안드로이드 환경에서 권한 요구를 통해 발생 가능한 잠재적 보안위협에 관한 연구

박인규*, 최슬기*, 곽진**
*아주대학교 컴퓨터공학과 정보보호응용및보증연구실
**아주대학교 사이버보안학과
e-mail:ikpark.isaa@gmail.com*, skchoi.isaa@gmail.com*, security@ajou.ac.kr**

A Study on Potential Security Threats that Occurred Possibly by Permission Requests in Android Environment

In-Kyu Park*, Seul-Ki Choi*, Jin Kwak**
*ISAA Lab., Department of Computer Engineering, Ajou University
**Department of Cyber Security, Ajou University

요 약

안드로이드 애플리케이션 개발 시 특정 기능을 수행하기 위해 필수적으로 AndroidManifest.xml 파일에 권한을 등록한다. 등록된 권한은 애플리케이션 설치 시 사용자에게 동의를 요구하며 애플리케이션을 설치하기 위해 해당 애플리케이션이 요구하는 사용 권한 모두를 동의해야 한다. 애플리케이션이 요구하는 권한에 동의하지 않을 경우 애플리케이션 설치가 취소되며, 권한 요구에 대한 동의 없이 애플리케이션을 설치할 수 없다. 이러한 이유로 대부분의 사용자는 애플리케이션이 본래의 기능과 맞지 않는 필요 범위 이상으로 요구하는 권한까지도 상세하게 살펴보지 않고 동의하여 본래의 기능과 관계없는 권한을 통해 개인정보 유출, 비정상적인 과금 유발 등 다양한 보안위협이 발생한다.

본 논문에서는 안드로이드 애플리케이션의 APK 파일을 통해 애플리케이션이 요구하는 권한과 실제 애플리케이션이 기능 제공을 위해 사용하는 API 목록을 확인하여 애플리케이션이 필요 범위 이상으로 요구하는 권한을 분석한다. 또한, 애플리케이션 업데이트 시 API 추가로 인해 발생할 수 있는 보안위협을 분석한다. 이를 통해 필요 범위 이상으로 요구하는 권한으로 발생할 수 있는 보안위협에 대해 사전에 인지하여 이를 방지할 수 있도록 한다.

1. 서론

전 세계 스마트폰 이용자가 점차 증가하면서 2016년 기준 스마트폰 보급률은 약 70%까지 상승하였으며, 이 중 구글의 안드로이드가 모바일 플랫폼 시장의 약 86.2%를 차지하였다. 안드로이드 사용자가 점차 증가함에 따라 안드로이드에 대한 공격 증가로 인해 개인 정보 유출 등 다양한 취약점이 발견되었다[1,2].

안드로이드에 존재하는 다양한 취약점 중 필요 범위 이상의 권한 요구를 통해 발생할 수 있는 취약점이 존재한다. 안드로이드 애플리케이션은 개발 시 특정 기능을 수행하기 위해 필수적으로 AndroidManifest.xml 파일에 권한을 등록한다. 안드로이드 환경에서 애플리케이션을 설치할 경우 개발 시 등록된 애플리케이션에 대한 필수 권한을 요구하게 되며 권한 사용을 거부할 경우 해당 애플리케이션을 설치 할 수 없다. 악성행위를 하는 애플리케이션의 경우 일반적으로 애플리케이션이 동작하기 위해 필요한 권한 외에 추가적으로 필요 범위 이상의 권한을 요구하여 해당 권한을 통해 악성행위를 한다. 예를 들어 단순히 SMS를 전송하는 애플리케이션에 WIFI 정보, 카메라 및 패키지 설치·삭제와 관련된 권한을 추가하여 WIFI, 카메라, 패키지를 조작하는 악성행위를 수행할 수 있다. 따라서, 애플리케이션이 요구하는 권한 중 필요 범위 이상의 권한 요구를 식별하고 필요 범위 이상의 권한 요구로 인해 발생할 수 있는 보안위협을 분류하여 보안위협을 사전에 방지할 필요가 있다.

본 논문에서는 애플리케이션의 APK 파일의 분석 통해 해당 애플리케이션이 요구하는 권한과 실제 애플리케이션이 동작하며 사용하는 API를 확인한다. 확인된 내용을 바탕으로 애플리케이션이 필요 범위 이상으로 요구하는 권한을 분류하고 이를 통해 애플리케이션 업데이트 시 API 추가를 통해 발생할 수 있는 잠재적 보안위협에 대해 분석한다. 분석된 내용을 바탕으로 애플리케이션의 필요 범위 이상의 권한 요구를 통해 발생할 수 있는 잠재적 보안위협을 파악하고 이를 인지하여 사전에 방지 할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문과 관련하여 APK 파일, 안드로이드 권한 및 권한 보호 수준에

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2014R1A2A1A11050818).

대해 살펴보고, 3장에서 안드로이드 애플리케이션의 APK 파일로부터 해당 애플리케이션이 요구한 권한과 실제 애플리케이션이 실행되면서 사용하는 API를 확인하여 필요 범위 이상의 요구된 권한을 분류하고 이를 통해 애플리케이션 업데이트 시 발생할 수 있는 보안위협을 분석한다. 마지막으로 4장에서 결론을 맺는다.

2. 관련연구

2.1 APK 파일

APK 파일은 크게 4가지 구성요소로 이루어지며 DEX 파일, 컴파일 된 리소스, 컴파일 되지 않는 리소스, AndroidManifest.xml 파일로 구성된다. APK 파일은 파일 ZIP 파일 형식의 압축파일로 구성되어 있기 때문에 일반적인 압축해제 방식으로 해제가 가능하지만, 단순 압축해제 방식으로는 확인할 수 없는 파일들이 있어 Apktool과 같은 역공학 도구를 이용한 디컴파일 과정이 필요하다[3,4].

다음 <표 1>은 APK 파일을 구성하는 4가지 요소들과 이에 대한 설명을 나열한 표이다[4].

<표 1> APK 파일 구성요소 정의

구성 요소	설 명
DEX 파일	- 모든 바이트코드를 포함하며 일반적으로 한 개로 구성되지만 필요에 따라 두 개 이상으로도 구성
AndroidManifest.xml	- 권한이나 시작 클래스 등과 같은 정보 포함
컴파일 된 리소스	- 문자열 상수나 윈도우 레이아웃 등으로 구성되며 컴파일 될 수 있는 XML 파일 포함
컴파일 되지 않는 리소스	- 이미지나 네이티브 코드 등과 같은 원본 그대로 보관되는 요소

2.2 안드로이드 권한

안드로이드 애플리케이션은 개발 시 특정 기능을 사용하기 위해 AndroidManifest.xml 파일에 필수적으로 권한을 등록한다. 등록된 권한은 애플리케이션을 설치할 때 사용자에게 보여지며 해당 권한들에 대한 동의를 요구한다. 권한을 동의할 때 애플리케이션이 요구한 모든 권한에 대한 동의가 이루어져야 하며 일부의 권한만 별도로 동의를 할 수가 없다. 애플리케이션이 요구한 사용 권한에 동의할 경우 해당 애플리케이션이 설치되며, 동의하지 않을 경우 애플리케이션의 설치가 취소된다. 이러한 이유로 사용자는 애플리케이션을 설치하기 위해 본래의 기능과 관련 없는 필요 범위 이상의 권한 요구까지 별도의 확인 없이 동의하는 경우가 많다[5].

2.3 안드로이드 권한 보호 수준

안드로이드 애플리케이션 개발 시 사용자 지정 권한을 선언할 경우, 권한에 대한 보호 수준을 지정할 수 있다. 권한 보호 수준은 총 4가지이며 권한 보호 수준은 수준별로 normal, dangerous, signature, signatureOrSystem가 존재한다.

다음 <표 2>는 4가지 권한 보호 수준에 대한 설명을 나열한 표이다[5].

<표 2> 권한 보호 수준 정의

권한 보호 수준	설 명
normal	- 일반적으로 애플리케이션에 대해 발생할 수 있는 보안위협이 적은 권한들에 부여 - 애플리케이션을 설치하는 과정에서 공지되지 않아 사용자의 동의 없이 사용 가능
dangerous	- 사용자의 동의하에 해당 권한 사용이 가능하며 normal에 비해 위험성이 높은 권한에 부여 - 애플리케이션을 설치할 경우 사용자에게 요구 권한을 알리고 해당 권한에 대한 동의 여부 확인
signature	- 개발자에 의해 서로 동일한 인증서로 서명된 경우에 설치 가능 - 사용자에게 특별히 공지되지 않음
signatureOrSystem	- signature와 유사하지만 안드로이드 시스템 이미지의 애플리케이션이나 애플리케이션과 동일한 서명을 갖는 경우에 부여 - 일반적으로 여러 제조사에서 애플리케이션 공동 구축을 위해 특정 기능을 공유할 필요가 있을 경우에 사용 - 사용자에게 특별히 공지되지 않음

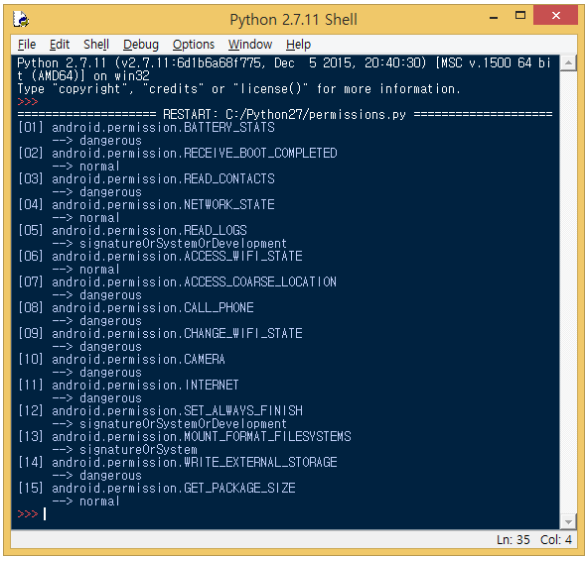
3. 필요 범위 이상의 권한 요구 및 보안위협 분석

애플리케이션에 대한 권한과 사용하는 API 등 본 논문에서 수행되는 모든 분석은 APK 파일을 통해 이루어진다. 애플리케이션의 필요 범위 이상의 권한 요구로 인해 잠재적으로 발생할 수 있는 보안위협에 대한 분석은 권한 분석, 실제 사용된 API 분석, 필요 범위 이상으로 요구된 권한 분석, 필요 범위 이상으로 요구된 권한으로 인해 발생할 수 있는 보안위협 분석 순으로 이루어진다.

3.1 안드로이드 권한 분석

사용자의 스마트폰에 설치된 안드로이드 애플리케이션이 필요 범위 이상으로 권한을 요구하는지 확인하기 위해 해당 애플리케이션의 APK 파일로부터 요구 권한을 분석한다. APK 파일로부터 애플리케이션이 요구하는 권한을 분석하기 위해 안드로이드 역공학 도구인 Androguard 도구를 사용한다. APK 파일의 경우 압축 해제 후 Android Manifest.xml 파일 내에 있는 내용을 통해 애플리케이션이 요구하는 권한을 확인할 수 있지만, 본 연구에서는 빠르고 효율적인 분석을 위해 Androguard 도구를 사용하여 분석을 수행한다[6,7].

다음 (그림 1)은 특정 애플리케이션이 요구하는 권한을 분석하기 위해 임의적으로 만든 애플리케이션의 APK 파일을 통해 Androguard 도구로부터 추출한 애플리케이션에 존재하는 권한 목록을 보여준다.



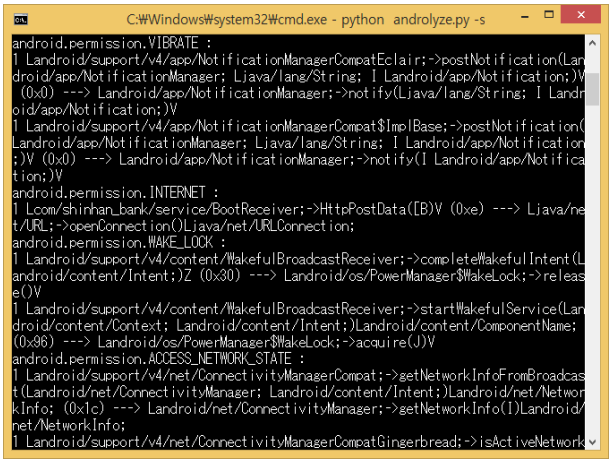
(그림 1) APK 파일로부터 추출한 요구 권한 목록

3.2 안드로이드 API 분석

본 절에서는 안드로이드 애플리케이션이 3.1에서 분석한 권한을 실제로 사용하는지 확인하기 위해 APK 파일로부터 애플리케이션이 기능 제공을 위해 사용하는 모든 API를 분석한다. 사용된 API에 대한 분석은 애플리케이션 설치 시 요구된 권한과 관련된 API가 어느 위치에서 사용이 되었는지 확인하여 요구된 권한에 대한 실제 사용 여부를 판별한다. 애플리케이션의 API에 대한 실제 사용 여부 분석은 Androguard에서 제공하는 Androlyze.py를 통해 이루어진다[6,7].

다음 (그림 2)는 안드로이드 애플리케이션의 APK 파일을 분석하여 애플리케이션에서 기능 제공을 위해 실제 사용한 API를 분석한 결과를 보여준다. 해당 결과를 통해 애플리케이션에서 요구한 권한이 어느 위치에서 사용되었는지 파악할 수 있다.

애플리케이션이 기능 제공을 위해 사용한 API를 식별할 수 있게 될 경우 API를 통해 해당 API와 관련된 권한과 매핑시켜 애플리케이션이 요구한 권한이 모두 사용되는지 확인할 수 있다.



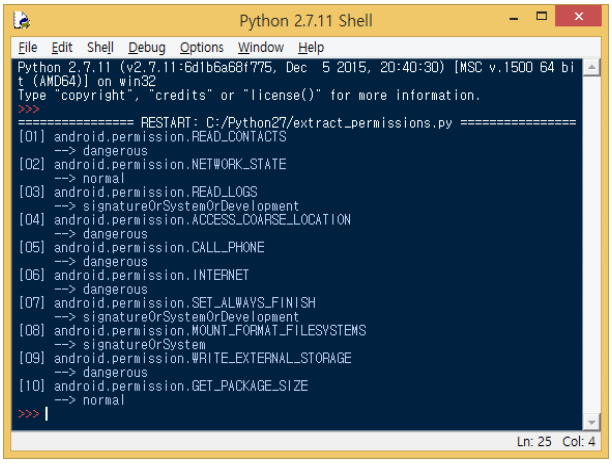
(그림 2) APK 파일로부터 확인한 사용된 API 목록

Androlyze.py의 분석 과정에서 권한별 사용되는 API에 대한 정의는 androguard-2.0/androguard-2.0/androguard/core/bytecodes/api_permissions.py 파일에 되어 있으며, 본 연구에서 권한별 사용되는 API는 api_permission.py에 정의된 내용을 따른다.

3.3 필요 범위 이상의 요구 권한 분석

3.1과 3.2를 통해 분석된 내용을 바탕으로 애플리케이션이 필요 범위 이상으로 요구하고 있는 권한을 분석한다. 3.1에서 분석한 애플리케이션에서 요구하고 있는 권한과 3.2에서 분석한 애플리케이션에서 기능 제공을 위해 실제 사용하고 있는 API 목록을 바탕으로 권한 요구만 되어있고 실제 사용되지 않는 권한을 분류한다.

다음 (그림 3)은 테스트용으로 제작한 애플리케이션에서 3.1과 3.2의 분석 내용을 바탕으로 사용되지 않는 권한 목록을 추출한 결과의 예시이다.



(그림 3) 필요 범위 이상으로 요구된 권한 목록

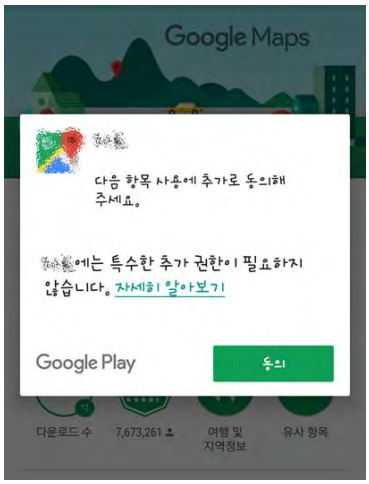
3.4 보안위협 분석

애플리케이션 개발 시 개발자는 AndroidManifest.xml 파일에 의도적이거나 실수로 필요 이상의 권한을 명시하여 사용자로부터 필요 이상의 권한에 대한 동의를 요구하는 경우가 발생한다. 3.1에서 분석한 것처럼 애플리케이션 설치 시 요구한 권한과 다르게 일부 사용되지 않는 권한만으로는 악성행위를 한다고 판단하기 어렵다. 그러나 현재 사용되지 않는 권한들로 인해 발생 가능한 보안위협이 식별 되지 않는다고 하더라도 애플리케이션 업데이트를 통해 악성행위를 수행할 수 있다. 업데이트 시 발생할 수 있는 악성행위 절차는 다음과 같다.

Step 1. 사용자로부터 필요 이상의 권한을 요구하여 필요 이상의 권한을 갖는 애플리케이션을 설치한다.

Step 2. 설치된 애플리케이션에서 업데이트 시 기존에 요구한 권한을 통해 API를 추가한다. 이때, 기존에 요구된 권한을 통해 API를 추가하는 것이기 때문에 (그림 4)와 같이 사용자에게 API 추가에 대해 알리지 않는다.

Step 3. 업데이트 시 추가된 API를 통해 데이터 유출, 비정상적 과금 유발 등의 악성행위를 수행한다.



(그림 4) 업데이트 시 요구하는 추가 권한

업데이트 시 애플리케이션에 악성 행위를 목적으로 권한을 추가할 경우 안드로이드에서 추가된 권한에 대해 명시하여 권한에 대한 동의를 추가적으로 요구한다. 하지만 이미 요구된 권한에 대해서는 동의를 요구하지 않으며, API가 추가되더라도 이미 동의가 완료된 권한 또한 동의를 요구하지 않는다.

따라서, 기존에 애플리케이션 설치 시 권한 요구에 대한 동의가 이루어졌다 하더라도 필요 이상으로 요구된 권한을 식별해야 하며, 필요 이상으로 요구된 권한으로부터 추후 업데이트 시 추가된 API를 통해 발생할 수 있는 보안위협을 사전에 인지할 필요가 있다. 또한, 인지한 정보를 통해 보안위협을 사전 방지할 필요가 있다.

4. 결론

본 논문에서는 애플리케이션 설치 시 요구한 권한이 필요 범위 이상으로 요구된 것인지에 대한 여부와 필요 범위 이상의 권한 요구로 인해 발생할 수 있는 보안위협을 분석하기 위해 애플리케이션이 요구한 권한과 실제 애플리케이션이 실행되면서 사용하는 API를 분석하였다. 분석한 결과 애플리케이션이 요구한 권한 목록과 애플리케이션이 실제 사용하는 API 목록을 확인할 수 있었으며 이로 인해 애플리케이션이 불필요하게 많은 권한을 요구하는지 확인할 수 있었다. 또한, 애플리케이션 업데이트 시 악의적인 목적으로 추가된 API로 인해 발생할 수 있는 보안위협을 분석하여 사용자로부터 사전에 보안위협을 인지하고 이를 방지할 수 있도록 하였다.

향후 본 논문에서 분석한 잠재적 발생위협을 수치화하여 해당 애플리케이션이 업데이트 시 발생시킬 수 있는 보안위협을 식별할 뿐만 아니라 수치화된 결과를 바탕으로 분석된 애플리케이션의 악성여부를 판별할 수 있는 연구를 진행하려한다. 또한, 악성여부를 판별할 수 있는 연구가 완료되면 애플리케이션에 대한 분석 서비스를 제공하는 웹 서버를 구축하려한다.

참고문헌

[1] “2016 모바일 트렌드 전망”, KT 경제경영연구소, 2016.01.
[2] 정호윤, "인터넷 | 계속 확장되는 구글의 영토", 유진투자증권, 2016.10.
[3] HNS(HackNSecurity), “안드로이드 악성코드 분석", http://www.hacknsecurity.com, 2014.01.
[4] 하동수, 이강효, 오희국, "안드로이드 애플리케이션 역공학 보호기법", 제25권 제2호. 정보보호학회지, 2015.06.
[5] 김영동, 김익환, 김태현, "안드로이드 권한과 브로드캐스트 인텐트 메커니즘의 사용 현황 및 보안 취약성 분석", 제22권 제5호. 정보보호학회논문지, 2012.10.
[6] 조병철, 최진영, "모바일 앱 최소권한 사전검증에 관한 연구 - 금융, 안드로이드 운영체제 중심으로 -", 제17권 제1호. 인터넷정보학회논문지, 2016.02.
[7] Daiki Chiba, Shigeki Goto, "Detecting Android Malware by Analyzing Manifest Files", Asia-Pacific Advanced Network, 2013.
[8] 유홍렬, 권태경, "안드로이드 권한의 위험을 판단하기 위한 평가척도에 관한 연구", 1009-1011. 한국컴퓨터종합학술대회 논문집, 2014.06.
[9] Zami Aung, Win Zaw, "Permission-Based Android Malware Detection", VOLUME 2 ISSUE3. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, 2013.03.