디지털산업정보학회 논문지 제11권 제2호-2015년 6월
http://dx.doi.org/10.17662/ksdim.2015.11.2.047

변조 업데이트를 통해 전파되는 모바일 악성어플리케이션 모델 연구

주 승 환* · 서 희 석**

A Research on Mobile Malware Model propagated Update Attacks

Ju Seunghwan · Seo Heesuk

〈Abstract〉

The popularity and adoption of smart-phones has greatly stimulated the spread of mobile malware, especially on the popular platforms such as Android. The fluidity of application markets complicate smart-phone security. There is a pressing need to develop effective solutions. Although recent efforts have shed light on particular security issues, there remains little insight into broader security characteristics of smart-phone application. Now, the analytical methods used mainly are the reverse engineering-based analysis and the sandbox-based analysis. Such methods are can be analyzed in detail. but, they take a lot of time and have a one-time payout. In this study, we develop a system to monitor that mobile application permissions at application update. We had to overcome a one-time analysis. This study is a service-based malware analysis, It will be based will be based on the mobile security study.

Key Words : Mobile Security, Application Permission, Application Analysis, Mobile Malware

Ⅰ. 서론

안드로이드 기반 악성코드의 빠른 진화와 달리, 이들을 탐지하기 위한 모바일 보안 솔루션은 아직 초기 단계에 머물러 있다.

Dissecting Android Malware:Characterization and Evolution 연구[1]의 실험 결과에 따르면 1,260개의 악성코드에 대한 기존 모바일 안티바이러스 제품의 탐지율은 20.2%에서 79.6% 밖에 보이지 못하고 있다. 이러한 이유로 애플리케이션 유통 과정 전반에 걸쳐 모바일 악성코드의 빠른 탐지를 위한 다양한 보안 기법[2-4]들이 현재 연구되고 있다.

악성코드 유포자의 목적은 탐지되지 않은 채로 악성코드를 많은 단말에 감염시키는 것이다. 악성코드 탐지를 회피하면서 악성코드를 유포하기 위해 주로 사용되는 기법이 바로 애플리케이션 업데이트이다.

애플리케이션 업데이트는 악의적인 행동을 수행하는 로직을 애플리케이션 업데이트를 통해 단말로 배포하는 방식이다[5-6]. 즉, 사용자에 의해 처음 설치되는 애플리케이션에는 보안 솔루션의 탐지 대상이 되

* 한국기술교육대학교 컴퓨터공학과 박사과정
** 한국기술교육대학교 창의융합협동과정 부교수(교신저자)

는 악성코드가 포함되지 않으며, 악성코드를 획득하여 설치하기 위해 사용할 업데이트 로직을 갖는다. 현재 악성코드 유포를 위해 사용되는 업데이트 방식으로는 설치된 애플리케이션의 업데이트 버전을 다운받는 형식과 애플리케이션의 특정 부분만을 업데이트하는 방식이 있다.

Ⅱ. 행위기반 악성코드 분류

전자의 경우 업데이트에 대한 사용자의 승인이 필요하며, 업데이트 버전은 기존 애플리케이션의 resource나 asset파일로 저장된다[5]. 반면, 특정 부분만 업데이트 하는 방식은 사용자의 승인 절차 없이 진행될 수 있다[5].

이처럼 정상적인 서비스를 제공하면서 악성 로직을 수행하는 어플리케이션이 증가함에 따라 해당 어플리케이션이 비정상 로직을 수행하는지를 검증하는 방법론이 필요하게 되었다.

본 연구진은 어플리케이션이 갖는 요구권한(Permission)을 업데이트 시마다 기록하는 시스템을 개발하였고, 서비스를 제공하는데 필요하지 않은 요구권한을 검출하고자 하였다. 그 연구의 결과로 악성 어플리케이션이 갖는 요구권한 모델을 개발하였다.

악성코드의 행동 특성은 위 표와 같이 분류될 수 있고, 각각의 행동 특성들은 자신에 맞는 권한 [7]을 요구하고 이벤트를 처리한다. 악성코드들이 어떤 권한들을 주로 요구하는지 확인해보았다.

아래 그림은 정상적인 어플리케이션과 변조된 악성 어플리케이션[8]의 권한 비교한 것이다. 변조[10]

<표 1> 모바일 악성코드 행동 특성 분석

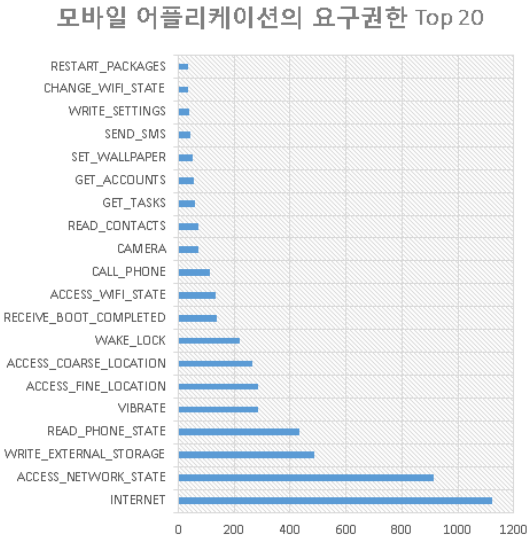
악성코드	BOOT	SMS	NET.	CALL	USB	PKG	BATT	SYS
ADRD	○		○	○				
AnserverBot	○	○	○		○		○	○
BaseBridge	○	○	○				○	○
BeanBot		○		○				
BgServ	○	○						
CoinPirate	○	○						
Crusewin	○	○						
DroidCoupon	○		○	○		○		
DroidDream	○						○	○
DroidKungFu	○						○	○
Endofday	○	○						
GamblerSMS	○							
Geinimi	○	○						
GGTracker	○	○					○	
jSMSHider						○		
Pjapps	○	○						○
Spitmo		○		○				

<표 2> 행동특성 분석을 위한 이벤트 분류

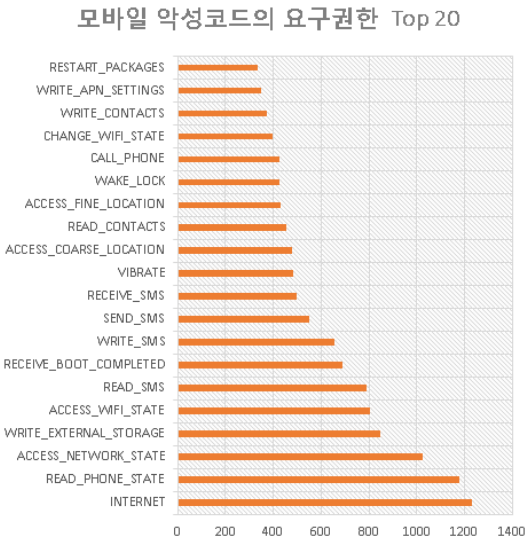
범주	이벤트
BOOT (Boot Completed)	BOOT_COMPLETED
SMS (SMS/MMS)	SMS_RECEIVED WAP_PUSH_RECEIVED
NET (Network)	CONNECTIVITY_CHANGE PICK_WIFI_WORK
CALL (Phone Event)	PHONE_STATE NEW_OUTGOING_CALL
USB (USB Storage)	UMS_CONNECTED UMS_DISCONNECTED
PKG (Package)	PACKAGE_ADDED PACKAGE_REMOVED PACKAGE_CHANGED PACKAGE_RESTARTED PACKAGE_REPLACED PACKAGE_INSTALL
BATT (Power/Battery)	ACTION_POWER_CONNECTED ACTION_POWER_DISCONNECTED BATTERY_LOW BATTERY_OKAY BATTERY_CHANGE_ACTION
SYS (System Events)	INPUT METHOD CHANGED SIG_STR SIM_FULL

된 악성어플리케이션은 정상적인 어플리케이션에 비해 단말 상태를 확인하기 위한 권한들이 요구됨을 확인[9]할 수 있었다.

단순히 이런 패턴 특성만 가지고 단말상태를 확인하기 위한 권한들이 악성코드에 사용되는 것이라고 판단할 수는 없기 때문에 다양한 악성코드들의 권한 요구 패턴을 확인해보고자 한다.



<그림 1> 모바일 어플리케이션의 주요 요구권한



<그림 2> 모바일 악성코드의 주요 요구 권한

Ⅲ. 권한요구 모니터링 시스템

3.1 권한 모니터링 방법

우리는 안드로이드 권한(Permission) 분석을 위한 시스템을 개발하였다. 그 시스템은 어플리케이션 설치 시에 어플리케이션의 요구 권한을 모니터링 한다.

여러 어플리케이션들이 요구하는 권한을 관리할 수 있도록 하였다. 안드로이드 어플리케이션은 신규 설치되거나 업데이트를 통해 설치 될 때 서비스를 위

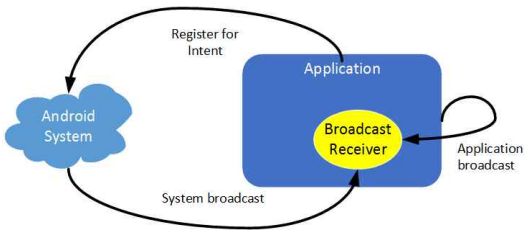
한 권한을 요구한다. 그렇기 때문에 안드로이드 어플리케이션이 설치되고 업데이트 될 때마다 AndroidManifest.xml 파일을 확인함으로써 요구 권한을 추적할 수 있다.

우선, 어플리케이션이 설치되거나 업데이트 되는 이벤트를 탐지하기 위한 방법이다 ;

BroadcastReceiver

Intent

안드로이드 시스템에는 Intent라는 메시지 처리 메커니즘이 있는데, Intent는 안드로이드 디바이스와 어플리케이션 컴포넌트 간의 상호작용을 위하여 사용된다.



<그림 3> Intent와 Broadcast Receiver의 흐름도

Intent는 시스템 간의 메시지 전달을 위해 사용될 수 있는데, 어플리케이션은 Intent(Broadcast intent)를 수신하기 위한 Broadcast Receiver를 등록하고 이를 통해서 반응할 수 있다. Broadcast Intent는 인터넷 접속 상태, 배터리 충전량 등의 시스템 이벤트에서도 사용할 수 있으며, 전화/SMS 수신과 같은 안드로이드 기본 이벤트에도 반응하는 어플리케이션을 개발할 수 있다. 이렇게 Intent와 Broadcast Receiver는 내부 시스템 또는 3rd Party 어플리케이션의 이벤트를 활용하는 이벤트 기반의 어플리케이션을 개발할 수 있다.

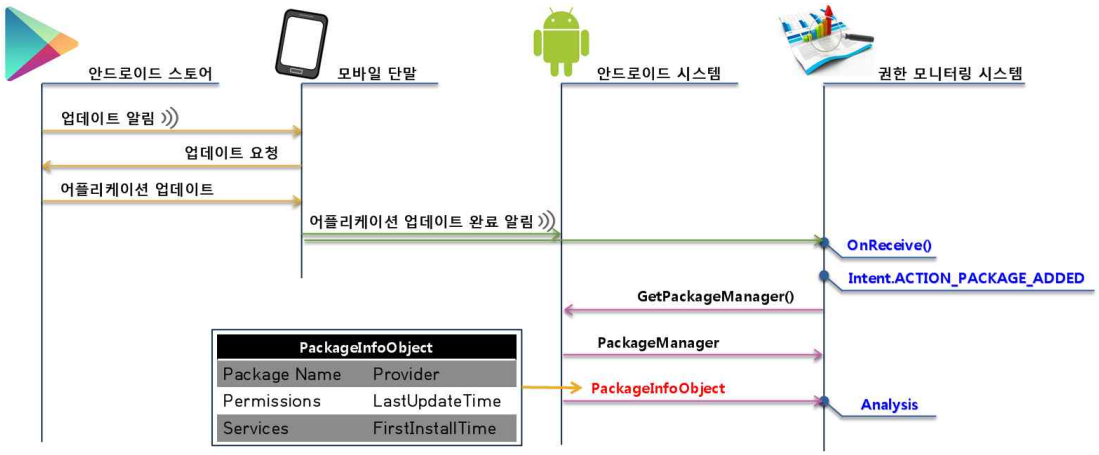
권한 모니터링 시스템은 *BroadcastReceiver* 객체를 상속받는다. 이는 다양한 안드로이드 시스템 이벤트에 접근 권한이 있다. *BroadcastReceiver* 가 안드로이드 시스템에서 발생하는 시스템 브로드캐스트를 수집하고, 필터링을 통해 어플리케이션의 설치나 업데이트를 확인한다. 그리고 필터에 적합한 상황일 때 브로드캐스트 메시지로부터 정보를 취득할 수 있다.

3.2 권한 모니터링 시스템 구조

권한 모니터링 시스템은 업데이트 이벤트를 탐지하기 위해서 안드로이드 시스템의 Intent 객체를 사용한다. 어플리케이션에서 이벤트가 발생했을 때, onReceive() 함수를 호출하면 Intent 객체가 이벤트 상태 값과 함께 전달된다. 요구 권한 모니터링 시스템은 이 intent 객체를 참조하는데, Intent.ACTION_PACKAGE_ADDED 라는 이벤트를 탐지해야 한다.

안드로이드 OS에는 다양한 시스템 어플리케이션들이 설치되어 있다. 이 시스템 어플리케이션 중에는 사용자가 직접 어플리케이션을 설치하기 위한 패키지 매니저도 포함되어 있다. 이 패키지 매니저는 어플리케이션의 설치 혹은 업데이트를 성공적으로 완료되었을 때 모든 서비스 어플리케이션이 탐지할 수 있도록 브로드캐스트 메시지를 발송한다. 이 모니터링 시스템은 이 브로드캐스트 메시지를 수신할 수 있는 브로트캐스트 리시버를 상속받아 작동하기 때문에 패키지 매니저가 발송한 메시지를 수신 받을 수 있다. 평시에는 브로드캐스트 메시지를 수집하여 그 중에서 어플리케이션의 추가 혹은 업데이트에 대한 정보만 필터링한다. 필터링한 메시지로부터 발생한 어플리케이션에 대한 정보를 확인한다.

패키지 매니저가 발송한 메시지를 이용해 다양한 정보를 획득할 수 있는데, 이 모니터링 시스템은 어

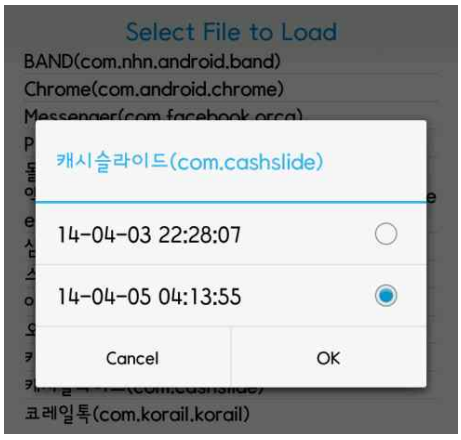


<그림 4> 요구권한 모니터링 시스템의 시퀀스 다이어그램

플리케이션의 패키지명, 설치 경로, 실행 권한, 액티비티 정보 등 어플리케이션의 다양한 정보를 수집한다. 모니터링 시스템은 해당 패키지명과 메시지를 수신한 시간으로 파일을 생성한다. 이 파일에는 어플리케이션의 레이블 명과 실행 권한이 저장된다. 저장된 파일은 모니터링 시스템 외의 타 어플리케이션은 접근할 수 없도록 하였다.

어플리케이션 권한 모니터링 시스템의 작동 원리는 다음과 같다. onReceive() 의 첫 번째 인자인 Context 객체를 이용한다. Context의 getPackageManager() 메소드를 통해 PackageManager 객체를 얻어온 후, PackageManager 로부터 PackageInfo 객체를 획득한다. PackageInfo 객체는 각종 패키지 정보 및 어플리케이션 요구 권한 정보도 가지고 있다. 이를 이용해 방금 설치된 어플리케이션의 요구 권한을 조회할 수 있다. 모니터링 시스템을 실행했을 경우 지금까지 수집된 파일 목록을 출력한다. 출력된 파일 목록은 패키지명과 메시지 수신 시간으로 조합되어 있어 해당 어플리케이션이 설치된 시각과 요구하는 권한, 이전에 요구했던 권한 등 다양한 정보를 확인할 수 있다.

모니터링 시스템이 설치된 이후에 설치된 어플리케이션의 경우 최초 설치부터 업데이트와 현재까지 사용했던 모든 권한에 대한 정보를 보관하여 관리 및 분석이 가능하다. 사용자는 본 시스템을 이용하여 각 어플리케이션의 요구권한 추이를 확인하고 어플리케이션을 관리할 수 있다.



<그림 5> 어플리케이션 요구권한 변경 이벤트 리스트

Ⅳ. 어플리케이션 요구권한 패턴

<표 5> DroidDream 악성코드 권한 패턴

DroidDream
ACCESS_NETWORK_STATE
ACCESS_WIFI_STATE
CHANGE_WIFI_STATE
READ_HISTORY_BOOKMARKS
WRITE_HISTORY_BOOKMARKS
INSTALL_SHORTCUT
INTERNET
KILL_BACKGROUND_PROCESSES
READ_CONTACTS
READ_LOGS
READ_PHONE_STATE
RESTART_PACKAGES
WRITE_CONTACTS

DroidDream 악성코드의 패턴은 북마크 히스토리와 Wi-Fi 연결 정보에 대한 상태를 접근하는 패턴을 가지고 있음을 확인할 수 있다. DroidDream 악성코드는 대부분 구글 Play 마켓으로 전파되었다는 것이 중요 특징이다.

<표 6> Genimi 악성코드 권한 패턴

Genimi pattern:
ACCESS_COARSE_LOCATION
ACCESS_FINE_LOCATION
CALL_PHONE
READ_CONTACTS
MOUNT_UNMOUNT_FILESYSTEMS
READ_PHONE_STATE
SEND_SMS
SET_WALLPAPER
WRITE_CONTACTS
WRITE EXTERNAL_STORAGE.
INTERNET

Geinimi 악성코드는 모바일 트로이목마로 알려져 있다. Geinimi 악성코드 외에도 트로이 목마는 파일 시스템과 마운트 등과 같이 일반적인 어플리케이션은 잘 요청하지 않는 권한들을 요청하는 경우가 있다.

<표 7> GoldDream 악성코드 권한 패턴

GoldDream pattern:
ACCESS_NETWORK_STATE
READ_PHONE_STATE
ACCESS_WIFI_STATE
WRITE_EXTERNAL_STORAGE
ACCESS_COARSE_LOCATION
ACCESS_FINE_LOCATION
RECEIVE_ / SEND_ / READ_ SMS
SEND_SMS
CALL_PHONE
PROCESS_OUTGOING_CALLS
PACKAGES / INSTALL_PACKAGES
RECEIVE_BOOT_COMPLETED
WRITE EXTERNAL_STORAGE.
INTERNET

GoldDream 악성코드도 Geinimi 악성코드 패턴처럼 고유한 패턴을 갖는다. 시스템이 부팅되었음을 확인하고 모바일 단말의 상태에 접근하며 패키지를 추가로 설치할 수 있는 권한을 요구한다.

<표 8> Pjapps 악성코드 권한 패턴

Pjapps pattern:
ACCESS_NETWORK_STATE
READ_PHONE_STATE
ACCESS_WIFI_STATE
DISABLE_KEYGUARD
RECEIVE_SMS / SEND_SMS
WRITE_EXTERNAL_STORAGE
INTERNET

초기의 Pjapps 트로이목마는 세 가지의 권한이 추가되는 악성코드였다. 공격자에 의해 더욱 변조되면서 더 많은 권한을 요구하게 되었는데, 외부 메모리에 접근하고 사용자 SMS을 제어하는 요구 권한이 추가 된 형태로 발전하였다.

<표 9> adSMS 악성코드 권한 패턴

adSMS pattern:
RECEIVE_ / SEND_ / READ_ SMS
DEVICE_POWER
WRITE_APN_SETTINGS
ACCESS_NETWORK_STATE
BROADCAST_PACKAGE_REMOVED
ACCESS_WIFI_STATE
CHANGE_WIFI_STATE
WAKE_LOCK
WRITE_EXTERNAL_STORAGE
READ_PHONE_STATE
KILL_BACKGROUND_PROCESSES
INTERNET

adSMS 악성 어플리케이션은 모바일 단말의 SMS를 제어하는 권한이 추가적으로 요청된다. APN 설정을 통해 모바일 통신망에 대한 접근도 하며, 브로드캐스트 메시지에 대한 권한도 요구하는 것을 알 수 있다. 잠금화면을 해제할 수 있으며 백드라운드 프로세스에 대한 접근도 하는 등의 악성코드 로직 수행을 위한 권한들이 요구됨을 확인할 수 있다.

이러한 악성코드들의 권한 패턴을 확인한 결과, INTERNET 권한은 거의 모든 악성코드들이 공통적으로 요구하였고, SMS 제어와 관련된 권한과 외장메모리 접근에 대한 권한 요구가 많음을 확인할 수 있었다.

그리고 모바일 통신 상태를 조회하고 변경하고 다른 프로세스를 종료하는 권한을 갖는 등의 악성코드도 확인하였다.

<표 10> JimmRussia 악성코드 권한 패턴

JimmRussia pattern:
ACCESS_NETWORK_STATE
RECEIVE_SMS / SEND_SMS
WRITE_EXTERNAL_STORAGE
INTERNET

Ⅴ. 결론

최근 권한 변조로 인한 어플리케이션 악성코드가 많이 발견됨에 따라 보안성 확보를 위해 어플리케이션의 요구 권한을 모니터링 시스템을 연구하였다.

사용자는 본 시스템을 이용하여 각 어플리케이션의 요구권한 추이를 확인하고 어플리케이션을 관리할 수 있다. 어플리케이션 모니터링 시스템은 어플리케이션을 단말에 최초로 설치하는 과정에 설치 날짜와 요구하는 권한을 저장하도록 한다. 그 후 어플리케이션이 업데이트가 수행될 때, 업데이트 이후의 어플리케이션의 요구 권한과 업데이트 날짜를 획득하여 기존에 요구했던 권한과 비교를 수행한다.

어플리케이션 권한 요청 모니터링 시스템이 설치된 단말은 어플리케이션의 최초 설치부터 업데이트 시점과 현재까지 사용했던 모든 요구 권한에 대한 정보를 보관하여 관리 및 분석이 가능하다. 이로써 이 모니터링 시스템은 무분별한 권한을 요구하는 어플리케이션의 패턴분석과 악성코드가 요구하는 안드로이드 권한 등의 다양한 연구에서 사용될 수 있다.

우리는 모바일 악성코드들의 권한들을 연구하여 권한 모델 및 패턴을 도출하고자 하였다. 향후 연구에서는 본 시스템을 이용하여 어플리케이션 서비스와 요구 권한의 관계에서 패턴을 찾고 이를 이용하여 어플리케이션을 검증할 수 있는 시스템으로 고도화 하고자 한다.

감사의 글

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2010- 0021951).

본 논문은 교육부 및 한국연구재단의 BK21 플러스 사업 (미래기반 창의인재양성형)으로 지원된 연구임.

참고문헌

[1] Y. Zhou and X. Jiang, “Dissecting Android Malware: Characterization and Evolution,” Proc 33rd IEEE Symp Security and Privacy, 2012.

[2] L. K. Yan and H. Yin, “DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis,” Proc 21st USENIX conf. Security Symp., Security, 2012.

[3] R. Xu, H. Saidi, and R. Anderson, “Aurasium: Practical Policy Enforcement for Android Applications,” 21st USENIX Security Symp. USENIX, 2012.

[4] M. Chandramohan, and H. B. K. Tan, “Detection of Moblie Malware in the Wild,” Comput., vol, 45 no. 9, Sept. 2012.

[5] A. Moser, C. Kruegel, and E. Kirda, “Exploring Multiple Execution Paths for Malware Analysis,” Proc. IEEE Symp. Security Privacy, SP, 2007, pp. 231-245.

[6] W. Enck et al., “A Study of Android Application Security,” Proc. 20th USENIX Conf. Security, Security, 2011.

[7] Google. <permission>. Available: http://developer.android.com/guide/topics/manifest/permission-element. html.

[8] Google. Android market developer program policies. Available: http://www.android.com/us/developer-content-policy.html.

[9] F-Secure, “MOBILE THREAT REPORT, " 2014.

[10] 최희식, 조양현, “사물인터넷 보안 문제제기와 대안,” 디지털산업정보학회 논문지, 11권 1호, 2015, pp. 69-78.

[11] 신현조, 이경동, 박태형, “인적 및 직무특성과 보안교육 이수율 및 사이버테러 대응과의 연관성 분석,” 디지털산업정보학회 논문지, 10권 4호, 2015, pp. 97-107.

■ 저자소개 ■

주 승 환
Ju Seunghwan

2011년 3월~현재 한국기술교육대학교 컴퓨터공학과 (박사과정)
2011년 2월 한국기술교육대학교 컴퓨터공학과 (공학석사)
2009년 8월 한국기술교육대학교 컴퓨터공학과 (공학사)

관심분야 : 모바일 보안, 보안제품 평가
E-mail : judeng@kut.ac.kr

서 희 석
Seo Heesuk

2005년 3월~현재 한국기술교육대학교 컴퓨터공학과 (부교수)
2005년 2월 성균관대학교 전기전자 및 컴퓨터공학과 (공학석사)
2009년 8월 성균관대학교 전기전자 및 컴퓨터공학과(공학사)

관심분야 : 네트워크보안, 보안시뮬레이션, USN
E-mail : histone@kut.ac.kr

논문접수일: 2015년 5월 30일
수 정 일: 2015년 6월 5일
게재확정일: 2015년 6월 9일