안드로이드 환경에서 악성 어플리케이션 탐지를 위한 검증 시스템 제안[1)]

조효제*, 이광형, 염흥열**
*순천향대학교 정보보호학과
**순천향대학교 정보보호학과
e-mail:sfkino@gmail.com

A proposed verification system for detecting malicious applications in the Android applications enviroinment

Hyo-Je Jo*, Kwang-Hyung Lee, Heung-Youl Yeom*
*Dept of Information Security, SoonChunHyang University

요 약

스마트폰이 보급화 되면서 많은 사람들이 스마트폰을 사용하게 되었고 그중에서도 안드로이드는 높은 사용률로 여러 악성 어플리케이션들의 타겟이 되고 있다. 본 고에서는 앞의 문제점을 방지하기 위해 안드로이드의 보안기법을 소개하고 정상적인 마켓을 통하지 않고 인터넷 등에서 받은 보안성이 검증되지 않은 어플리케이션 등에 의한 피해를 방지하기 위한 검증 시스템을 제안한다

1. 서론

스마트폰이 보급화 되지 않았던 과거와 달리 스마트폰의 기술이 빠르게 발전하면서 그 편의성 및 기능성이 높아졌고 그와 함께 스마트폰의 보급률이 엄청나게 높아지게 되었다. 스마트폰들은 다양한 운영체제들을 탑재하고 있으며 스마트폰 운영체제중 가장 대표적인 운영체제들로는 안드로이드, 윈도우 모바일, IOS등이 존재한다. 각 운영체제들은 다양한 현재 스마트폰은 높은 보급률을 자랑하고 있으며 그중에서도 특히 안드로이드는 높은 개방성과 이식성으로 인해 현재 50%에 가까운 시장 점유율을 가지고 있다. 특히 안드로이드는 특유의 개방성과 다양한 기기에 포팅되어 사용자 선택이 넓어 많은 사용자들에게 각광받고 있다. 특히 스마트폰은 Wifi나 3G 네트워크를 이용한 인터넷 엑세스가 가능하다는 점과 앱을 사용할 수 있다는 두가지 큰 특징을 가진다, 특히 앱이라 불리는 어플리케이션들은 사용자에 의해 개발된 어플리케이션들로 스마트폰의을 이용한 다양한 작업을 가능하게 하는 큰 역할을 한다. 하지만 이러한 어플리케이션들을 악의적인 목적으로 개발해 이를 이용한 해킹 사례 등의 문제점이 대두되게 되었고 특히 안드로이드의 경우 인터넷으로 다운받은 어플리케이션 파일인 APK를 자유롭게 설치할 수 있고 마켓에 업로드 되는 어플리케이션에 대한 검증과정이 없어 악성 어플리케이션에 많이 노출되어 있다. 본 고에서는 마켓에 업로드 되는 어플리케이션들에 대한 보안성 검증 시스템을 제안한다. 2장에서는 제안하는 검증시스템을 위한 배경연구를 서술한 후 3장에서는 제안하는 시스템의 구조와 검증과정을 설명하고 4장에서 결론을 맺는다.

1) 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2010-0025393)

* 주 저자. sfkino@gmail.com
* 교신저자. hyyoum@sch.ac.kr

2. 안드로이드 보안기술

1.1 킬 스위치(Kill-Switch)[1]

킬 스위치는 거의 모든 스마트폰 OS에 도입되어 있는 기술로 사용자의 단말기에 설치된 어플리케이션을 강제적으로 삭제할 수 있는 기능을 의미한다. 안드로이드의 경우 안드로이드 마켓의 약관에 킬 스위치에 대해 명시하고 있으며 삭제된 어플리케이션에 대해서는 24시간 이내에 환불하도록 조취하고 있다.

1.2 바운서(Bouncer)

2012년 2월 구글에서 공개한 바운서[2]는 안드로이드 마켓에 게시된 어플리케이션을 대상으로 적용되는 보안기술이다. 바운서는 공식 안드로이드 마켓에 게시된 어플리케이션이나 새롭게 게시된 어플리케이션을 대상으로 기존에 알려진 악성코드가 포함되어 있는 지를 확인하고 어플리케이션의 오작동으로 인한 취약점 발생 가능성 등을 검사하며 숨겨진 악성행위를 탐지하기 위해 어플리케이션

의 동작을 시뮬레이션 한다. 구글에서는 이 기술을 적용하여 2011년 상반기동안 악성 앱의 다운로드가 약 40%정도 감소한 것으로 확인 되었다.

1.3 코드 사이닝(Code-signing)[3]

개발자에 의해 개발된 모든 어플리케이션에는 인증서를 이용한 서명이 필요하다.. 이는 응용프로그램의 신뢰성을 보장하는 데에 사용되고 어플리케이션 제어 등에는 사용되지 않아 코드사이닝을 통해 올바른 개발자로부터 개발된 어플리케이션인지 확인한다.

1.4 기타

위에서 명시된 보안기술 외에도 샌드박싱(Sandboxing), 퍼미션(Permission) 등의 기술을 통해 보안성을 향상시키고 있다.

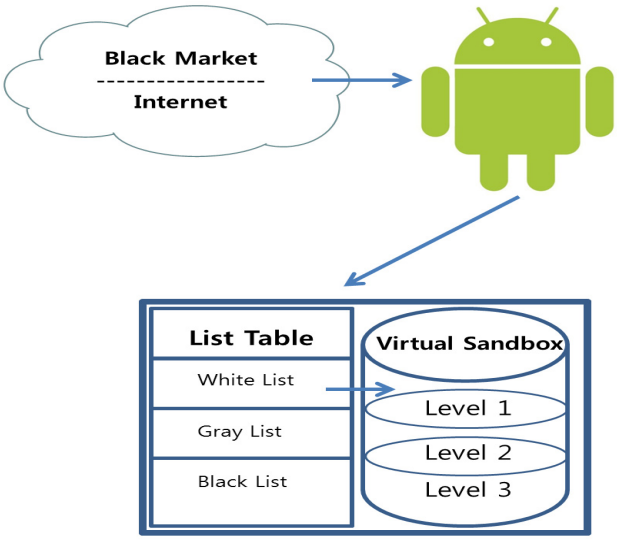
샌드박싱은 어플리케이션이 동작할 때 타 어플리케이션 등에 영향을 받지 않도록 실행되는 어플리케이션 단위로 구분하여 타 어플리케이션이 실행중인 다른 어플리케이션의 접근을 막아 어플리케이션이 악성코드 등에 의한 위협을 줄일 수 있다.

퍼미션은 안드로이드에서 사용되는 권한기반의 제어 기법으로 어플리케이션이 설치될 때 어플리케이션이 사용하게 되는 권한을 사용자가 확인할 수 있게 하며 권한이 명시되지 않은 기능은 사용할 수 없게 한다. 하지만 이는 전적으로 사용자에게 의존하고 있기 때문에 사용자가 권한에 대해 확실한 확인 없이 설치하게 될 경우 아무런 효과를 내지 못한다는 단점을 가진다.

3. 어플리케이션 검증 시스템 제안

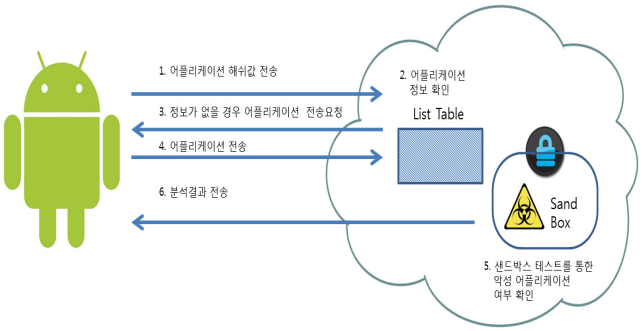
지금까지의 안드로이드 공식마켓에는 코드사이닝을 제외한 다른 보안정책이 존재하지 않아 마켓에 다수의 악성 어플리케이션들도 포함되어 있었다. 하지만 이는 12년 2월 구글이 발표한 안드로이드 마켓 보안을 위한 바운서(bouncer)가 공개되면서 공식마켓에 존재하는 악성 어플리케이션을 탐지하기 시작했다. 하지만 이 바운서는 안드로이드 공식마켓에 국한되어 온라인이나 블랙마켓 등을 통해 다운받은 불법 APK파일등의 경우 탐지해 낼 수 없다는 단점을 가진다. 본 논문에서 제안하는 검증 시스템은 이미 게시된 마켓의 어플리케이션이 아닌 인터넷이나 불법적인 통로를 통해 다운로드 된 안드로이드 파일 검증 시스템을 제안한다. 인터넷 등에서 다운받은 어플리케이션의 경우 바운서 등의 보호를 받을 수 없으므로 제안하는 방식을 통해 마켓을 거치지 않은 어플리케이션에 대한 안전성을 검증한다.

제안하는 프레임워크의 구성은 (그림 1)과 같이 APK파일을 다운받는 블랙마켓이나 인터넷 등을 제외한 스마트



(그림 1) 프레임워크의 구성요소

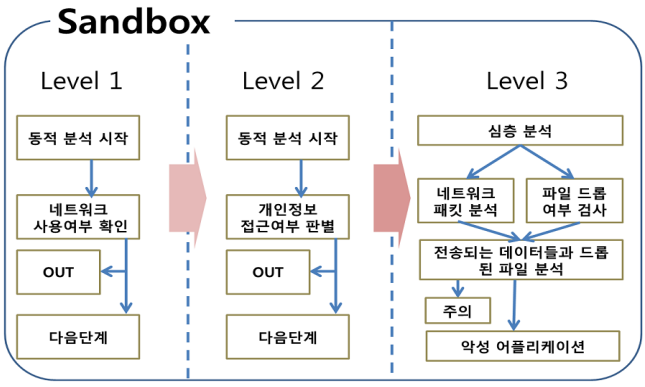
폰의 모듈과 검증시스템의 리스트 테이블, 가상 샌드박스 등이 존재한다. 스마트폰 모듈은 스마트폰에서 어플리케이션을 설치할 때 마켓에 게시된 정상적인 어플리케이션인지 확인하고 등록되지 않은 어플리케이션에 대해서 분석서버로 파일의 넘기는 등의 동작을 수행하는 모듈이다. 리스트 테이블은 기 분석된 알려지지 않은 어플리케이션의 분석결과를 모아놓는 테이블로 안드로이드 환경에서 검증되지 않은 서버 접근제어 기법연구[4]에서 사용하였던 리스트 서버의 구조를 채용하여 어플리케이션의 분석결과에 따라 위험도를 3 단계로 나누어 관리한다. 가상 샌드박스는 가상의 안드로이드 환경으로 분석을 의뢰받은 어플리케이션에 대해 단계별 분석을 진행해 어플리케이션의 악성행위 여부를 판단해 주고 이 결과를 리스트 테이블 등에 저장하는 등의 동작을 수행한다.



(그림 2) 전체적인 동작과정

전반적인 프레임워크의 동작과정은 (그림2)와 같다. 사용자가 인터넷이나 불법적인 경로를 통해 다운받은 실행파일을 설치할 때 시스템은 설치 파일의 해쉬값을 검증서버로 보내 이전의 검사여부를 확인하고 검사되지 않은 어플리케이션의 경우 안드로이드 환경으로 구축된 앱 분석

서버로 어플리케이션을 전송한다. 앱 분석 서버는 안드로이드 환경에서 어플리케이션 권한 및 동작 과정등 총 세 가지 단계의 테스트를 통해 악의적인 어플리케이션 여부를 판별하고 판별여부와 결과를 리스트에 등록한다. 정상적인 어플리케이션일 경우 설치를 허용하고 불법적인 행위를 하는 것으로 의심이 되는 경우 사용자에게 경고메시지를 전송한다. 이 때 가장 중요한 것은 앱 분석 서버에서 분석하게 되는 과정으로 이 과정은 (그림 3)과 같으며 이 과정은 총 세 단계로 나누어지게 된다.



(그림 3)

첫 번째 단계에서는 전송받은 어플리케이션 파일의 권한을 검사하여 네트워크 활동여부를 판별한다. DDoS 공격을 시도하거나 네트워크를 통해 스마트폰이 가진 정보를 탈취할 경우 네트워크를 활용해야만 하므로 이를 위해 기본적으로 네트워크 접근권한을 검사하게 된다. 이 과정을 통해 네트워크 활동을 하지 않는 경우 악의적인 목적을 가지지 않은 애플리케이션으로 판단하고 바로 마켓에 게시한다.

두 번째 단계에서는 네트워크 활동을 하는 어플리케이션이 사용자의 민감한 정보에 접근하게 될 경우 개인정보가 유출될 수 있으므로 애플리케이션이 사용자의 중요한 개인정보의 접근여부를 검사한다. 안드로이드에서는 어플리케이션에서 사용되는 권한에 대한 명시하도록 되어있으며 권한이 명시되지 않은 자원을 어플리케이션이 요청할 경우 어플리케이션이 종료된다. 이렇게 필요한 자원에 접근하기 위해서는 반드시 해당 권한이 명시되어야 하며 이를 이용해 민감정보의 접근여부를 확인할 수 있다. 안드로이드에서 명시되는 권한은 매우 많지만 이들 중 두 번째 단계에서 확인하는 정보는 개인정보 접근, 하드웨어 정보 접근 등이있다. 개인정보의 경우 사용자의 중요한 개인 데이터에 접근해 이를 유출 할 가능성이 존재하므로 이에대한 접근권한을 확인하고 하드웨어 제어는 사용자의 단말기기의 하드웨어를 제어하여 녹화, 녹음 등이 가능하기 때문이다.

위와 같이 네트워크 접근과 개인정보의 접근여부 모두 포함되는 어플리케이션일 경우 위협가능성이 있는 어플리케이션으로 판단하고 어플리케이션의 동작을 기반으로 분석을 하게 된다. 안드로이드 환경으로 구축된 시뮬레이터에 어플리케이션을 설치하고 네트워크를 통해 전송되는 정보, 시스템에 미치는 영향 등을 분석한다. 네트워크를 통해 전송되는 정보의 경우 단말기기에 저장되어 있는 사용자 정보를 네트워크를 통해 전송하는지를 확인하고 단말기 영역에 새로운 실행파일 등을 다운로드 하는지의 여부를 검사한다. 시스템 영역에서는 어플리케이션이 시스템의 루트권한을 갖게 하는 공격코드가 포함되어 있는지를 확인한다. 루트권한을 갖게 하는 공격코드가 포함될 경우 시스템의 중요한 파일들에 허가 없이 접근이 가능하게 되며 이를 통한 2차적인 피해가 발생할 수 있기 때문이다.

앞의 두 단계를 통과하고 마지막 세 번째 단계에서 악의적인 행위가 발견되지 않을 경우 주의가 필요한 어플리케이션으로 분류하고 리스트 테이블에 주의가 필요한 어플리케이션으로 저장하며 검사시 악의적인 행동이 발견되면 악성코드로 분류하여 사용자에게 경고메시지를 보내고 파일의 해쉬값을 리스트 테이블에 저장하여 차후 같은 어플리케이션에 대한 검사요청이 올 경우 리스트테이블에서 검사결과를 검색해 사용자에게 전송해 준다.

4. 결론

본 논문에서는 안드로이드 환경에서 악의적인 행위를 하는 어플리케이션을 탐지하는 기법을 제안하였다. 알려지지 않은 어플리케이션을 설치할 때 어플리케이션의 정보를 리스트 테이블에 조회해 악성코드의 여부를 확인하고 존재하지 않을 경우 단계별 동적 분석을 통해 어플리케이션의 동작을 탐지해 악성 어플리케이션 여부를 판별한다. 이와 같은 방법으로 알려지지 않은 악성코드에 대한 분석이 가능하며 이를 통해 보안성 향상이 가능하다. 하지만 리스틀 테이블에 저장되지 않은 어플리케이션의 경우 파일 전송 및 즉각적인 대응 등에 어려움이 있으며 정상적인 파일에 대한 오탐 등의 문제가 발생할 수 있다. 향후 연구에서는 앞에서 언급한 문제점을 개선해 좀 더 효율성이 높은 검증 시스템에 대한 연구가 필요 할 것이다.

참고문헌

[1]http://blog.naver.com/PostView.nhn?blogId=huewu&logNo=110130711671&parentCategoryNo=18&viewDate=&currentPage=1&listtype=0&from=postList

[2] http://googlemobile.blogspot.com/2012/02/android-and-security.html

[3] http://developer.android.com/guide/publishing/app-signing.html

[4] 조효제, 정영곤, 장기헌, 염홍열 "안드로이드 환경에서 검증되지 않은 서버 접근제어 기법 연구" 정보보호학회 춘계학술대회, 2011