JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 20, No. 1 : 87~93 Jan. 2016

안드로이드 기반 애플리케이션의 시스템 수준 접근 권한에 대한 선택적 관리

정종문[1] · 이훈[2*] · 황민태[2]

Selective Management of System-level Access Permission in Android-based Application

Jongmun Jeong[1] · Hoon Lee[2*] · Mintae Hwang[2]

[1]Department of Eco-friendly Offshore Plant FEED Engineering, Graduate School of Changwon National University, Changwon, Korea
[2*]Department of Information & Communications Engineering, Changwon National University, Changwon, 51140, Korea

요 약

본 논문에서는 안드로이드의 보안체계 강화를 위한 애플리케이션 권한의 선택적 관리방안에 대해 새로운 방법을 제안하였다. 먼저 기존의 안드로이드의 보안 체계에 대해 분석하고 발생 가능한 취약점을 도출하였다. 이어서 도출된 취약점을 해결하기 위한 방안으로서 애플리케이션 권한을 선택적으로 관리하는 툴을 구현하였다. 이 툴은 애플리케이션을 설치할 때 반드시 허용해야 하는 애플리케이션 권한을 선택적으로 허용하여 필요한 권한만 가지게 하는 기능을 가지고 있다. 구현한 관리 도구를 이용한 실험을 통하여 개발된 툴이 안드로이드의 시스템 레벨 보안 강화에 도움이 됨을 입증하였다.

ABSTRACT

In this paper, we propose a new method to enhance an android security by exploiting a selective management of application permission. To that purpose, we analyze behavior of the current android security, via which we draw out possible vulnerabilities. After that, we develop a tool to implement the selective management of the application permission, witch has a function to give a permission selectively for the application when we install a new application. Via experiment we show validity of the developed tool in solving the drawn vulnerability in the current android security.

키워드 : 안드로이드 보안, 권한관리, 선택적 관리, 시스템 수준 보안

Key word : Android security, Permission management, Selective Management, System-level security

접수일자 : 2015. 09. 17 심사완료일자 : 2015. 09. 25 게재확정일자 : 2015. 10. 13

*** Corresponding Author** Hoon Lee (E-mail:hoony@cwnu.ac.kr, Tel:+82-55-213-3833)
Department of Information & Communications Engineering, Changwon National University, Changwon, 51140, Korea

Open Access http://dx.doi.org/10.6109/jkiice.2016.20.1.87 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

2015년 1분기를 기준으로 안드로이드(Android)를 사용하는 스마트폰의 댓수는 약 12억 개로 전체 스마트폰 운영체제의 79%를 점유하고 있다[1]. 또한 현재까지 밝혀진 약 7백만 개의 모바일(Mobile) 멀웨어(Malware, Malicious software) 중 약 110만 개의 새로운 모바일 멀웨어가 2015년 1분기에 생겼으며, 이는 2014년 4분기의 새로운 모바일 멀웨어 보다 약 49%가 증가한 것이다[2].

이와 같이 시장 점유율이 크고 2015년에 들어서 급격하게 증가하고 있는 안드로이드 기반 모바일 애플리케이션에 대한 멀웨어에 대응하는 방안으로서 애플리케이션의 접근권한을 관리할 필요가 있다.

스마트폰 애플리케이션의 접근권한을 관리하는 방안으로 iOS는 모든 애플리케이션을 아이튠스(iTunes)로만 배포하여 보안상 생기는 문제에 즉각적으로 대응하고 있다. 아이튠스에서는 애플(Apple)에서 검증한 애플리케이션만 배포한다. 거기에 보안상의 문제가 있는 애플리케이션이 배포되는 것을 막기 위해 최신버전만 설치 할 수 있으며 다운그레이드가 불가능하게 되어있다. 한편 안드로이드 6.0 미만의 환경에서 작동되는 애플리케이션의 경우에는 애플리케이션의 개발자가 요청하는 권한을 사용자가 일괄로 허용해야만 설치 할 수 있게 되어있다. 다만 본 연구의 수행 중에 새로이 출시된 6.0 이상의 환경에서 작동되는 애플리케이션은 사용자가 허용할 권한을 선택해서 설치 할 수 있다.

본 연구에서는 안드로이드 환경에 집중해서 다루기로 한다. 이 환경의 경우 일괄적으로 사용권한을 허용하는 방식은 보안 침해에 악용하기 쉬운 방식이고, 특히 한국에서의 안드로이드 개발 환경에서는 실제로 사용하지 않더라도 주로 사용되는 권한들을 전부 요청하는 추세이다. 따라서 안드로이드 6.0 미만 환경을 가진 사용자들은 모든 권한을 무조건적으로 허용하지 않고 선택적으로 관리하는 것이 필요하다[3].

안드로이드 보안에 대한 기존의 연구 동향에 대해서 대표적인 몇 가지만 소개한다. 먼저, 김동민 등은 사용자 정책 기반의 안드로이드 권한 모델을 제안하였다[4]. 이 논문에서는 사용자가 권한 정책을 수립하는 사용자 중심의 안드로이드 권한 모델을 제시하였으며, 문자 메시지 전송에 필요한 권한을 제어하는 프로토타입을 구현하였다.

김영동 등은 안드로이드 권한과 브로드캐스트 인텐트(Intent) 매커니즘의 사용 현황 및 보안의 취약성에 대해서 분석하였다[5]. 이 논문에서는 권한 기반 모델에 사용자의 이해 부족과 개발자의 과도한 권한 요청으로 인한 취약점과 컴포넌트(Component)간의 통신 수단인 인텐트가 사용되는데 반해 이에 대한 보안 보안정책이 없다는 점을 제시하였으며, 권한과 브로드캐스트 리시버(Broadcast Receivers)의 사용 현황을 분석하여 악성 프로그램을 분석하는 안드로이드 플랫폼 수정 방안을 제시하였다.

한편, 정종문 등은 안드로이드의 시스템 수준 보안 취약점의 대응 방안으로 애플리케이션의 권한을 관리하는 방법을 제시하였다[6]. 그러나, [6]에서는 애플리케이션의 권한을 선택적으로 관리하는 구체적인 방안이 제시되지 못했다.

이와 같은 배경에서 본 연구에서는 [6]의 연구에서 더 나아가서 [4]와 [5]의 연구와 다른 방향으로 안드로이드기반의 모바일 애플리케이션에 대한 권한관리를 선택적으로 수행하는 방안을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 안드로이드의 보안 체계에 대해 살펴보았으며, III장에서는 안드로이드용 애플리케이션 권한 관리 툴을 구현하고 소개한다. 그리고 IV장에서는 시중에 출시된 유명 애플리케이션들의 권한을 평가하고 그에 적합한 활용을 보여주며, 마지막으로 V장에서 본 연구의 결론을 기술한다.

Ⅱ. 안드로이드의 보안체계

2005년에 구글은 안드로이드사를 인수했다. 그리고 2007년 11월 5일에 구글 외 78개의 기업 및 단체는 개방형 휴대 전화 동맹(OHA, Open Headset Alliance)을 설립함과 동시에 리눅스 커널(Linux Kernal) 2.6.25를 기반으로 한 안드로이드 1.0 알파를 공개하며 안드로이드를 휴대용 장치를 위한 운영체제(OS, Operating System)로 무료 공개했다[7].

안드로이드는 리눅스 커널을 기반으로 하여 모바일 환경에 적합하게 수정된 보안체계를 가지고 있다. 안드로이드의 보안 체계는 리눅스 커널 기반의 보안 기능과 안드로이드의 특성에 걸맞은 고유의 보안기능으로 나

눌 수 있다.

2.1. 리눅스 커널 기반의 보안 기능

2.1.1. ID

설치되는 애플리케이션에는 각각 고유의 리눅스 사용자 ID(User Identification)와 그룹 ID가 부여된다. 이는 다른 ID를 가진 애플리케이션에 접근하는 것을 막아준다[8, 15]. 그러나 예외적으로 동일한 개발자 서명을 가지고 있는 애플리케이션의 경우 애플리케이션의 정보를 담고 있는 앱 매니페스트(App Manifest) 파일인 AndroidManifest.xml의 android:sharedUserID 속성 등을 지정하면 동일한 사용자 ID를 갖게 하는 것도 가능하다[9]. 안드로이드의 파일 시스템 헤더파일인 android_filesystem_config.h에서 ID 번호 지정 방식을 알 수 있다.

2.1.2. 파일 권한

파일권한에는 각 디렉터리와 파일별로 소유자(Owner), 그룹, 소유자도 그룹도 아닌 사용자(Others)에 대한 읽기(Read), 쓰기(Write), 실행(Execute)의 조합으로 이루어진 총 9개가 있다. 안드로이드는 마켓에서 받은 APK(Android application package) 파일로 애플리케이션을 설치하는데, 이 파일에는 달빅 가상 머신에서 사용하는 dex(Dalvik Executable format)파일이나 안드로이드 런타임(ART, Android Runtime)에서 사용하는 art 파일 또는 둘 다 사용 가능한 AOT(Ahead-of-time) 파일, 리소스 파일, 매니페스트 파일, 인증서(CERT, Certificate) 파일 파일 등으로 구성되어 있다.

한편 위의 파일 가운데서 설치 시에 검증되는 매니페스트 파일과 CERT를 제외하면, apk 파일과 dex, art, AOT 파일은 소유자에게 읽기 및 쓰기 권한을 주고 나머지 사용자들에게는 읽기 권한만을 준다. 리소스 파일은 소유자에게는 모든 권한을 주되 나머지 사용자들에게는 읽기와 실행 권한을 준다[8]. 그러므로 루팅(Rooting)을 하지 않는 이상 파일 권한에 의해 다른 애플리케이션의 파일에 영향을 끼칠 수 없다.

2.2. 안드로이드의 보안 기능

2.2.1. 컴포넌트

컴포넌트는 애플리케이션에서 상호작용 할 수 있는 식별 가능한 일부분이다. 컴포넌트의 종류로는 액티비티(Activity), 서비스(Service), 컨텐트 프로바이더(Content Provider), 브로드캐스트 리시버, 인텐트 등이 있다.

액티비티는 화면을 구성하는 가장 기본적인 컴포넌트이고, 브로드캐스트 리시버는 다른 곳에서 전달되는 방송 메시지를 받기 위한 것이며, 콘텐트 프로바이더는 데이터를 공유하기 위한 것이고, 서비스는 화면 구성과 상관없이 백그라운드에서 동작하는 것이며, 마지막으로 인텐트는 컴포넌트간의 상호작용수단이다[10].

안드로이드에서는 ID에 의해 접근 할 수 없게 된 다른 애플리케이션의 데이터를 컨텐트 프로바이더와 인텐트 등에 의해 허용만 된다면 공유 받을 수 있다. 예를 들면 AndroidManifest.xml의 액티비티에서 android:export가 true가 될 경우 그 애플리케이션의 액티비티는 전할 수 있게 되는 것이다[9].

2.2.2. 안드로이드 런타임

안드로이드는 자바를 사용하기 때문에 가상머신의 사용이 필수적인데, 자바 가상머신은 라이선스 문제가 있어서 안드로이드 5.0 이전까지 달빅 가상머신을 따로 개발해 사용하였다.

하지만 달빅 가상머신은 부하가 심한 것 등 여러 가지 단점이 있어 새로운 가상머신인 안드로이드 런타임을 개발해 4.4부터 시험 운영을 하여 5.0부터 달빅 가상머신을 폐지하였다[11-13].

하나의 ID를 가진 애플리케이션과 그 컴포넌트들은 하나의 가상머신에서 동작된다. 이는 애플리케이션끼리는 서로 데이터에 직접적으로 접근 할 수 없고, 컴포넌트를 통한 명시적으로만 접근으로만 데이터를 공유할 수 있다는 것이다[8].

2.2.3. 애플리케이션 서명

안드로이드 애플리케이션은 개발자의 개인키(Private key)로 서명되어야만 마켓에 올릴 수 있고 설치가 가능하다. 하지만 안드로이드에서는 소스코드에 대한 검증이 없으므로 멀웨어들도 얼마든지 서명해서 마켓에 올릴 수 있다. 따라서 이 때 사용자들은 검증된 개발자가 서명 했는지를 확인해야 한다[14].

2.2.4. 애플리케이션 권한

안드로이드에서 다른 애플리케이션의 컴포넌트나

데이터에 접근하기 위해서는 애플리케이션 권한이 필요하다. 애플리케이션 권한은 알람을 울릴 수 있게 하거나 진동을 울리게 하는 등 사소한 권한부터 주소록을 읽고 문자 메시지를 보내는 등의 위험한 권한까지 종류가 아주 많다.

권한을 얻기 위해서는 AndroidManifest.xml에서 user-permission에 선언해야 한다[15]. 이는 설치 시에 검증하는 파일이므로 사용자는 애플리케이션 설치 시에 개발자가 선언해둔 모든 권한을 하나도 빠짐없이 허용해야만 설치 할 수 있다. 그래서 어쩔 수 없이 사용자는 필요한 애플리케이션을 설치하기 위해 요청해둔 권한에 상관없이 허용하게 된다.

그러나 이를 이용해서 정상적인 애플리케이션으로 속여 설치되는 멀웨어들이 갈수록 증가하고 있다. 특히 한국에서는 정상적인 애플리케이션도 과도한 권한을 요청하는 추세이므로 이에 대비하여 권한을 선택적으로 허용하는 기능이 필요하다.

본 연구를 수행하는 과정에 새로이 밝혀진 사실로서 안드로이드 6.0부터의 애플리케이션은 권한을 선택적으로 허용하도록 시스템이 바뀌었지만, 6.0 이전에 만들어진 애플리케이션들을 6.0에 적용할 경우 설치 후 권한을 해제 할 수 있게 된다[3]. 이 기능은 본 연구에서 구현하고자 하는 방식과 유사한 기능이지만, 2015년 9월의 통계에 따르면 4.4 이하의 버전을 사용하는 안드로이드 사용자가 약 79%일 정도로 구버전의 사용자가 많고, 특히 이번에 구현한 방식을 활용 할 수 있는 버전인 4.3부터 5.1까지의 사용자는 약 65%인만큼 선택적 권한 허용 기능은 현재로서도 필요하다고 할 수 있다[16].

Ⅲ. 관리 툴의 구현

본 연구에서 대상으로 하는 안드로이드의 버전은 4.3부터 4.4.1까지와 앱 운영(App Ops) 도구가 삭제되지 않고 화이트리스트에서도 제거 되지 않는 4.4.2부터 5.1인 경우를 중심으로 하는 앱 권한 운영(App Permission Ops)이다.

앱 권한 운영은 버전을 체크하는 기능과 안드로이드의 숨겨진 도구인 앱 운영을 사용하는 기능이 있다. 앱 운영은 안드로이드에서 관리 할 수 있게 허용한 권한들을 위치(Location), 개인(Personal), 메시지(Message), 기기(Device)로 나눠 애플리케이션 권한을 관리하는 기능을 가지고 있다. 위치는 위치정보, 개인은 주소록과 같은 개인정보, 메시지는 메시지를 읽거나 쓰는 것이고, 기기는 기기의 설정을 수정하는 것이다.

아래의 그림 1은 앱 운영의 개인 탭이다. 개인 탭에는 개인정보와 관련된 권한을 가진 애플리케이션들의 목록이 나와 있으며, 가장 최근에 권한을 사용한 애플리케이션일수록 위에 위치해 있다. 옆의 숫자는 몇 분 전에 사용했는지 알려주는 것이다. 각 탭은 좌우로 드래그 하여 바꿀 수 있다. 본 연구에서는 구글 갤럭시 넥서스 7 2013에 안드로이드 4.3을 설치한 것을 가상머신으로 만들어서 테스트하였다.

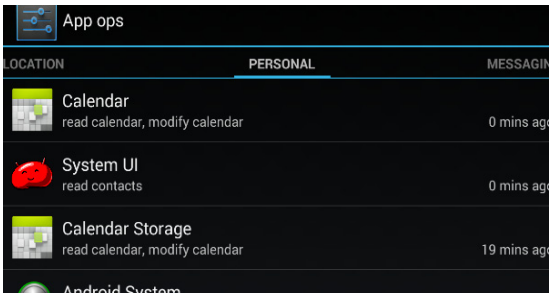

Fig. 1 Personal tab of App operations

애플리케이션을 누르면 그 애플리케이션이 가진 권한 중 안드로이드에서 해제 할 수 있게 설정한 위치, 개인, 메시지, 기기관련 권한들의 목록을 보여준다. 하지만 안드로이드에서 자체적인 문제로 한 번도 사용하지 않은 권한은 그 권한을 사용 할 때 까지 인식되지 않을 수 있다.

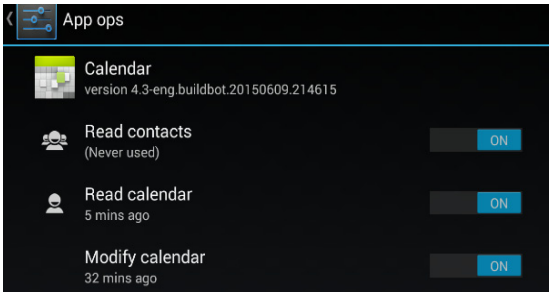

Fig. 2 Permission of calendar application

그림 2는 애플리케이션중의 하나인 캘린더(Calendar) 애플리케이션을 확인한 것이다. 그림 2로부터 주소록 읽기(Read contracts)는 한 번도 사용되지 않은 권한이고, 캘린더 읽기(Read calendar) 권한은 5분 전에 사용되었으며 캘린더 수정(Modify calendar) 권한은 32분 전에 사용되었다는 것을 알 수 있다. 옆의 On/Off 스위치는 그 권한들을 허용/비허용 하는 스위치이다.

이렇게 애플리케이션 권한을 선택적 관리하는 것으로 사용자의 위치정보, 개인정보, 메시지, 기기정보를 보호 하여 스마트폰을 안전하게 사용 할 수 있다. 또한, 필요 없는 기능의 관련 권한을 허용하지 않음으로써 애플리케이션을 선택적으로 사용할 수도 있다.

Ⅳ. 실험 및 결과

제 III장에서 개발한 툴을 사용하여 한국의 상용 애플리케이션의 권한을 파악하고 그에 대한 분석과 실험을 하기로 한다.

알람몬(AlarmMon)은 말랑 스튜디오(Malang Studio)에서 제작한 알람 애플리케이션이다. 알람몬은 약 19개의 권한을 가지고 있는데 그 중에서 위치, 개인, 메시지, 기기관련 권한은 위치(Location), 위에 그림(Draw on top), 소식 알림(Post notification), 오디오 녹음(Record Audio)인데 그림 3이 이것을 나타내고 있다.

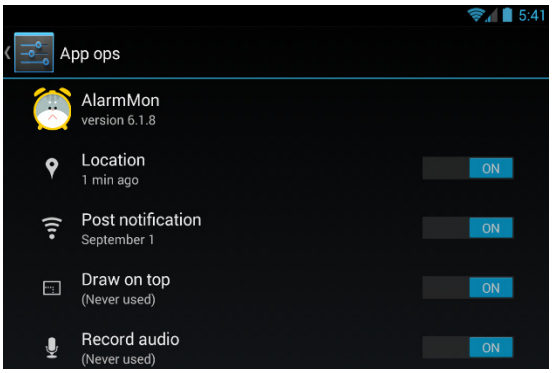

Fig. 3 Permission of AlarmMon

그림 3에서 보는 바와 같이 네 권한 전부 알람과는 관계가 적다. 그래서 네 권한을 전부 해제하더라도 알람을 사용 할 수 있다. 또한 위치를 추적당하거나 목소리를 녹음당할 염려도 없어진다.

이번에는 직접적으로 확인 가능한 권한을 허용하지 않는 실험을 하기로 한다. 아래의 그림 4와 같이 알람을 켜면 그 위에 소식 알림 권한을 사용한 시계모양 아이콘이 뜬다.

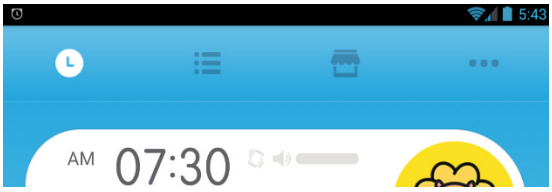

Fig. 4 Permission for the post notification of AlarmMon

아래의 그림 5는 소식 알림 권한을 해제한 것이다.

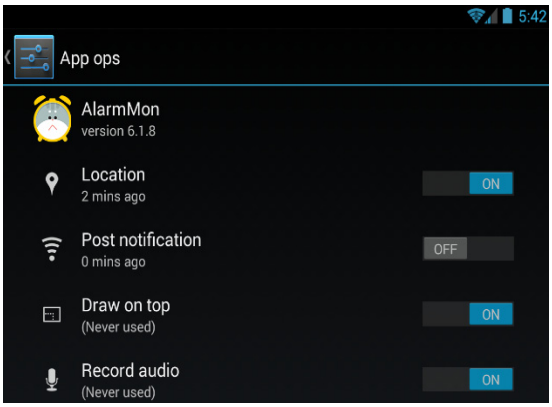

Fig. 5 Non-permission of the post notification of AlarmMon

아래의 그림 6는 소식 알림 권한을 해제해서 아이콘이 뜨지 않게 한 것이다.

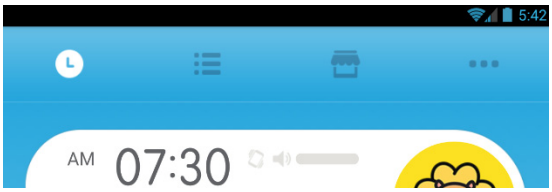

Fig. 6 Non-permission of AlarmMon

이와 같이 선택적 권한 관리를 사용하면 애플리케이션 레벨에서의 보안을 더욱 강화 할 수 있다.

Ⅴ. 결 론

본 연구에서는 안드로이드의 보안체계와 애플리케이션 권한의 선택적 관리 기술에 대해 연구하였다.

이를 위하여 안드로이드의 버전을 체크하는 기능과 애플리케이션의 권한을 선택적으로 관리할 수 있는 툴을 구현하였고, 개발된 툴을 이용하여 한국의 상용 애플리케이션의 과다한 권한 요청에 대한 분석과 그를 관리하는 실험을 실시하였다.

본 연구의 기대효과로서 두 가지를 들 수 있는데, 먼저 개발된 툴을 이용하면 보안에 민감한 개인정보를 보호할 수 있음을 알 수 있다. 그리고 애플리케이션의 동작에 반드시 필요하지 않는 기능을 사용하지 않게 할 수도 있어서 애플리케이션의 동작에 있어서 효율이 향상됨을 알 수 있다.

향후에는 앱 운영 도구가 설치되지 않은 안드로이드 기기라도 선택적으로 애플리케이션 관리가 가능한 기술에 대해 연구하고 이를 구현해 볼 예정이다.

ACKNOWLEDGMENTS

This research is financially supported by Changwon National University in 2015~2016.

REFERENCES

[1] Doug Olenlck. (2015, May). Apple iOS And Google Android Smartphone Market Share Flattening: IDC [Internet]. Available: http://www.forbes.com/sites/dougolenick/2015/05/27/apple-ios-and-google-android-smartphone-market-share-flattening-idc/.

[2] Intel Security. (2015, May). McAfee Labs Threat Report [Internet]. Available: http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2015.pdf.

[3] Taizo Sueyasu, "Application permission model and its management in a new version of Android M," *Nikkei Communications*, pp. 46-47, Jul. 2015.

[4] Dongmin Kim, Heeyoul Kim, "A Novel Android Permission Model Based on User's Policies," *The Journal of Korean Institute of Information Technology*, vol. 12, no. 5, pp. 101-106, May. 2014.

[5] Youngdong Kim, Ikhwan Kim, Taehyoun Kim, "Analysis of Usage Patterns and Security Vulnerabilities in Android Permissions and Broadcast Intent Mechanism," *Korea Institute of Information Security and Cryptology*, vol. 22, no. 5, pp. 1145-1157, Oct. 2012.

[6] Jongmun Jeong, Hoon Lee, Mintae Hwang, "A Study on Vulnerability of Information Security for Android-based Mobile System," Proc. of electronics and communications symposium, vol.4, no.1, pp.99-102, June 2015.

[7] Daeil Yang, *Information Security Introduction*. Hanbit Academy Inc., ch. 7, pp. 323-342, 2013.

[8] Univercity of Seoul Industry Cooperation Foundation, *Analysis of Android Mobile Platform Security Model*, Korea Internet & Security Agency, Ch. 4, pp. 74-126, Aug. 2010.

[9] Android Developers. <manifest> [Internet]. Available: http://developer.android.com/guide/topics/manifest/manifest-element.html.

[10] Min Jae Jo, Ji Sun Shin, "Study on Security Vulnerabilities of Implicit Intents in Android," *Korea Institute of Information Security and Cryptology*, vol. 24, no. 6, pp. 1175-1184, Dec. 2014.

[11] Android Developers. Verifying App Behavior on the Android Runtime (ART) [Internet]. Available: http://developer.android.com/guide/practices/verifying-apps-art.html.

[12] Android Open Source Project. ART and Dalvik [Internet]. Available: https://source.android.com/devices/tech/dalvik/.

[13] Android Developers. Android 5.0 Behavior Changes [Internet]. Available: https://developer.android.com/about/versions/android-5.0-changes.html.

[14] Android Developers. Signing Your Applications [Internet]. Available: https://developer.android.com/tools/publishing/app-signing.html.

[15] Android Developers. System Permissions [Internet]. Available:http://developer.android.com/guide/topics/security/permissions.html.

[16] Android Developers. (2015, September) Dashboards [Internet]. Available: https://developer.android.com/about/dashboards/index.html.

정종문(Jongmun Jeong)

2014 창원대학교 정보통신공학과 공학사
현재 창원대학교 친환경해양플랜트FEED공학과 대학원 석사과정
※관심분야 : 네트워크 보안

이훈(Hoon Lee)

1984 경북대학교 전자공학과 공학사
1986 경북대학교 대학원 전자공학과 공학석사
1996 일본 도호쿠대학 공학연구과 공학박사
현재 창원대학교 정보통신공학과 교수
※관심분야 : 인터넷 설계, QoS 및 네트워크 보안

황민태(Mintae Hwang)

1990 부산대학교 컴퓨터공학과 공학사
1992 부산대학교 컴퓨터공학과 공학석사
1996 부산대학교 컴퓨터공학과 공학박사
1996 한국전자통신연구원 표준연구센터 선임연구원
1999 인제대학교 정보컴퓨터공학부 전임강사
현재 창원대학교 정보통신공학과 교수
※관심분야 : 안드로이드 응용, 매체접속제어 프로토콜, 네트워크 보안