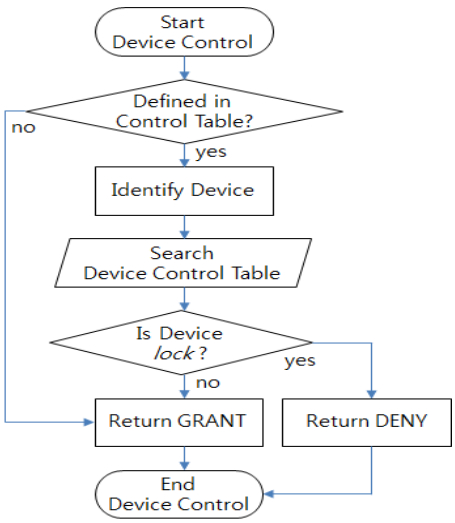

Fig. 6. Enforced algorithm for device control

때 제어 엔진은 질의를 허가할지 거부할지를 판단하게 된다. 반환된 결과는 그림 6으로 돌아가서 최종적으로 장치 접근 허가 여부를 되돌려주게 된다.

그림 7은 LSM 모듈을 적재한 후, 장치 통제 테이블에 잠금/해제를 설정했을 때의 커널 내 구조이다. 이 구조는 LSM 모듈을 커널에 적재함으로써 security_ops_TABLE 내의 file_perm 함수가 원래의 security_file_perm 함수를 호출하는 경로를, infosec_file_perm 함수로 후킹(hooking)하여 변경하는 모습을 보여준다. 그림 8은 관리자 Agent 앱을 통해서 장치 통제 테이블을 관리하는 커널 구조를 보여준다.

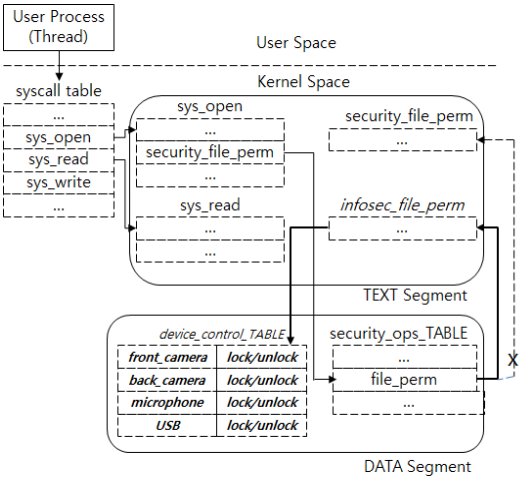

Fig. 7. Kernel structure after LSM initialization

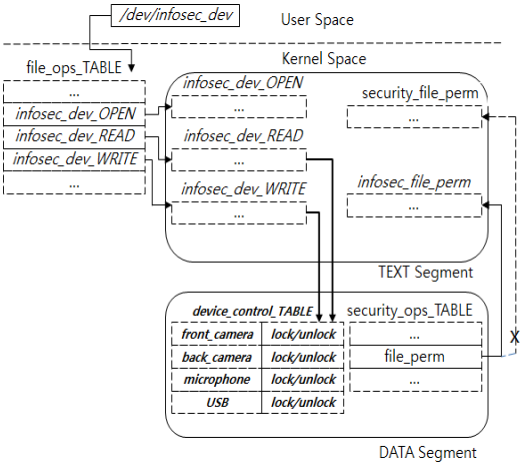

Fig. 8. Device control table manipulation by Agent App.

V. 모바일 장치 통제 시스템 구현 및 시험

본 논문에서 구현한 모바일 장치 통제 방법은 리눅스 보안 모듈을 구현하여 커널 수준에서 장치에 강제적으로 접근을 제어한다. 이 보안 모듈을 이용하여 사진 촬영(카메라 장치 제어)과 음성 녹음(마이크로폰 장치 제어)에 대해 커널 수준에서 통제의 설정(lock)/해제(unlock)가 가능하다. 구현에 따른 오버헤드 성능 평가를 위하여 안드로이드 태블릿의 표 1과 같은 환경에서 커널 모듈과 관리자 Agent 앱을 프로토타입으로 구현하고 시험하였다.

Table 1. Environment for implementation and testing

Device	SAMSUNG Galaxy Tab 10.1 Wi-Fi
Kernel version	2.6.36
Android version	3.2 (Honeycomb)

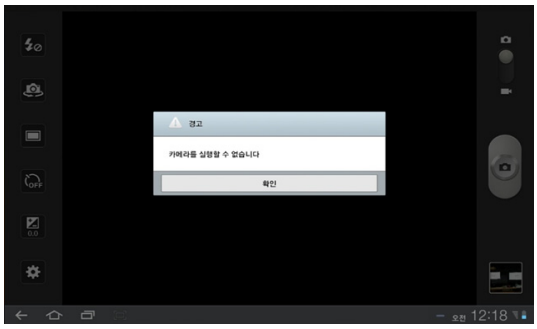

Fig. 9. Warning message disabling camera app. when cameras are locked.

5.1 장치 통제 구현

카메라를 통제 할 경우 그림 9와 같이 영상이 카메라로부터 입력이 되지 않고 검은 화면이 뜨는 것을 볼 수 있고, 화면 중심부에 카메라를 실행할 수 없다는 경고 메시지가 뜨게 된다. 마이크로폰을 통제 할 경우 그림 10에서 보는 것과 같이 응용이 정상적으로 구동이 되지 않음을 표시하는 경고 메시지가 표시되는 것을 볼 수 있다.

NFC 인증의 데이터구조는 그림 11을 통해 알 수 있다. idx는 순번 자신이 몇 번째인지 나타내는

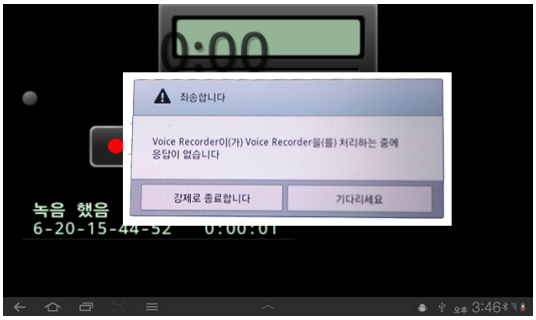

Fig. 10. Warning message disabling voice recorder app. when microphone is locked.

NFC authentication	Data structure
idx	integer
key	string
master_id	string
device_id	string
hash	string
date	datetime

Fig. 11. Data structure for NFC authentication

것이고, key는 암호화를 위한 키를 나타내는 것이고, master_id는 관리자가 여러 개 일 수 있으니 아이디를 설정해 놓는 것이다. device_id는 해당하는 기계 아이디를 설정하는 것이고, hash는 hash 값과 키 값을 대조하는 것이며, date 는 날짜와 시간을 기록하는 것이다.

kernel logging file의 데이터구조는 Fig. 12를 통해 알 수 있다. idx는 순서를 나타내는 것이고, master_sec은 어디 지역에서 Lock이 되었는지 장소를 나타내며, slave_device는 어떠한 장비가 접촉을 시도 했는지 알려주는 것이다. option은 어떠한 기능을 정지 혹은 실행 시켰는지 알려주는 것이며, data는 시간과 날짜를 기록하는 것이다.

NFC를 통한 접근제어는 입출 시 커널로깅을 기

kernel logging file	Data structure
idx	integer
master_sec	string
slave_device	string
option	integer
date	datetime

Fig. 12. Data structure of kernel logging file

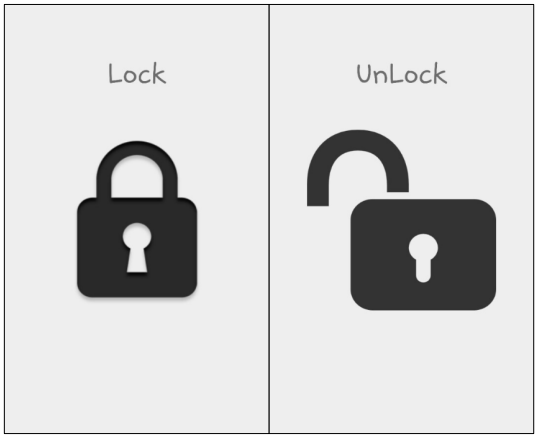

Fig. 13. Agent app. menu for security manager

록하여 관리자가 모니터링을 가능하게 한다.

태그 2개 중 1개를 start로 등록시키고, 나머지 1개를 end로 등록시킨다. 그림 13과 같이 핸드폰에 start 태그를 가져다대면 NFC 인증을 통해 Lock 상태가 된다. 그 후 end를 등록시킨 태그를 핸드폰에 가져다 대면 마찬가지로 NFC 인증을 통해 UnLock 상태가 된다.

5.2 벤치 마킹 시험

구현한 안드로이드 시스템은 임베디드 리눅스 커널 기반이지만 일반적으로 리눅스에서 사용하는 시스템 성능 측정 방법을 적용하기에는 무리가 있다. 따라서 안드로이드용 매크로 벤치마킹 어플리케이션인 AnTuTu[6]와 마이크로 벤치마킹을 위한 LMBench[7]를 Android에 맞게 수정한 것을 사용하여 본 논문에서 구현한 장치 통제 방법이 시스템에 어느 정도의 부하를 일으키는지를 시험하였다. 시험은 구현한 모듈이 커널에 적재 되지 않은 상태와, 구현한 모듈을 커널에 적재 시키고 장치 통제가 이루어지는 상태에서 비교를 진행 하였다. 그림 14는 AnTuTu 벤치마크를 통해 측정된 상대적 수치를 비교한 것이다. 보안 모듈이 적재 되지 않는 상황에서의 벤치마크의 Score가 근소하게 낮은 것을 볼 수 있다. 그림 15는 LSM 모듈의 적재 전과 적재 후의 메모리 지연속도를 LMBench를 통해 측정된 상대적 수치를 비교한 것으로 근소한 차이를 보인다. 그림 16은 LSM 모듈의 적재 전과 적재 후의 프로세스 지연속도를 LMBench를 통해 측정된 상대적 수치를 비교한 것이다. 그림 17은 LSM 모듈의 적재

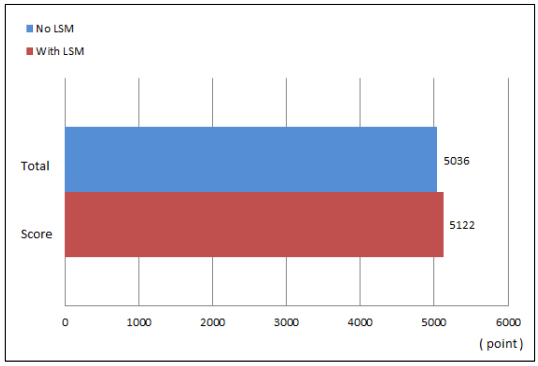

Fig. 14. AnTuTu Benchmark ranking(Lower is better.)

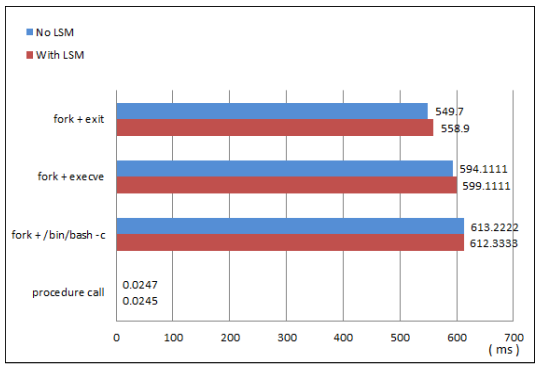

Fig. 15. LMBench memory latency(㎲)(Lower is better.)

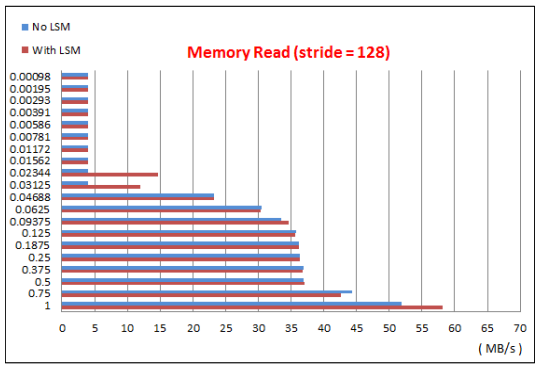

Fig. 16. LMBench process latency(㎲)(Lower is better.)

전과 적재 후의 시스템호출 지연속도를 LMBench를 통해 측정된 상대적 수치를 비교한 것이다. 그림 14~17 모두 LSM 모듈이 적재되지 않은 상황에서의 LMBench의 점수가 LSM 모듈을 탑재했을 때보다 매우 근소하게 낮은 것을 볼 수 있다.

이상의 벤치마킹 결과로 판단할 때, 장치 제어 보안 모듈 적재 전과 적재 후의 처리 시간에 따른 오버헤드는 무시할 수 있을 만큼 미미한 수준이라 볼 수 있다. 보안 모듈이 Open 시스템호출 경로에서 항상 동작 하지만 그림 18에서도 open/close 시스템 콜에 대해 소요된 시간은 보안 모듈이 적재 되어있는 상황이더라도 지연이 거의 차이가 없음을 알 수 있다.

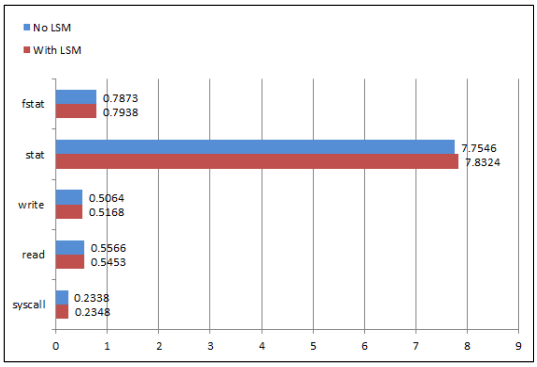

Fig. 17. LMBench syscall latency(㎲)(Lower is better.)

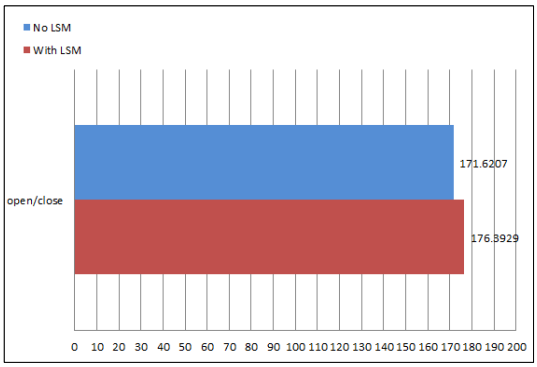

Fig. 18. LMBench open/close syscall latency (㎲)(Lower is better.)

VI. 결 론

본 논문을 통하여 모바일 안드로이드 장치를 리눅스 보안 모듈을 통하여 커널 수준에서 유연하게 통제할 수 있음을 프로토타입(prototype)의 구현을 통하여 확인하였다. 다시 말해 일반적인 MDM 솔루션과 다르게 커널 수준에서 장치를 제어하는 것이 가능함을 확인 하였다. 본 시스템은 Wi-Fi를 이용하지 않기 때문에 효율적으로 Wi-Fi 전파의 음영지역 문

제가 해결되며, 보안 구역별로 다르게 정책 테이블 변경만으로 유연하게 보안 정책 집행도 가능하다. 안드로이드 시스템의 가장 하위 수준의 커널에서 장치를 제어하기 때문에 사용자가 보안 통제를 우회 할 수 있는 여지가 최소화 된다. 또한 본 논문의 제안 방법을 이용할 경우 기존 시스템 전체의 성능에 대해 미치는 오버헤드가 거의 없기 때문에 무시할 정도라는 것도 확인하였다.

따라서 본 시스템은 중·소규모 조직이나 박물관, 갤러리, 항공기, 회의실, 공공 공간, 군 보안 구역 등의 조직에서 비교적 용이하게 다양한 보안 정책에 따라 유연하게 선택적으로 모바일 단말기의 장치를 보안 통제할 수 있다.

그러나 아직 몇 가지 향후 과제가 있다. 커널 수준에서 안전한 통제를 가능하게 하였지만, 단말기 루팅을 할 경우 무력화 될 가능성이 남아 있다. 또한 상용화를 위해서는 단말기의 NFC 인증과 출입자 데이터베이스를 함께 활용한 보안 정책 자동 설정 방법, 통제할 대상 장치(예로 스피커, 블루투스, Wi-Fi 등)의 추가 연구가 필요하다. 그리고 NFC 기능이 없는 단말기에 대해서는 전통적인 스티커 부착 혹은 단말기 자체의 지참을 제한하는 등의 자체 보안 정책 설정도 필요할 것이다.

References

〔1〕 Chris Wrigth et al., Linux Security Modules: General Security Support for the Linux kernel, August 2002.

〔2〕 Chris Runge, "SELinux:A New Approach to Secure System," Red Hat global resource library, September 2008.

〔3〕 Toshiharu Harada et al., "Task Oriented Management Obviates Your Onus on Linux," Linux Conference 2004.

〔4〕 Stephen Smalley and Robert Craig, "Android: Bringing Flexible MAC to Android," In Network & Distributed System Security Symposium (NDSS'13), 2013.

〔5〕 National Security Agency, Security-Enhanced Linux. http://www.nsa.gov/research/selinux

〔6〕 AnTuTu benchmark http://www.AnTuTulabs.com/AnTuTu-Benchmark

〔7〕 LMBench-Tools for Performance Analysis http://www.bitmover.com/lmbench

〔8〕 White Paper : An Overview of Samsung KNOX, April 2013.

〔9〕 Sun-Kuk Noh and Jae-Sub Kim, "A Study on the Improvement of Propagation Shadow Region for Mobile Communications," Journal of the Korean Institute of Comm. and Information Sciences, Vol.23, No.11T, November 1998.

〔10〕 Kang-Hyun Lee and Do-Sik Yoon, "Efficient Approach of MDM for Mobile Security," Review of the Korea Institute of Information Security and Cryptology, Vol.23, N.2, April 2013.

〈저자소개〉

배 희 성(Hee-sung Bae) 정회원
2015년 2월: 한서대학교 항공소프트웨어공학과 졸업
2015년 3월~현재: 공군 교육사령부 교육운영담당
〈관심분야〉 정보보호, 항공소프트웨어공학, 빅 데이터공학

김 소 연(So-yeon Kim) 정회원
2013년 3월: 한서대학교 항공소프트웨어공학과 입학
2017년 2월: 한서대학교 항공소프트웨어공학과 졸업
〈관심분야〉 정보보호, 항공소프트웨어공학

박 태 규(Tae-kyou Park) 종신회원
1980년10월: 경북대학교 컴퓨터공학과 학사
1989년 8월: 충남대학교 전산학과 석사
1996년 2월: 성균관대학교 정보공학과 박사
1981년 2월 ~ 1982년12월: 한국국방연구원 연구원
1982년12월 ~ 1992년 2월: 한국전자통신연구원 선임연구원
1992년 3월 ~ 현재: 한서대학교 항공소프트웨어공학과 교수
〈관심분야〉 보안운영체제, 항공정보보안(Drone Cyber Security)