Journal of the Korea Academia-Industrial cooperation Society
Vol. 15, No. 11 pp. 6849-6855, 2014

http://dx.doi.org/10.5762/KAIS.2014.15.11.6849
ISSN 1975-4701 / eISSN 2288-4688

안드로이드 어플리케이션 위변조 방지를 위한 방안 연구

이광형[1*], 김재용[2]
[1]서일대학교 인터넷정보과, [2]숭실대학교 컴퓨터학과

Study on Mechanism of Preventing Application Piracy on the Android Platform.

Kwang-Hyoung Lee[1*], Jae-Yong Kim[2]
[1]**Dept. of Internet Information, Seoil University**
[2]**Dept. of Computer Science, Soongsil University**

요 약 최근 다양한 형태의 서비스를 제공하기 위해 스마트폰의 활용도가 증가함에 따라 안드로이드 앱의 활용에 대한 안전성과 신뢰성 등 보안 문제가 이슈화되고 있다. 안드로이드 앱은 apk 파일 형태로 활용되며, 몇몇의 중요 파일에 의해 실행이 된다. 하지만 이러한 apk파일에 악의적인 소스코드가 삽입되어 통제권 상실이나 권한탈취 등 부정사용에 대한 대상이 될 수 있다. 본 논문은 안드로이드 환경에서 앱의 소스코드 부정사용에 관한 위협을 정의하고, 분석 결과를 기반으로 안드로이드 앱 소스코드 부정사용을 방지하기 위한 방안을 제안한다. 본 논문에서는 불법으로 위변조된 안드로이드 앱을 탐지하고 일반 사용자의 안드로이드 디바이스에 설치되는 것을 방지하기 위한 제 3기관을 이용하여 안드로이드 앱의 무결성을 제공하는 시스템을 제안한다. 제안하는 기법은 일반 사용자와 안드로이드 앱을 제공하는 서비스 서버로만 구성되어 있는 기존의 안드로이드 앱 제공 서비스 시스템과 다르게 안드로이드 앱의 무결성 검증과 사용자 등록을 위한 신뢰할 수 있는 제 3기관을 추가하여 안전한 안드로이드 앱을 제공한다.

Abstract Recently, with the increasing use of smart phones, security issues, such as safety and reliability of the use of the Android application has become a topic to provide services in various forms. An Android application is performed using several important files in the form of an apk file. On the other hand, they may be subject to unauthorized use, such as the loss of rights and privileges due to the insertion of malicious source code of these apk files. This paper examines the Android environment to study ways to define the threats related to the unauthorized use of the application source code, and based on the results of the analysis, to prevent unauthorized use of the application source code. In this paper, a system is provided using a third body to prevent and detect applications that have been counterfeited or forged illegally and installed on Android devices. The application provides services to existing systems that are configured with only the service server that provides users and applications general, This paper proposes the use of a trusted third party for user registration and to verify the integrity of the application, add an institution, and provide a safe application.

Key Words : Android, App Encryption, Code Fraud Prevention, Device Authentication

1. 서론

제한된 기능을 제공하던 기존 스마트폰과 달리 일반 데스크톱과 같이 멀티미디어, 인터넷, 게임 등 다양한 기능을 제공하고 있는 최근 스마트폰은 나날이 사용자의 수가 늘어감에 따라 급속한 시장 성장을 하고 있다. 스마트폰 안에는 많은 개인 정보들이 포함되어 있어 비정상적인 접근을 통해 개인정보 유출과 금전적 피해에 대한 경각심이 높아지고 있다. 일례로 아이폰, 안드로이드 폰의 위치정보가 무단으로 수집되어 광고 등에 이용되었고,

본 논문은 2013년도 서일대학교 학술연구비에 의해 연구되었음.
*Corresponding Author : Kwang-Hyoung Lee(Seoil Univ.)
Tel: +82-2-490-7226 email: dreamace@seoil.ac.kr
Received October 7, 2014 Revised November 5, 2014 Accepted November 6, 2014

80만 명에 달하는 피해자가 발생한 바 있다.

스마트폰은 사용자로 하여금 앱을 통해 다양한 서비스를 제공해준다. 또한 안드로이드 플랫폼의 스마트폰을 위한 구글 안드로이드 마켓은 꾸준히 발전하여 40만개 이상의 앱이 등록되어 있고, 다운로드 회수도 200억 회를 넘고 있다 [1,2]. 하지만 이에 따라 관련 보안사고 역시 큰 폭으로 증가 하고 있으며, 피해 또한 커져 가고 있다[3].

스마트폰은 데스크톱에 적용되던 다양한 악성 코드의 목표가 되고 있지만, 데스크톱에 사용되는 보안 기술은 제한된 처리능력과, 저전력, 부족한 메모리 공간 등과 같은 모바일 환경의 특성을 고려하지 않아 바로 적용하는 것이 쉽지 않다. 또한 많은 모바일 단말기에 사용되고 있는 개방형 플랫폼 안드로이드는 플랫폼 소스가 공개되어 있어 보안 취약점이 노출될 가능성이 크며 그에 따른 2차 피해가 발생할 확률이 굉장히 크다[4].

2011년 이후에는 DroidDream, DroidKungFu, Ginermaster 등의 악성코드 들이 정상 안드로이드 앱을 역공학 후에 악성코드가 포함된 형태로 재패키지 되어 유포 되었다[5-9].

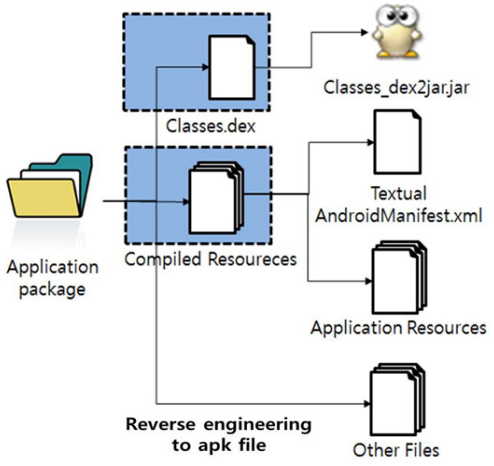
본 논문은 안드로이드 환경에서 앱의 소스코드 부정사용에 대한 기술을 정의하고, 그 특징에 맞춰 해결방안을 제시하여 안드로이드 앱 소스코드 부정사용 방시 시스템을 설계한다. 또한 기존의 연구 결과를 바탕으로 구현결과 및 향후 연구 방향을 제시한다. 일반 사용자와 안드로이드 앱을 제공하는 서비스 서버로만 구성되어 있는 기존의 안드로이드 앱 제공 서비스 시스템과 다르게 안드로이드 앱의 무결성 검증과 사용자 등록을 위한 신뢰할 수 있는 제 3기관을 추가하여 안전한 안드로이드 앱을 제공하는 시스템을 제안한다. 5장 결론에서는 본 논문의 안드로이드 앱 위변조 점검에 대한 취약성 분석을 최종 정리하고, 향후 연구방향으로 안드로이드 앱 전체 항목의 취약성 검증프로세스의 구현을 제시하고자 한다.

2. 관련연구

2.1 정적 역공학

안드로이드 앱에 대한 정적 역공학은 프로그램의 실행 없이 앱의 소스 코드를 분석하는 기법으로 역어셈블(disassemble), 역컴파일(de-compile) 등이 있다[6-9].

다음 Fig. 1은 안드로이드 앱에 대한 정적 역공학의 일반적인 과정을 나타낸다. 먼저 일반 사용자들이 설치, 실행하기 위해 다운 받는 안드로이드 앱의 apk파일에서 바이트코드에 해당하는 Classes, dex 파일을 역어셈블러를 사용하여 추출한다. 다시 바이트코드를 smail 형태의 어셈블리 파일로 변환하고, 어셈블리 코드 형태의 dex 파일을 직접분석과 수정을 가능하게 하기 위한 자바 소스로 복원하는 과정을 거친다.



[Fig. 1] Process of Reverse engineering to apk file

jar 파일로 압축되어 있는 자바 클래스 파일들은 특정 툴을 사용하여 손쉽게 수정이 가능하다. 그렇기 때문에 악의적 공격자들이 다양한 형태의 프로그램 조작과, 악성코드의 삽입이 가능하게 된다. 이후, 변조된 애플리케이션을 대상으로 새로 서명하고 apk 파일로 재패키징 하여 불법 애플리케이션을 배포한다.

2.2 동적 역공학

동적 역공학은 안드로이드 앱을 수행시키면서 분석하는 방법으로 정적역공학에 비해, 실제 안드로이드 앱의 실행 상의 특징을 반영할 수 있다. 디버깅과 동적 인스트루먼테이션(dynamic instrumentation)을 이용하여 동적 역공학을 수행할 수 있다. 구글에서는 ADB, CDMS(Calvik Debug Monitor Service) Device, Android Virtual Device, JDWP Debugger 등의 동적 디버깅 방법을 제공한다. 각 동적역공학에 사용되는 디버깅 방식에 따라 차이는 존재하지만, 동적 역공학을 통하여 내부 변수값 확인, 실행 코드 확인 등이 가능하고, 안드로이드 앱의 핵심 모듈이 노출 될 수 있다[6]. JDWP를 사용하는 JDB는 스마트폰 디바이스와 PC 사이에 세션을 생성한 후, 디버깅

을 수행한다. 구글의 ADB(Android Debug Bridge)에서 JDWP를 지원하기 때문에 마찬가지로 세션을 생성하고 동적 디버깅을 수행한다.

2.3 예방 난독화 기술

난독화는 프로그램 바이너리나 소스코드가 역공학에 의한 분석을 어렵게 하여 방지하는 기술로서, 변환 프로그램의 일종이다[10]. 예방 난독화는 이미 알려진 역난독화 방법을 알고 그 방법을 봉쇄하는 것을 말한다. 기존 프로그램 변환보다는 추가적인 기능을 삽입하는 것에 초점을 두고 있다[11-16]. 대상 배제(Targeted)는 특정 역컴파일러나 디버거 등 역공학 도구를 대상으로 해당 도구가 가진 버그나 취약점을 이용해 정상 동작하지 않도록 하는 기능을 추가하는 것이다. 예를 들어, 어떤 역컴파일러가 메서드의 리턴 뒤에 명령어가 있을 경우 에러가 발생한다고 가정할 때, 난독화 도구는 이와 같은 명령어를 인위적으로 삽입하여 대상컴파일러 사용 시 오류가 발생하도록 유도한다. 무결성 검증은 난독화 대상 프로그램의 위·변조 여부를 검사하는 루틴을 삽입하는 기법이다. 공격자가 역컴파일후 위·변조한 프로그램을 사용 시 해당 루틴이 실행되어 위·변조를 탐지한다[10].

2.4 사용자 식별 및 디바이스 인증

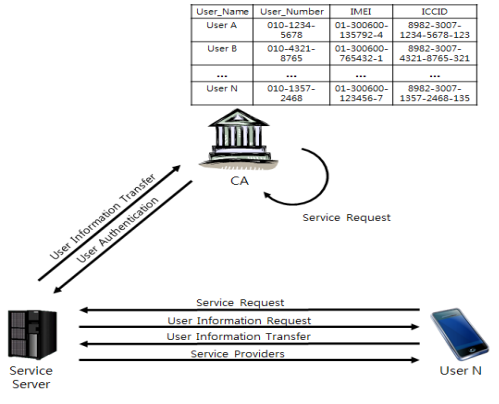
루팅 안드로이드 스마트폰에서 앱이 동작 할 경우 GDB(arm-eabi-gdb), Tcpdump-arm 등이 Root권한으로 분석프로그램을 실행시켜 앱의 동작 상태와 메모리정보 등을 분석하여 악성 앱을 작성 하거나 수정하여 사용할 수 있다. 따라서 안드로이드 환경의 루팅 스마트폰에서 실행되는 금융관련 안드로이드 앱들은 여러 가지 보안 문제를 미연에 방지하고자 그 실행이 차단된다. 스마트 폰 백신에서나 모바일 뱅킹 앱에서 Rooting 탐지 모듈을 적용하여 루팅 된 스마트 폰을 식별하고 프로그램을 종료하는 루틴 적용을 권고한다. 다음은 루팅 탐지하는 모듈의 소스코드 일부이다.

금융 업무를 위한 안드로이드 앱을 사용하는 기관에서는 루팅 탐지를 위한 모듈을 사용하여 루팅 되지 않은 스마트폰에서의 뱅킹 안드로이드 앱의 로그인만 허용하도록 설계 되어 있으나, 모바일 뱅킹 위·변조 안드로이드 앱을 루팅 된 스마트폰에서 실행하면 시스템이 루팅 된 스마트폰을 정상으로 인식하여 로그인이 가능하다. 모바일 뱅킹 위·변조 안드로이드 앱은 스마트폰 용 모바일뱅킹 안드로이드 앱이 나온 수년 전부터 등장해 확산되고 있음에도 아직 뚜렷한 대안이 없는 상태이다. 루팅 스마트폰을 사용한 뱅킹 시스템의 접속은 원천적으로 불가능한 이유로, 루팅 스마트폰 사용자들은 위·변조 안드로이드 앱을 스스로의 필요에 의해 사용기도 한다. 이때 공격자가 위·변조 안드로이드 앱에 다른 의도의 명령어를 심어놓았다면 사용자의 개인정보나 금융정보가 유출되어 대형 금융사고가 발생 할 수 있다. 따라서 금융권 안드로이드 앱에 한해 프로그램 소스를 쉽게 위·변조할 수 없도록 별도의 인증절차를 마련하거나 루팅 스마트폰 사용자들의 정식 안드로이드 앱 사용을 허가 하는 방안에 대한 논의도 이루어지고 있으며, 사용자의 별도 인증과정이 필요한 실정이다.

3. 제안기법

3.1 제안 시스템 구성도

본 논문에서 제안하는 시스템은 안드로이드 환경에서 안드로이드 앱 소스코드 부정사용을 방지하기 위한 신뢰할 수 있는 제 3기관을 이용한 시스템이다. 일반 사용자와 안드로이드 앱을 제공하는 서비스 서버로만 구성되어 있는 기존의 안드로이드 앱 제공 서비스 시스템과 다르게 안드로이드 앱의 무결성 검증과 사용자 등록을 위한 신뢰할 수 있는 제 3기관을 추가하여 안전한 안드로이드 앱을 제공한다. 다음의 Fig. 2는 제안하는 시스템의 전체 구성도이다.



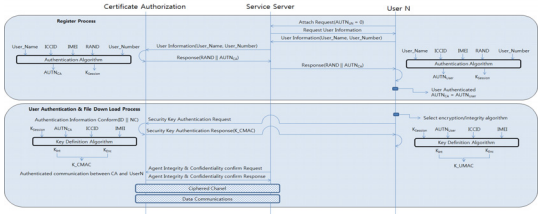
[Fig. 2] Diagram of the proposed system

CA : 사용자의 정보를 저장하고 SS로부터 인증 요청을 처리
SS : 실질적인 서비스를 제공하며, 사용자와 직접적인 통신을 수행
User : 사용자 또는 모바일 단말기

제안 시스템에 필요한 구성 요소는 CA(Certificate Authorization), SS(Service Server), User와 같으며, 각 콘텐츠의 역할을 다음과 같다.

제안하는 시스템의 구조는 사용자의 요청으로 사용자 정보를 저장하고 해당 정보를 CA에 인증 받음으로서, 올바른 사용자를 인증하고 안드로이드 앱에 대한 인가된 서비스를 제공해 준다. 이때 인증 받은 사용자의 디바이스가 아니면 안드로이드 앱이 올바로 실행이 되지 않으며, 인가받은 안전한 안드로이드 앱을 사용하게 함으로써, 안드로이드 앱의 소스가 부정 사용되는 사례를 미연에 방지할 수 있다.

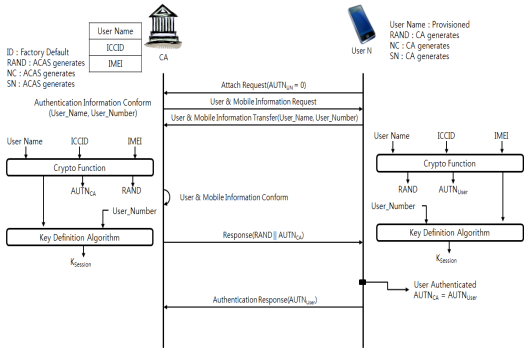
다음 Fig. 3은 제안하는 시스템 구성도의 상세 프로토콜이다. 해당 프로토콜은 등록 과정과 사용자 인증 및 파일 다운로드 과정으로 나뉜다. 각 인증과정마다 사용되는 인증 방식은 다르며, CA는 사용자 단말의 IMEI값과 ICCID값을 통신사로부터 제공 받았다고 가정한다.



[Fig. 3] Details of the proposed system protocol

3.2 사용자 등록

제안하는 시스템의 사용자 등록 과정은 사용자의 단말정보를 활용하여 생성하는 인증 값과 CA로부터 수신받은 랜덤 값을 조합하여 세션키 값을 생성하게 된다. 생성된 세션키 값은 사용자 인증 과정 및 파일 다운로드 과정에서 사용된다.



[Fig. 4] User Registration Process

사용자는 SS에 서비스를 요청하기 위해 사용자가 보유하고 있는 값을 활용하여 접근 요청을 시도하며, SS는 사용자의 정보 인증과 사용자 정보 업데이트를 위해 CA에 데이터를 전송한다. 사용자는 SS에 접근 요청을 초기에 시도할 때 다음 (1)과 같은 메시지를 전송한다.

$$NWID, NWP, AUTN_{UN} = 0 \tag{1}$$

CA는 NWID값과 NWP값이 사용자의 초기 접근이 올바른지를 확인하기 위한 정보이다. NWID값과 NWP값이 올바른 값인지 확인되면, CA는 사용자의 올바른 인증을 하기 위해 사용자 정보를 요청한다. AUTNUN=0 : 사용자에 대한 인증 시도가 이루어지지 않은 초기 접근을 나타낸다.

초기 접근요청 메시지를 전송한 후 CA는 사용자의 정보를 확인하기 위한 인증정보를 요청한다. 사용자는 인증정보를 CA에게 전송하며 전송 요청 메시지의 정보는 다음 (2)와 같다.

$$User\_Name, User\_Number \tag{2}$$

사용자_Name은 사용자가 지정한 장비의 이름이며, 사용자_Number은 사용자 디바이스의 전화번호이다. 해당 값을 수신한 CA는 휴대폰 개통 시에 등록된 사용자의 전화번호와 이름에 맞는 ICCID 값과 IMEI값을 추출하여 User_Name값과 조합한 값과 랜덤 값을 생성 한다. 해당 값을 생성하는 과정은 수식 (3)과 같다.

$$Crypto\_Function(User\_Name \| ICCID \| IMEI) = RAND \| AUTN_{CA} \tag{3}$$

CA는 랜덤값 생성 후 패딩 값과 같이 사용자의 전화번호를 조합하여 알고리즘 연산 후 사용자 인증 및 파일 다운로드를 할 때 사용될 세션키를 생성하게 되며 해당 연산 과정은 수식 (4)와 같다.

$$Key\_Definition\_Alogorithm(User\_Nmber \| Padding) = K_{Session} \tag{4}$$

CA는 사용자가 세션키를 생성하기 위한 입력값 중 랜덤값과 CA의 인증 토큰 값을 전송해준다. 전송하는 데이터 값은 (5)와 같다.

$$RAND \| AUTN_{CA} \tag{5}$$

사용자 단말기가 보유하고 있는 정보와 CA로부터 수신한 정보를 조합하여 사용자의 인증 토큰 값과 세션키를 생성하기 위한 랜덤값을 생성해 낸다. 해당 값을 생성하는 과정은 수식 (6)과 같다.

$$Crypto\_Function(User\_Name \| ICCID \| IMEI) = RAND \| AUTN_{User} \tag{6}$$

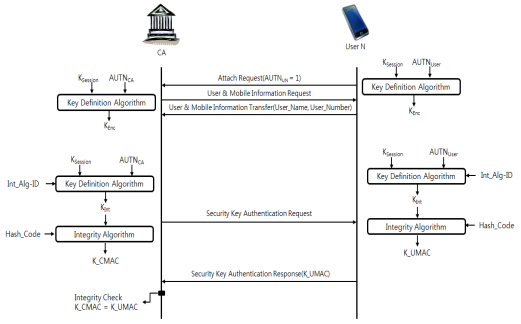
사용자 단말기는 랜덤값 생성 후 패딩 값과 같이 사용자의 전화번호를 조합하여 알고리즘 연산 후 사용자 인증 및 파일 다운로드 할 때 사용될 세션키를 생성하게 되며 해당 연산 과정은 수식(7)과 같다.

$$Key\_Definition\_Alogorithm(User\_Nmber \| Padding) = K_{Session} \tag{7}$$

사용자는 사용자 단말이 생성한 인증 토큰 값과 CA로부터 전송받은 인증 토큰 값을 비교하여 CA의 올바른 인증을 수행한다. CA가 올바로 인증되었으면 사용자가 생성한 인증 토큰을 전송하여 사용자 등록과정을 종료한다.

3.2 사용자 인증 및 파일 다운로드

사용자 단말기와 CA간 인증이 완료된 후 파일 다운로드가 가능하며, 사용자 단말기와 CA간 올바른 인증 및 파일 다운로드 과정은 다음 그림과 같다.



[Fig. 5] User Registration Process

CA에 인증이 완료된 사용자 단말기는 각각에게 주어진 정보와 사용자 등록 과정에서 생성한 세션키를 활용하여 사용자를 안전하게 인증하는 과정을 거친다. CA는 기밀성과 무결성을 보장하기 위해 사용될 알고리즘을 정의하여 기밀성 키와 무결성 키를 생성하게 되며, 생성과정은 수식 (8), (9)와 같다.

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{CA}) = K_{Enc} \tag{8}$$

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{CA} \| Int\ \_Alg-ID) = K_{Int} \tag{9}$$

CA는 수식 (9)의 연산을 통해 산출된 값과 해시코드를 조합하여 무결성 알고리즘에 삽입하여 해쉬값을 추출해 내며, 과정은 수식 (10)과 같다.

$$Integrity\_Algorithm(Hash\_Code \| K_{Int}) = K\_CMAC \tag{10}$$

CA는 알고리즘 ID값을 기밀성 키로 암호화하고 해쉬코드 값을 알고리즘 연산 후 사용자 단말기에 전송하며, 연산 및 전송하는 데이터는 (11)과 같다.

$$E_{K_{Enc}}(Int\ \_Alg-ID \| Hash\_Code) \tag{11}$$

CA로부터 데이터를 수신한 사용자 단말기는 기밀성 키를 구하기 위한 연산을 시작하며, 연산과정은 수식 (12)와 같다.

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{User}) = K_{Enc} \tag{12}$$

수식 (12) 과정에서 얻은 기밀성 키값을 사용하여 CA로부터 수신한 데이터를 복호화 하여 무결성 키와 무결성에 사용될 해시코드를 구한다. 연산 과정은 수식 (13)과 같다.

$$D_{K_{Enc}}(E_{K_{Enc}}(Int\ \_Alg-ID \| Hash\_Code)) \tag{13}$$

사용자 단말은 수식 (13)의 연산과정을 통해 얻은 해쉬코드와 무결성 알고리즘 ID를 통해 무결성 키와 해쉬값을 추출해내는 알고리즘 과정을 거친다. 해당 연산 과정은 수식 (14), (15)와 같다.

$$Key\_Definition\_Algorithm(K_{Session} \| AUTN_{User} \| Int\_Alg-ID) = K_{Int} \quad (14)$$

$$Integrity\_Algorithm(Hash\_Code \| K_{Int}) = K\_UMAC \quad (15)$$

연산이 완료 후 사용자 단말기는 최종결과로 산출된 해쉬값을 CA로 전송한다. CA는 사용자 단말기로부터 수신한 해쉬값이 CA가 생성한 CA값과 동일하면 무결성 체크가 완료된다.

4. 분석 및 성능평가

본 논문은 안드로이드 환경에서 안드로이드 앱 소스코드 부정사용을 방지하기 위한 시스템을 제안 및 설계하였다. 안드로이드 앱 소스코드 부정사용에 대한 여부와 탐지 율은 탐지차단의 양에 따른 정확도에 따라 평가되기 때문에 본 논문에 대한 평가는 안드로이드를 구현하여 임의의 파일 다운로드에 따른 다른 기기 공유 여부와 소스코드 부정사용을 위한 파일 역공학 여부에 대한 기준으로 성능 분석을 진행하였다. 본 논문에서 제안하는 소스코드 부정사용 방지 시스템을 구현한 환경은 다음의 Table 1과 같다.

[Table 1] Implementation

No	Parameter	Description
1	Hardware : CPU : PentiumIV Inter(R) Core(TM) Quad 2.66 GHz	Server
2	Software : Microsoft Visual Studio C# 2010	CA & SS
3	OS : Windows 7 Enterprise K 64bit	Server OS
4	Galaxy 3 ; Version 4.1.2	User Device
5	Evaluation List	Block unauthorized access frequency

다음 Table 2는 기존 시스템과 제안 시스템과의 성능을 비교하여 나타낸 것이며, 제안 시스템은 기존 시스템에 비해 4가지 항목에 대해 개선된 것을 알 수 있다.

[Table 2] Comparative analysis of the proposed system with existing systems

Parameter	Original	Proposed
Application management	X	O
Secure Communication between Device	X	O
Strength of access control	Low	High
Potential of infringement reverse engineering	High	Low

Strength of access control은 Communication between Device와 같이 단말기에 대한 고유 정보가 없으면 악의적인 방식을 통한 접근을 차단하므로 기존 시스템보다 접근제어에 대한 강도가 강하고, Potential of infringement reverse engineering에 대한 제어가 강하여 침해 가능성이 낮게 측정됨을 알 수 있다. Application management와 Secure Communication between Device에 관련된 항목은 프로토콜 구조상 기존 시스템보다 제안 시스템이 더 강화된 보안적 특성을 제공한다.

5. 결론

안드로이드 환경에서의 악성코드와 취약성을 이용한 보안 위협 사례는 현재까지도 꾸준히 증가하고 있다. 안드로이드 앱의 위·변조는 그 자체로 인한 저작권법 위반 등의 피해를 유발 할뿐만 아니라, 스마트폰에 저장된 개인 사용자 정보의 유출과 오남용 등의 2차 피해를 유발하기 때문에 그 심각성이 더 크다.

본 논문에서는 안드로이드 환경에서 안드로이드 앱의 소스코드 부정사용에 관한 위협을 정의하고, 분석 결과를 기반으로 안드로이드 앱 소스코드 부정사용을 방지하기 위한 시스템을 제안하였다. 제안 하는 시스템은 사용자의 요청을 통하여 사용자의 정보를 저장하고 CA에서 인증과정을 거침으로서, 올바른 사용자와 악의적인 사용자의 소스코드 부정사용 여부를 구분한다. 제안 시스템은 소스코드 부정사용을 위한 역공학의 사용 여부에 대한 기존 탐지 방법에 비해, 접근제어에 대한 강도가 강하며, 침해 가능성을 보다 낮추었다. 안드로이드 앱 위·변조 방지를 해결하며 적용되는 기술의 향상은 안드로이드 및 기타 모바일 환경에서의 보안 기술의 파급효과와 보안 안정성의 증가 부분에서 효과적인 영향을 미칠 것이다.

단, 본 논문이 제안하는 기법의 안정성을 향상시키기 위해서는, 안드로이드 환경에서 안드로이드 앱이 사용되는 유저 디바이스와 별도의 CA, SS등의 부가적인 시스템의 구성과, 사용자 등록 및 인증 절차의 복잡함의 개선에 대한 연구가 필요하다. 향후 보완점이 해결된 향상된 시스템을 통하여 위변조 방지를 통한 S/W 저작권 및 지적 재산권에 대한 보장뿐 아니라, 금융시장을 중심으로 한 다른 산업에도 긍정적인 영향을 미칠 것으로 기대한다. 또한 위변조를 인해 발생되는 2차 피해를 방지하여 안드로이드 분야의 보안사고 및 침해 사례 감소 효과를

기대한다.

References

[1] Loyce Consulting.: 2011 Market Survey of Smart Content. KOCCA Research Report 11-66, Korea Creative Content Agency (2011)

[2] Jegal Byeongjik.: Smartphone Market and Mobile OS Trends. Semiconductor Insight (2011)

[3] Android Under Attack: Malware Levels for Google's OS Rise Threefold in Q2 2012, http://www.kaspersky.com/about/news/press/2012/Android_Under_Attack_Malware_Levels_for_Googles_OS_Rise_Threefold_in_Q2_2012

[4] Chan-Hee Lee, Yeong-Ung Park, Ji-Hyeog Lim, Hong-Geun Kim, Choong-Hyun Lee, Seong-Je Cho and Jaesoo Yang, Jounal of KIISE : Computing Practices and Letters, Vol 18, No.10, pp.692-700, (2012)

[5] T. Bradley, "DroidDream Becomes Android Market Nightmare," PCWorld, Mar.2011, http://www.pcworld.com/article/221247/droiddream_becomes_android_market_nightmare.html

[6] Steve Gold, "Android insecurity," Network Security, vol.2011, Issue.10, pp.5-7, (Oct. 2011).

[7] GingerMaster, http://www.cs.ncsu.edu/faculty/jiang/GingerMaster/

[8] Juniper Networks, "Mobile Signature," http://www.juniper.net/us/en/security/mobile-threat-center/#ANDROID

[9] DroidDream, http://blog.mylookout.com/blog/2011/03/01/security-alert-malware-found-in-official-androidmarket-droiddream/

[10] C. Collberg, C. Thomborson, and D. Low, "A taxonomy of obfuscating transformations," Technical Report No. 148, Univ. Auckland, New Zealand, (1997).

[11] ByeongYong Lee1, YongSoo Choi2, : The Status and Analysis of Obfuscation Techniques and Perspective Development. Journal of Security Engineering, Republic of Korea, vol.5, No. 3, pp.692-700, (2008).

[12] S. Choi, M. Kim, J. Han, B. An, "Android Based Mobile Student Identity Card", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 2, Apr. 2013.

[13] E.-J. Jo, C.-H. Lin, "Smart Emotion Lighting Control System Based on Android Platform", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 14, No. 3, pp.147-153, Jun. 2014.

[14] J.-M. You, I.-K. Park, "Android Storage Access Control for Personal Information Security", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 6, Dec. 2013.

[15] S.-C. Lim, "A Study of Android Launcher based on Application Virtualization", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 2, Apr. 2013.

[16] J.-g. Lim, C.-s. Choi, T.-e. Park, H.-s. Ki, B. An, "Android Based Mobile Combination Login Application", The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 13, No. 3, Jun. 2013.

이 광 형(Kwang-Hyoung Lee) **[종신회원]**

- 1998년 2월 : 광주대학교 컴퓨터공학과 졸업(공학사)
- 2002년 2월 : 숭실대학교 컴퓨터공학과 (공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 서일대학 인터넷정보과 부교수

<관심분야>
멀티미디어 데이터 검색, 영상처리, 멀티미디어 보안, DRM, USN, 학습콘텐츠

김 재 용(Jae-Yong Kim) **[정회원]**

- 2010년 2월 : 숭실대학교 일반대학원 컴퓨터공학(공학석사)
- 2010년 3월 ~ 현재 : 숭실대학교 일반대학원 컴퓨터공학과 (박사수료)

<관심분야>
정보통신, 통신보안, 암호이론, 네크워크보안