권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션에 관한 연구

김준섭*, 곽진**
*순천향대학교 정보보호학과 정보보호응용및보증연구실
**순천향대학교 정보보호학과
e-mail:jskim0911@sch.ac.kr, jkwak@sch.ac.kr

A Study on Android Antivirus Application through Permission Management

Jun-Sub Kim*, Jin Kwak**
*ISAA Lab, Dept of Information Security Engineering, Soonchunhyang University
**Dept of Information Security Engineering, Soonchunhyang University

요 약

안드로이드는 스마트폰에서 프로그램을 실행하도록 하는 구글에서 개발한 모바일 전용 운영체제로써 현재 수백만 대의 스마트폰, 태블릿PC에 탑재되어 있다. 안드로이드는 빠른 속도의 웹브라우저, 멀티태스킹, 클라우드 컴퓨팅 기능, 다른 장치와 쉽게 연결하여서 공유하는 기능 등을 제공하고 있다. 이에 따라 많은 스마트폰 제품들이 안드로이드 운영체제를 탑재하여 출시하고 있으며, 안드로이드는 전 세계 스마트폰 운영체제 점유율의 절반가량을 차지하고 있다. 하지만 안드로이드 운영체제가 많이 사용됨에 따라 그에 따른 안드로이드 악성코드 또한 급격하게 증가하고 있다. 따라서 본 논문에서는 안드로이드 악성코드를 탐지 및 차단할 때 분석 비용을 감소시킬 수 있는 권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션을 제안한다.

1. 서론

최근 국내·외 스마트폰 시장이 급성장함에 따라 스마트폰을 목표로 하는 악성 프로그램 또한 급격하게 증가하고 있다. 2009년 이후 모바일 악성코드의 숫자가 급격하게 증가하였으며, 2006년을 기준으로 스마트폰 기반의 악성코드는 약 500%가량 증가하였다. 이처럼 급격하게 증가하고 있는 스마트폰 악성코드는 직접적으로 금전적인 피해를 입히거나 스마트폰에 담겨 있는 개인정보를 유출함으로써 큰 피해를 발생시키고 있다[1, 2].

전 세계 스마트폰 운영체제 점유율의 52.5%를 차지하고 있는 안드로이드는 2010년에 발견된 악성코드가 0.5%에 불과하였으나, 2011년에는 발견된 악성코드가 46.7%로 1년 사이 엄청나게 증가하였다[3, 4]. 현재, 안드로이드의 악성코드 발견 수 또한 급격하게 증가하고 있다. 이처럼 안드로이드 운영체제의 악성코드를 탐지하고 차단하기 위해 많은 안드로이드 안티바이러스 어플리케이션이 출시되고 있지만, 이러한 안티바이러스 어플리케이션들은 전체 시그니처를 분석하여 탐지하기 때문에 높은 분석 비용이 들게 된다. 이에 따라 본 논문에서는 한정된 권한의 시그니처 분석으로 분석 비용을 감소시킬 수 있는 권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 안드로이드와 안드로이드 어플리케이션 파일 구조 및 보안 위협에 대해 설명하고, 3장에서는 권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션의 제안 방식과 동작 절차를 제안하다. 마지막으로 4장에서는 결론을 맺는다.

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임
(No. 2012-010886)

2. 관련 연구

2.1 안드로이드

안드로이드(Android)는 휴대 전화를 비롯한 휴대용 장치를 위한 운영 체제와 미들웨어, 사용자 인터페이스 그리고 표준 응용 프로그램(웹 브라우저, 이메일 클라이언트, 단문 메시지 서비스(SMS), 멀티미디어 메시지 서비스(MMS) 등)을 포함하고 있는 소프트웨어 스택이자 모바일 운영 체제이다. 안드로이드는 개발자들이 자바 언어로 응용프로그램을 작성할 수 있게 하였으며, 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다. 또한 안드로이드 소프트웨어 개발 키트(SDK : Software Development Kit)를 통해 응용프로그램을 개발하기 위해 필요한 각종 도구들과 응용프로그램 프로그래밍 인터페이스(API)를 제공한다[5].

2.2 안드로이드 어플리케이션 파일 구조 및 보안 위협

Apk(Application Package) 파일은 안드로이드 패키지 파일로 안드로이드 플랫폼에 어플리케이션을 설치 시 사용하게 된다. Apk 파일은 알집이나 7-Zip 같은 압축 프로그램으로 간단히 확인할 수 있다. 그리고 일반적으로 권한 명시를 위한 AndroidManifest.xml 파일과 실제적인 실행 파일인 dex 파일, 그 외의 리소스 파일 등으로 구성되어 있다. 이에 따라 안드로이드 어플리케이션의 설치 과정은 먼저, 안드로이드 어플리케이션 개발 환경을 통하여 apk 파일을 생성하고, 이를 안드로이드 환경에 설치하면 압축이 풀리게 된다. 이 중 실제 실행 파일인 dex 파일이 안드로이드 어플리케이션 동작을 위한 가상 머신인 달빅 VM(Virtual Machine)에서 동작하게 된다.

안드로이드 악성코드는 디바이스에 설치되어야만 동작을 수행하는데 일반적으로 설치파일 내부에 존재하며, apk 설치 시 악성행위를 수행하게 된다. 안드로이드 어플리케이션의 악성코드는 실행 파일인 dex 파일에 존재를 하게 되며, 악성 행위를 하기 위한 악성코드는 해당 목적에 따라 SMS. WIFI, 위치정보, 웹 URL 연결 등의 악성 행위를 수행하기 위한 권한정보를 가지고 있어야 동작이 가능하다. 즉, 악성행위를 하기 위한 악성코드는 AndroidManifest.xml 파일에 악성행위에 해당하는 특정 권한정보를 가지고 있어야 동작이 가능하다.

또한 안드로이드의 경우에는 설치되어 있는 어플리케이션을 수정하고 배포하는 것이 가능하기 때문에 악성코드를 삽입한 어플리케이션을 블랙 마켓에 배포하여 다른 사용자들이 어플리케이션을 설치하게끔 유도한다. 사용자들이 블랙 마켓에서 악성코드를 삽입한 어플리케이션을 설치하게 되면, 악성코드에 감염되어 스마트폰 사용자들의 금융사고, 개인정보 노출 등과 같은 보안 위협이 발생하게 된다. 이에 따라 악의적인 사용자가 안드로이드 어플리케이션을 이용하여 dex 파일에 악성코드를 삽입하고 AndroidManifest.xml 파일에 권한정보를 수정하여 배포를 하게 되면, 해당 안드로이드 어플리케이션은 악성코드를 삽입한 어플리케이션으로 변경이 된다.

2.3 안드로이드 안티바이러스 어플리케이션

2.3.1 V3 Mobile

AhnLab의 V3 Mobile은 안드로이드 OS를 위한 안티바이러스 어플리케이션으로 실시간 바이러스 탐지와 치료를 담당한다. 구체적인 기능으로는 원격 잠금, 삭제, 위치 추적 등 도난방지 기능과, 스팸차단과 로깅 기능, 네트워크 접속 제어 기능을 제공한다. V3 Mobile은 시그니처 기반의 엔진 업데이트를 통해 신규 악성코드에 대응이 가능하다[6].

2.3.2 nProtect Mobile

잉카인터넷은 지난 2010년 8월 안드로이드용 안티바이러스 어플리케이션인 nProtect Mobile을 개발하여 국내 안티바이러스 개발업체 중 최초로 구글 안드로이드 마켓에 등록하였다. nProtect Mobile의 주요 기능으로는 설치 및 실행중인 악성 어플리케이션의 탐지 및 삭제, 사용자 예약을 통한 자동실행 기능 등이 있다[6].

2.3.3 알약 안드로이드

ESTsoft가 개발한 알약 안드로이드는 안드로이드 OS에 최적화된 안드로이드 안티바이러스 어플리케이션으로 악성 코드를 내포하고 있을 것으로 예상되는 어플리케이션을 정밀 검사하여 치료하며, 실시간으로 위험 어플리케이션 설치 여부를 확인 후 알림 서비스를 제공한다. 이외에도 어플리케이션 보안 위협, 등급 안내 및 실시간 감시, 검사로그, 스팸문자, 스팸전화 차단 및 실시간 감시, 실행중인 어플리케이션 종료, 3G/Wifi 네트워크 연결 설정 등 다양한 기능을 제공한다[6].

2.3.4 ViRobot Mobile

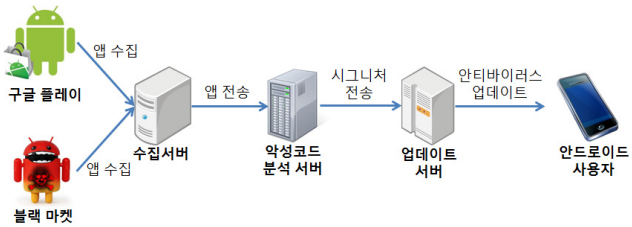
HAURI의 ViRobot Mobile은 스마트폰에 최적화하여 개발한 안드로이드 통합 보안 솔루션이다. ViRobot Mobile은 악성코드 진단뿐만 아니라 스팸차단, 네트워크감시, 도난방지 등의 기능을 제공하고 있다[6].

3. 권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션

3.1 제안 방식

안드로이드 악성코드는 2.2절에서도 언급한 바와 같이 특정 악성행위를 실행하기 위해서 AndroidManifest.xml 파일에 악성행위에 해당하는 특정 권한정보를 가지고 있어야 동작이 가능하다. 이에 따라 본 논문에서 제안하는 방식은 권한정보에 따른 악성코드 시그니처 탐지 기법이다.

전체적인 안드로이드 안티바이러스 수집 및 분석 구성도는 (그림 1)과 같다.



(그림 1) 안드로이드 안티바이러스 수집 및 분석 구성도

수집서버에서는 구글 플레이나 블랙 마켓에서 앱을 수집하여 이를 악성코드 분석 서버에 전송한다. 먼저, 악성코드 분석 서버에서는 수집한 앱의 AndroidManifest.xml 파일에서 패키지명, 버전코드, 버전이름, 권한정보를 수집하고, 실행 파일인 dex 파일에서 악성코드를 분석하는데 악성코드를 분석하게 되면 그에 따른 악성코드 탐지 시그

니처를 생성한다. 그리고 생성된 악성코드 탐지 시그니처들은 해당 특정 행위에 따라 SMS, WIFI, 위치정보, 웹 URL 연결 등의 권한정보에 따른 각각의 시그니처로 분류한다. (그림 2)는 권한정보 관리를 통한 악성코드 분석 과정이다. 이후 권한정보에 따라 분류된 시그니처와 AndroidManifest.xml 파일에서 수집한 패키지명, 버전코드, 버전이름, 권한정보를 수집한 파일을 업데이트 서버에 전송한다. 업데이트 서버는 안드로이드 안티바이러스 어플리케이션을 설치하고 있는 사용자에게 안티바이러스 정보를 업데이트 한다.

(그림 2) 권한정보 관리를 통한 악성코드 분석 과정

3.2 제안하는 안드로이드 안티바이러스 어플리케이션 동작 절차

(그림 3)은 제안하고자 하는 안드로이드 안티바이러스 어플리케이션 동작 절차이며, 자세한 동작 절차는 다음과 같다.

① 안티바이러스 어플리케이션을 실행하여 안티바이러스 탐지 및 차단을 실행하게 되는데 검사하고자 하는 파일이 어플리케이션인지 아닌지를 판별한다. 만약 안드로이드 어플리케이션 파일이 아닌 경우 전체 시그니처를 비교하여 악성코드를 탐지한다.

② 안드로이드 어플리케이션인 경우에는 기존에 수집된 패키지명, 버전코드·이름이 변경되었는지를 확인한다.

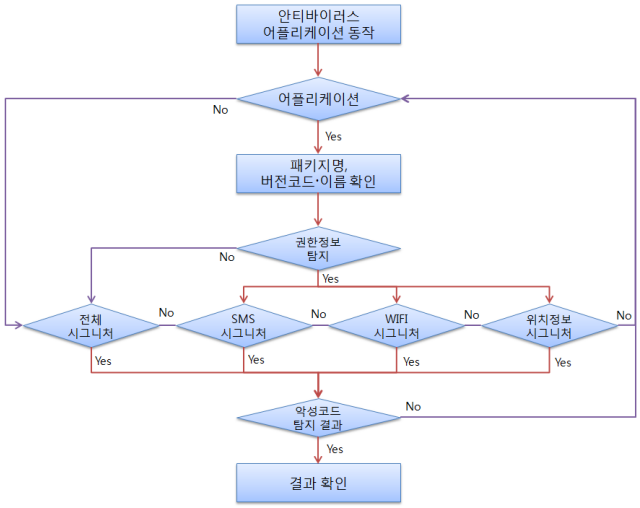
③ 그리고 기존에 수집된 권한정보와 검사하고자 하는 안드로이드 어플리케이션의 권한정보를 비교한다. 만약 안드로이드 어플리케이션의 권한정보가 없을 시에는 전체 시그니처를 비교하여 악성코드를 탐지한다.

④ 권한정보가 확인이 되면 해당 권한정보에 따른 시그니처들(SMS 시그니처, WIFI 시그니처, 위치정보 시그니처, 웹 URL 연결 시그니처 등)을 각각 비교하여 악성코드를 탐지한다.

⑤ 전체 시그니처, SMS 시그니처, WIFI 시그니처, 위치정보 시그니처, 웹 URL 연결 시그니처 등에서 악성코드가 탐지되지 않으면, 다시 검사할 다른 파일을 찾아 해당 파일이 어플리케이션인지 아닌지를 검사하게 된다.

⑥ 전체 시그니처, SMS 시그니처, WIFI 시그니처, 위치정보 시그니처, 웹 URL 연결 시그니처 등에서 악성코드가 탐지되면 탐지 결과를 사용자에게 알려준다. 그리고 나서 검사할 다른 파일이 있다면 해당 파일이 어플리케이션인지 아닌지를 검사하게 된다.

⑦ 검사할 다른 파일이 없다면 탐지한 결과를 사용자에게 알려준다.



(그림 3) 제안하는 안드로이드 안티바이러스 어플리케이션 동작 절차

4. 결론

본 논문에서는 권한정보 관리를 통한 안드로이드 안티바이러스 어플리케이션을 제안하였다. 기존의 안드로이드 안티바이러스 어플리케이션들은 악성코드를 탐지 및 차단하기 위해서 전체 시그니처를 비교하여 탐지 및 차단하지만, 제안한 안드로이드 안티바이러스 어플리케이션은 탐지하고자 하는 파일을 권한정보에 따라 분류된 악성코드 시그니처들과 각각 비교하여 악성코드를 탐지 및 차단한다. 따라서 기존의 안드로이드 안티바이러스 어플리케이션을 사용하는 것 보다 제안한 안드로이드 안티바이러스 어플리케이션을 사용하게 되면 시그니처 분석 비용이 감소되어 스마트폰 하드웨어 자원 사용량 또한 감소하게 된다. 본 논문에서 제안하는 안드로이드 안티바이러스 어플리케이션을 통해 보다 향상된 안드로이드 안티바이러스 어플

리케이션이 출시될 것으로 판단되며, 향후 연구 및 방향으로 제안한 안드로이드 안티바이러스 어플리케이션의 구현을 통한 검증 및 실험이 이루어져야 될 것으로 생각된다.

참고문헌

[1] 최윤희, 최은만, “안티 패턴 기반의 정적 분석을 이용한 안드로이드 어플리케이션 취약점 분석”, 정보과학회논문지, 제18권 제4호, pp. 316-320, 2012. 4.

[2] 심원태, 김종명, 류재철, 노봉남, “안드로이드 앱 악성행위 탐지를 위한 분석 기법 연구”, 정보보호학회논문지, 제21권 제1호, pp. 213-219, 2011. 2.

[3] 김동우, 조형진, 류재철, “안드로이드 모바일 플랫폼 환경에서 Radio Interface Layer를 통한 악성행위 및 대응 방안”, 한국컴퓨터종합학술대회 논문집, 제39권 제1호, pp. 221-223, 2012. 6.

[4] 민승욱, 조형진, 신진섭, 류재철, “머신러닝 기법을 이용한 안드로이드 악성코드 탐지 기법”, 한국컴퓨터종합학술대회 논문집, 제39권 제1호, pp. 280-282, 2012. 6.

[5] http://www.android.com/

[6] 한국인터넷진흥원, “스마트폰 백신 이용 안내서”, 2011.