情報保護學會論文誌
第 22 卷 第 5 號, 2012. 10

안드로이드 권한과 브로드캐스트 인텐트 매커니즘의 사용 현황 및 보안 취약성 분석*

김 영 동,† 김 익 환, 김 태 현‡
서울시립대학교 기계정보공학과

Analysis of Usage Patterns and Security Vulnerabilities in Android Permissions and Broadcast Intent Mechanism*

Young-dong Kim,† Ikhwan Kim, Taehyoun Kim‡
Mechanical and Information Engineering, University Of Seoul

요 약

구글 안드로이드 플랫폼은 응용프로그램 권한을 이용해 시스템 자원이나 다른 응용프로그램의 컴포넌트 사용을 제어하는 보안 모델을 사용하고 있다. 그러나, 권한 기반 보안 모델에는 사용자의 이해 부족과 개발자의 과도한 권한 요청으로 인한 취약성이 존재한다. 또한 응용프로그램의 내부 컴포넌트간 통신수단인 브로드캐스트 인텐트 메시지의 경우도 시스템 내에서 광범위하게 사용되는 데 반해 이와 관련된 보안정책이 없다는 문제점이 있다. 본 연구에서는 응용프로그램 권한과 브로드캐스트 인텐트 매커니즘과 관련된 보안 침해 시나리오를 제시하고, 실제 안드로이드 마켓의 일반 응용프로그램들과 악성 응용프로그램을 대상으로 권한과 브로드캐스트 인텐트의 사용 현황을 분석한다. 분석을 통해 기존 악성 프로그램의 권한 요구사항과 브로드캐스트 인텐트 리시버의 등록 사항에 관한 특성 집합을 추출할 수 있었다. 본 연구에서는 이 결과를 바탕으로 설치 시점에 악성 프로그램의 특성 집합을 이용해 악성 프로그램일 가능성이 높은 프로그램들을 추출하여 사용자에게 공지할 수 있는 안드로이드 플랫폼 수정 방안을 제안한다.

ABSTRACT

Google Android employs a security model based on application permissions to control accesses to system resources and components of other applications from a potentially malicious program. But, this model has security vulnerabilities due to lack of user comprehension and excessive permission requests by 3rd party applications. Broadcast intent message is widely used as a primary means of communication among internal application components. However, this mechanism has also potential security problems because it has no security policy related with it. In this paper, we first present security breach scenarios caused by inappropriate use of application permissions and broadcast intent messages. We then analyze and compare usage patterns of application permissions and broadcast intent message for popular applications on Android market and malwares, respectively. The analysis results show that there exists a characteristic set for application permissions and broadcast intent receiver that are requested by typical malwares. Based on the results, we propose a scheme to detect applications that are suspected as malicious and notify the result to users at installation time.

Keywords: Android, permission, broadcast intent, security vulnerability

접수일(2012년 9월 10일), 수정일(2012년 10월 17일), 게재확정일(2012년 10월 17일)

* 본 논문은 2012년 한국인터넷진흥원 "시큐어코딩 기반 SW 개발보안 기반기술 연구" 위탁과제의 연구결과로 수행되었음(KISA-2012-024).

† 주저자, ydkim@uos.ac.kr
‡ 교신저자, thkim@uos.ac.kr

I. 서 론

구글 안드로이드 플랫폼은 오픈 소스 플랫폼 특성으로 인하여 제조사, 통신사업자, 일반 사용자 등이 다양하게 커스터마이징을 할 수 있는 장점을 바탕으로 2012년 2분기 전세계 스마트폰 시장에서 68.1%의 시장 점유율로 가장 높은 비중을 차지하고 있다[1]. 하지만 이러한 성장세와 비례하여 안드로이드 관련 악성 응용프로그램도 2011년 하반기 7개월 동안 약 3,325% 증가하였으며[2], 개인정보 수집, 무단요금 부과 등의 악성코드가 2011년 상반기 대비 17배 증가하는 추세를 보이고 있다[3].

안드로이드 플랫폼의 경우 리눅스 운영체제의 사용자/그룹 ID 기반의 접근 권한 제어, 자바 가상기계의 사용에 따른 샌드박싱, 안드로이드 고유의 권한 기반 보호 모델 등을 이용한 보안 정책을 제공하고 있으나, [4]에서 제시한 바와 같이 다양한 형태의 보안 취약성이 존재한다. 본 연구에서는 이들 중 안드로이드 보안 모델의 기본인 응용프로그램 권한(permission)과 응용프로그램을 구성하는 컴포넌트간 기본 통신수단인 인텐트 메커니즘과 관련된 개인정보 유출 시나리오와 이에 대한 대책을 일반 응용프로그램과 악성 응용프로그램들의 특성을 이용해 분석, 제시하고자 한다. 이를 위해 응용프로그램 권한과 브로드캐스트 인텐트에 대한 보안 취약 시나리오를 제시하고 실제 안드로이드 플랫폼에서 그 위험성을 확인하였다. 또한, 실제 안드로이드 마켓에 등재된 인기 상위 응용프로그램과 잘 알려진 악성 응용프로그램의 권한 요구사항과 브로드캐스트 인텐트에 관한 사용 현황을 분석하고 이를 토대로 개발 과정에서의 유의사항과 플랫폼 수정을 통한 해결책을 제시한다.

본 논문의 구성은 다음과 같다. II장에서는 안드로이드 플랫폼의 보안 관련 특징과 관련 연구를 소개한다. III장에서는 개인정보 유출과 관련된 보안 취약 시나리오를 제시한다. 또한, 실제 응용프로그램과 악성 응용프로그램의 특성을 분석한 결과와 이에 따른 해결책을 제안한다. 마지막으로 IV장에서 결론과 향후 과제를 제시한다.

II. 연구배경

안드로이드 고유의 보안정책은 특정 응용프로그램이 시스템 내부의 데이터, 하드웨어 자원, 또는 다른 응용프로그램의 컴포넌트를 이용하려고 할 때 적절한 권한을 획득하여야만 이를 허용하도록 하는 개념인 "응용프로그램 권한 (permission)" 기반 모델을 사용한다. 권한은 특정한 문자열 형태로 정의되며, 시스템 차원에서 이미 정의되어 있거나 응용 개발자가 자신의 컴포넌트 보호를 위해 새롭게 정의한 권한으로 구분할 수 있다. 권한을 획득하기 위해서는 먼저 개발자가 응용프로그램 패키지에 포함된 매니페스트 (AndroidManifest.xml) 파일에 〈uses-permission〉 태그를 이용해 요구하는 권한명을 지정해야 한다. 이후 응용프로그램 설치 과정에서 기기 사용자가 설치를 허용하면 패키지 설치를 담당하는 시스템 응용의 권한 검증 과정을 거쳐 최종적으로 매니페스트 파일에 기재된 권한들이 승인된다. 다른 측면으로는 응용프로그램 내의 컴포넌트들을 외부의 허가되지 않은 접근으로부터 권한을 이용해 보호할 수도 있다. 권한을 이용한 보호는 시스템 정의 권한 또는 개발자가 직접 정의한 권한들을 매니페스트 파일의 〈permission〉 태그에 명시함으로서 이를 획득한 외부 프로그램만의 접근을 허용하도록 한다.

권한모델에 기반한 보안정책은 안드로이드 보안의 핵심이지만 시스템 관점으로 볼 때 단순한 문자열일 뿐이며 결국은 전적으로 사용자의 선택에 보안을 맡기는 정책이다. 응용프로그램 설치시 사용자가 선택할 수 있는 항목은 요구된 권한 전부를 허용하거나 불허하는 all-or-nothing 방식이며, 일단 응용프로그램이 설치되고 나면 응용이 요구한 권한들이 삭제 전까지 계속 유지되므로 현재 구현에서는 런타임에 과도한 권한 사용을 제한할 방법이 없다. 특히 최근의 응용프로그램들은 다양한 기능 제공을 위해 상당히 많은 수의 권한을 요구하는 데 반해, 사용자들이 가지는 권한 남용에 따른 보안 취약성에 대한 인식 수준은 상당히 낮은 수준이다. 안드로이드 플랫폼에서의 권한 보안모델의 문제점을 보완하기 위한 기존 연구는 다음과 같다. [5]에서는 권한 보호 수준변경 위험성 및 권한명 노출에 따른 문제점을 제기하며 플랫폼 수정을 통해 선택적 권한 허용 기법을 제안하였으며, [6]-[8]에서는 악성 응용프로그램들의 권한 요구사항을 분석하여 악성 프로그램 탐지에 사용하고자 하였다.

한편, 안드로이드 응용프로그램을 구성하는 주요 컴포넌트간 통신은 인텐트 (intent)라는 메시지 객체 전송 메커니즘을 통해 이루어진다. 인텐트의 전달 방법은 수신할 컴포넌트 이름의 기술유무에 따라 명시적 인텐트 (explicit intent)와 암시적 인텐트 (implicit intent)로 구분된다. 명시적 인텐트는 인텐트 송신자

가 해당 인텐트를 수신할 컴포넌트 이름을 직접 명시하도록 되어 있어서 컴포넌트 이름을 정확하게 알기 힘든 외부 응용프로그램보다는 동일 응용프로그램 내의 다른 컴포넌트의 동작을 구동하는 데 사용된다. 반면에, 암시적 인텐트를 사용할 때는 수신자 컴포넌트의 이름 대신 동작의 특징만을 기재하여 전송하므로 다른 응용프로그램 혹은 시스템에 내장된 프로그램 내의 컴포넌트를 구동하는 데 주로 사용된다. 암시적 인텐트의 경우, 시스템에 등록된 인텐트 필터 (intent filter)를 검색하여 최적의 수신 컴포넌트를 결정하도록 한다. 인텐트 필터는 특정 컴포넌트가 자신이 잠재적으로 어떤 특성을 가진 인텐트를 수신해 처리할 수 있는지를 시스템에 등록하는 메커니즘이다.

인텐트 메커니즘은 안드로이드 시스템 내에서 광범위하게 사용되고 있는데 반해 이와 관련해 특별한 보안 정책이 없으며 특히 암시적 인텐트의 경우 관련된 인텐트 필터를 등록한 컴포넌트가 여러 개일 경우 수신자 경합 상황이 발생한다. 이 때 실제 인텐트 수신자의 결정은 사용자의 판단 혹은 수신 프로그램이 시스템에 설치된 시간 순서에 따르게 되어 있어서 해당 인텐트가 의도하지 않은 악의적 응용프로그램에 전달된 후 그 내용이 쉽게 유출될 수 있는 취약성이 존재한다. 이와 관련된 보안 침해 시나리오는 이전 연구 [4]에 자세히 기술되어 있다. 한편 시스템 부트업 완료, 배터리 현황, SMS 수신 등 시스템 차원에서 전송되는 브로드캐스트 인텐트는 더욱 더 보안상 문제가 될 수 있다. 이러한 브로드캐스트 전달은 순서화된 브로드캐스트와 순서화되지 않은 브로드캐스트로 나누어지는데, 전자의 경우 악성 응용프로그램에 의한 정보 도용 공격의 가능성이 크다[9]. 만약 악성 응용프로그램이 높은 우선순위를 설정하여 정상적인 응용프로그램보다 먼저 인텐트를 수신하고 이에 대한 내용을 확인 후 해당 메시지에 대한 추가 전달을 막거나 변조해서 전달한다면 큰 문제가 된다.

III. 응용프로그램 조사를 통한 보안 취약성 도출 및 플랫폼 개선안 설계

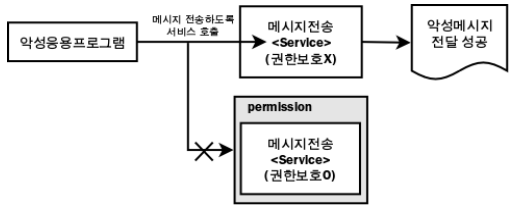
본 장에서는 권한보호 정책과 브로드캐스트 인텐트 정책에서 발생할 수 있는 보안 취약점을 시나리오 기반으로 설명하고 안드로이드 2.3 진저브레드 버전과 안드로이드 4.0 아이스크림 샌드위치 버전 기반의 실제 안드로이드 기기에서 확인한 결과를 제시한다. 또한 국내외 마켓의 일반 응용프로그램과 [10]에 제시된 악성 응용프로그램 집합을 이용하여 보안 취약 시나리오가 발생할 수 있는 가능성을 확인하고 이를 방지하기 위한 플랫폼 차원에서의 해결책을 제안한다.

3.1 권한 보호 정책과 브로드캐스트 인텐트의 잠재적인 보안 취약성 분석

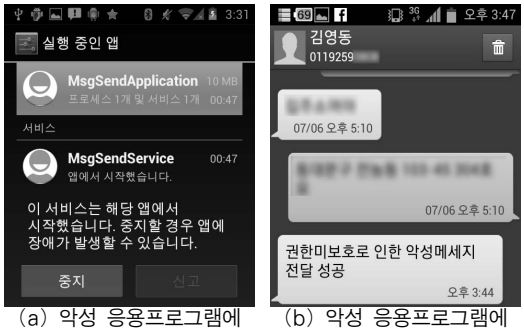
안드로이드 응용프로그램은 시스템에서 제공하는 여러 기능을 사용하기 위해 권한들을 요구한다. 하지만 개발자 입장과 사용자 입장에서는 권한이 할 수 있는 일의 범위를 파악하기에는 모호한 점이 존재한다. 예를 들어 READ_PHONE_STATE 라는 권한에 대하여 구글 개발자 공식 참조 사이트에서는 "Allows read only access to phone state." 라는 정도의 간단한 설명만이 제시되어 있다. 이는 휴대폰 번호, IMEI 번호 등의 민감한 개인 정보를 취득할 수 있는 보안상 문제가 있음에도 불구하고 권한 이름과 권한에 대한 설명이 너무 간결하기 때문에 개발자나 사용자 입장에서는 더욱 혼란을 줄 수 있다. 특히 많은 수의 권한을 요구하는 응용프로그램의 경우 사용자가 응용프로그램이 제공하는 기능과 권한과의 관계를 정확히 파악하기 어렵다. 실제로 [11]에 의하면, 84%에 해당하는 사용자들이 권한에 대해 모르거나 관심이 없는 것으로 나타났다.

한편 컴포넌트 보호를 위해 권한을 사용하는 경우에는 시스템에서 기본적으로 제공하는 권한을 이용하거나 개발자 자신이 정의한 권한을 이용할 수 있다. 만약 권한에 의한 보호가 이루어지지 않을 경우 해당 컴포넌트는 다른 악성 응용프로그램에 의해 호출될 수 있다. [그림 1]은 메시지를 전송하는 서비스를 권한을 이용해 보호하지 않은 시나리오를 고려한 것으로 컴포넌트에 적절한 권한 보호를 하지 않을 경우 [그림 2]와 같이 악성 응용프로그램이 문자 메시지를 전송하는 서비스를 호출할 수 있다.

브로드캐스트 인텐트의 경우 별도의 보안 메커니즘이 존재하지 않아서, 브로드캐스트 인텐트를 이용해 전달된 정보는 쉽게 수집이 가능하고 내용을 확인할 수 있는 약점이 있으며 순서화된 브로드캐스트의 경우 [그림 3]에 제시한 시나리오와 같이 메시지를 가로채 변조하거나 외부로 유출할 수 있는 가능성이 있다. 만약 악성 응용프로그램이 SMS 수신 관련 권한을 요구하며 인텐트 필터에 SMS 수신 브로드캐스트를 처리할 우선순위를 높게 설정한 상태로 리시버 컴포넌트의 구현부에서 abortBroadcast 메쏘드를 호출하게 한



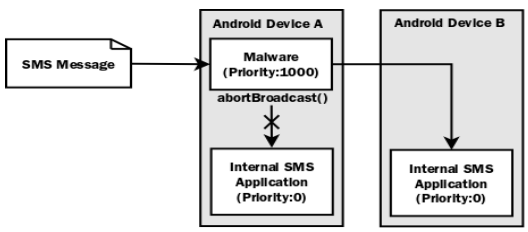
〔그림 1〕 권한 미보호로 인한 악성행위 시나리오



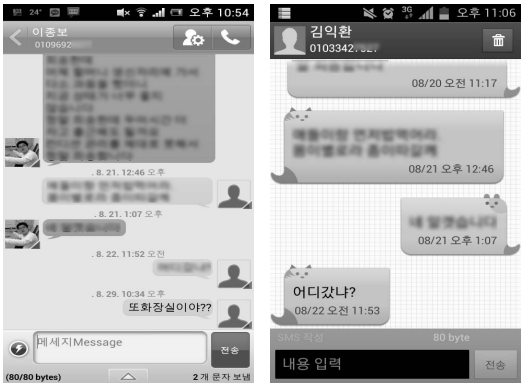
(a) 악성 응용프로그램에 의해 실행된 메시지 전송 서비스 (b) 악성 응용프로그램에 의해 전송된 메시지

〔그림 2〕 권한 미보호 컴포넌트를 사용한 악성 메시지 전달 결과

다면 기본 SMS 응용프로그램에는 SMS가 전달되지 않는다. 만약 악성 응용프로그램이 가로챈 SMS의 내용을 변조하여 전달하게 된다면 더욱 큰 문제가 된다. 구글 개발자 참조문서에서는 숫자로 표현되는 브로드캐스트 리시버의 우선순위를 1,000 이하의 값으로 사용하도록 권고하고 있지만, 1,000 이상의 우선순위 값을 사용하도록 설정할 때도 시스템에서 아무런 제약 없이 사용가능함을 확인할 수 있었다. 권한 관련 보안 취약점과 마찬가지로 응용프로그램 설치 시 사용자는 이러한 사실을 알 수 없어 사용자에게 많은 책임을 지우는 현재의 보안 정책상 잠재적인 보안 취약점이 될 수 있다. 〔그림 4〕는 이와 같은 보안 침해 시나리오를



〔그림 3〕 순서화된 브로드캐스트를 이용한 문자메시지 변조 및 가로채기 시나리오



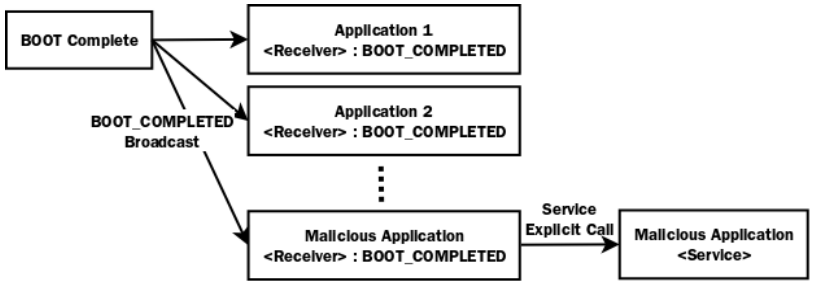
(a) 메시지 발신 장치 (b) 메시지 수신이 거부된 장치 (Android Device A)

(c) 메시지 가로챈 장치 (Android Device B)

〔그림 4〕 문자메시지 서비스 거부 및 가로채기 시나리오 결과

실제 안드로이드 기기 상에서 테스트한 결과이다.

정보 유출의 가능성 이외에도 악성 소프트웨어의 활성화 가능성 역시 존재하기 때문에 개발자는 브로드캐스트 인텐트의 송신에 있어서도 주의하여야 한다. 특히 BOOT_COMPLETED 브로드캐스트를 수신하는 응용프로그램의 컴포넌트는 안드로이드 시스템의 부팅과 동시에 활성화될 수 있다. 〔그림 5〕는 BOOT_COMPLETED를 수신한 응용프로그램의 리시버 컴포넌트가 악의적 서비스를 호출하는 시나리오를 나타낸 것으로 실제 기기에서 확인해본 결과 〔그림 6〕과 같이 부팅과 동시에 악의적 서비스가 백그라운드에서 활성화되는 것을 확인할 수 있었다. 이는 사용자 공지 없이 백그라운드에서 수행되는 서비스 컴포넌트의 특징으로 인하여 더 문제가 될 수 있다. 예를 들어, 안드로이드 4.0 아이스크림 샌드위치 버전에서



〔그림 5〕 BOOT_COMPLETED 브로드캐스트를 이용한 서비스 활성화 시나리오

〔그림 6〕 브로드캐스트의 수신을 이용해서 활성화 된 서비스

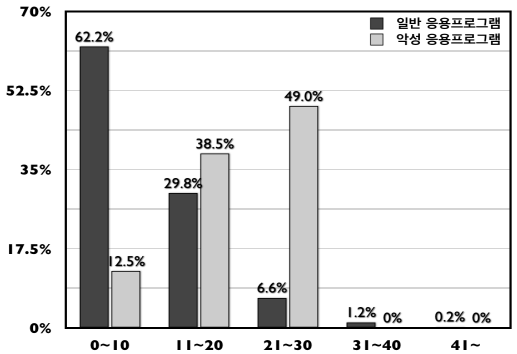
는 RECEIVE_BOOT_COMPLETED 권한을 획득하여야만 BOOT_COMPLETED 브로드캐스트 인텐트의 수신이 가능하도록 플랫폼이 수정되었으나, 현재까지 출시된 안드로이드 기기에 가장 많이 탑재된 안드로이드 2.3 진저브레드 버전에서는 시스템 버그로 인하여 권한 없이 BOOT_COMPLETED 브로드캐스트의 수신이 가능한 문제점이 남아 있다.

3.2 응용 프로그램 통계 분석 및 플랫폼 기반의 보안 모델 개선안 설계

권한은 앞서 언급한 바와 같이 응용프로그램에 있어서 개인정보 등에 접근하는 보안에 민감한 기능을 제공하지만 개발자가 얻을 수 있는 정보가 제한적이기 때문에 무분별하게 권한을 요구할 수 있고, 사용자는 권한에 대해 잘 모르거나 관심이 부족하기 때문에 보안취약성이 존재한다. 따라서 응용프로그램에서 권한을 이용한 컴포넌트 보호를 하지 않을 경우, 악성 응용프로그램에 의해 악의적 용도로 호출될 수 있는 보안 취약성이 존재한다. 본 연구에서는 권한 보호 모델과 관련해서 실제 권한 요구 실태와 권한 보호 수준(permission protection level)의 정의 여부를 확인하기 위해 구글 플레이 마켓에 등재된 응용프로그램을 대상으로 조사를 해 보았다.

실제 사용자들이 많이 사용하는 응용프로그램에서의 권한 요구 실태와 악성 프로그램의 권한 요구 실태에 관한 조사는 구글 플레이 마켓의 일반 응용프로그램 1,000개와 Contagio에서 제공하는 악성 응용프로그램 96개〔10〕를 대상으로 이루어졌다. 〔그림 7〕에서 확인할 수 있듯이 조사 대상 일반 응용프로그램 중 38% 가량이 10개 이상의 권한을 요구하는 것으로 나타났으며 심지어 시스템이 기본적으로 제공하는 130개 권한 중 30% 이상의 권한을 요구하는 응용프로그램도 존재하였다. 악성 응용프로그램의 경우에는 10개~30개 사이의 권한을 요구하는 경우가 전체의 87.5% 가량이며 20개 이상의 권한을 요구하는 경우도 50%나 되는 것으로 나타났다.

권한 보호 수준은 각각의 권한과 연관된 일종의 잠재적 위험 수준을 표현하는 개념으로 개발자가 정의한 권한과 마찬가지로 개발자가 응용프로그램의 매니페스트 파일에 정의하도록 되어 있다〔12〕. 권한 보호 수



〔그림 7〕 일반 응용프로그램과 악성 응용프로그램의 권한 요구 수 통계

〔표 1〕 권한 보호 수준의 정의

수준	설명
normal	시스템 또는 타 응용프로그램에 위험성이 적은 권한들에 부여. 설치 과정에서 사용자 공지 없음.
dangerous	normal에 비해 상대적으로 위험성이 높은 권한들에 부여. 설치 과정에서 사용자에게 허용 여부를 알리고, 확인 후 설치.
signature	권한을 요구하는 응용프로그램이 해당 권한을 정의한 기존 응용프로그램과 동일한 개발자 서명을 가지는지 검사 후 설치. 사용자에게는 특별한 공지 없음.
signatureOrSystem	시스템 이미지에 포함되었거나 시스템 이미지에 포함된 기존 프로그램과 동일한 개발자 서명을 가지는 응용프로그램에 대해서만 권한 허용. 주로 제조사에서 기본적으로 제공하는 응용프로그램들에 사용되며, 사용자에게 특별한 공지 없음.

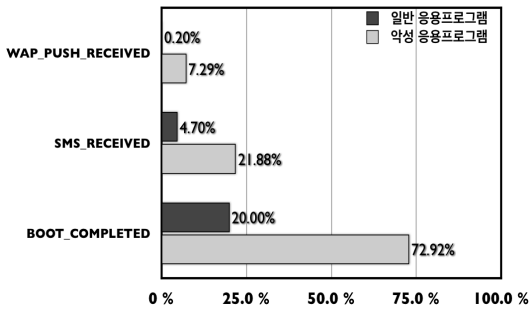
준을 정의함으로써 매니페스트 파일에 정의한 〈permission〉 태그의 정의와 더불어 시스템 차원에서의 권한 점검 과정을 더하여 선택적으로 보안성을 강화할 수 있는 특징이 있다. 안드로이드 플랫폼에 정의된 권한 보호 수준은 〔표 1〕과 같이 normal, dangerous, signature, signatureOrSystem의 4단계로 구분된다. 권한 보호 수준 중 normal과 dangerous 수준은 상대적으로 낮은 수준의 잠재적 위험성을 나타낸 것으로 사용자에게 공지 없이 허용하거나 사용자의 확인을 거쳐 허용하는 것이다. 개발자가 정의한 특정 권한과 연관된 보호 수준은 권한 정의와 마찬가지로 개발자가 응용프로그램의 매니페스트 파일에 정의하므로, 매니페스트 파일을 파싱함으로써 비교적 쉽게 확인할 수 있다. 권한 보호 수준의 정의 여부 조사는 구글 플레이 미국 마켓과 한국 마켓에 등재된 응용프로그램 중 인기무료 상위 200위에 해당하는 프로그램들을 대상으로 이루어졌다.

〔표 2〕 구글 한국 마켓과 미국 마켓 응용프로그램의 개발자 정의 권한 보호등급 통계

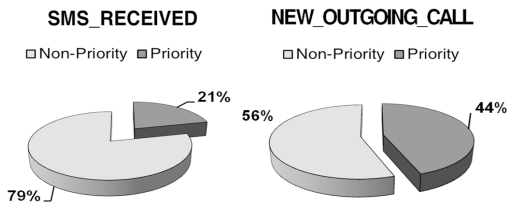
보호 수준 / 마켓 종류	normal	dangerous	signature	signatureOrSystem	No Protection
한국 마켓	4.6%	1.5%	77.7%	2.3%	13.8%
미국 마켓	1.7%	12.2%	76.5%	5.2%	4.3%

〔표 2〕는 개발자 정의 권한과 연관해서 개발자가 정의한 권한 보호 수준의 비율을 나타낸 것으로 실제 응용프로그램 중에는 보호 수준을 normal 또는 dangerous를 설정한 응용프로그램이 존재하였다. 개발자 정의 권한의 보호 수준을 normal 또는 dangerous로 설정한 경우에는 악성 응용프로그램 개발자가 매니페스트 파일 파싱 툴을 이용하여 권한의 이름을 파악하고 권한을 요구하도록 응용프로그램을 작성해서 배포하게 된다면, 권한에 무관심한 사용자에 의해 권한이 부여되어 3.1절에서 언급한 시나리오와 같이 문제가 발생할 수 있다. 또한 한국 마켓에 있는 응용프로그램에서는 보호 수준을 설정하지 않은 권한도 13.8%가 존재하여, 개발자가 사용자 정의 권한을 설정함에 있어서 보호 수준에 대한 관심이 부족함을 알 수 있었다. 따라서 개발자들은 보안과 밀접하게 관련된 컴포넌트에 대해 권한에 의한 보호를 설정해야 하며 signature 수준을 부여해야 한다. 일부 응용프로그램에서는 개발자 정의 권한만 선언한 채 내부 컴포넌트에 대해 권한 보호를 설정하지 않은 경우도 존재했으며 이는 개발자의 부주의로 판단된다. 현재 안드로이드 응용프로그램을 작성하는 통합 개발환경에서는 위와 같이 권한만 선언하고 사용하지 않는 경우에도 경고 없이 응용프로그램이 빌드된다. 이는 잠재적인 보안 취약점이 될 수 있기 때문에 개발환경 자체에서도 이러한 경우를 고려해서 경고 메시지를 출력하는 방법 등을 통해 개발자의 실수 혹은 부주의로 발생할 수 있는 보안 문제를 줄일 수 있도록 하는 방법이 필요하다.

이와 더불어 브로드캐스트의 송신으로 인한 악성 응용프로그램의 활성화 가능성을 조사하기 위하여 마



〔그림 8〕 일반 응용프로그램과 악성 응용프로그램의 브로드캐스트 리시버 등록 통계



〔그림 9〕 일반 응용프로그램을 대상으로 한 SMS_RECEIVED와 NEW_OUTGOING_ CALL의 우선순위 설정 통계

켓의 일반 응용프로그램과 악성 응용프로그램을 대상으로 브로드캐스트 리시버가 등록된 현황도 조사해 보았다. 〔그림 8〕과 같이 악성 응용프로그램의 경우 마켓에 등재된 일반 응용프로그램들과 비교할 때 상당히 많은 수의 브로드캐스트 리시버를 등록한 것을 확인했다. 특히 BOOT_COMPLETED 인텐트 메시지의 경우 조사 대상인 전체 악성 응용프로그램 중 73%에 해당하는 응용프로그램들이 해당 리시버를 등록하고 있는 것으로 나타났다. 이 결과를 통하여 앞서 언급된 BOOT_COMPLETED를 통한 악성 응용프로그램의 활성화 가능성이 상당히 높은 것으로 판단된다. 또한 우선순위가 부여된 순서화된 브로드캐스트를 사용한다면 민감한 개인 정보의 외부 유출, 변조, 차단 등의 문제가 발생할 수 있다. 이에 관련해서 순서화된 브로드캐스트 중 개인정보와 밀접하게 관련된 SMS_RECEIVED, NEW_ OUTGOING_CALL 브로드캐스트 리시버의 우선순위 설정여부를 확인해 본 결과, 〔그림 9〕와 같이 각각 21%, 44%에 해당하는 응용프로그램만이 우선순위를 설정해 놓은 것으로 나타나 보안측면에서 실제로 큰 문제가 될 수 있음을 확인할 수 있다.

앞에서 실제 마켓에 등재된 응용프로그램들의 권한과 브로드캐스트 인텐트 사용 실태를 살펴 본 결과, 다양한 형태의 보안 침해 시나리오가 존재함을 알 수 있었다. 한편, 이와 관련한 사용자의 인식은 부족하고 보안상 안전한 응용프로그램을 개발하여야 하는 개발자 역시 정보가 부족하거나 인식 수준이 높지 않아 실제 사용 실태에서 보안을 고려한 안전한 코딩을 적용하는 점에서 아직 부족함을 알 수 있다. 특히 응용프로그램 설치 시에 사용자에게 공지되는 권한에 대한 정보는 권한 분류와 권한명에 국한되므로 사용자가 허용한 권한이 실제 응용프로그램 수행시 어떠한 기능과 연관이 있다는 것을 비교, 판단하는 것은 현실적으로 상당히 힘들다. 더구나 응용프로그램이 우선순위가 설정된 브로드캐스트를 이용하는 경우는 대부분의 사용자가 이러한 기능이 백그라운드에서 사용되고 있다는 것조차 알기 힘든 상황이다.

정리하자면 사용자 측면에서 과도한 혹은 부적절한 권한 요청과 무분별한 브로드캐스트 인텐트 사용에 따른 보안 문제에 대응하는 데는 한계가 있다. 또한, 개발자 측면에서도 주의가 필요하지만 안전한 코딩을 위한 가이드라인과 정보 확산이 안드로이드 마켓의 급격한 성장세를 아직까지 쫓아가지 못하는 측면이 있다. 따라서 플랫폼 차원에서 악성 응용프로그램이라 판단되는 데이터를 토대로 이러한 한계를 보완하는 방안이 더욱 필요하다.

3.1절에서 기술한 보안 침해 시나리오를 예방하기 위하여 본 연구에서 제안하는 방법은 악성 응용프로그램이 가지는 전형적 특성 정보를 이용하여 새로운 응용프로그램 설치 과정에서 해당 프로그램을 검사하여 악성 응용프로그램으로 의심될 경우 이를 사용자에게 적절한 공지를 하는 방식이다. 특성 정보로는 악성 응용프로그램에서 주로 사용되는 것으로 조사된 권한 요청 사항과 브로드캐스트 인텐트 등록 사항을 사용하였다. 추가적으로 우선순위를 명시한 브로드캐스트 리시버가 설치하려는 응용프로그램에 등록되어 있을 경우, 해당 브로드캐스트를 수신할 수 있는 다른 브로드캐스트 리시버를 확인하여 우선순위 전도에 따른 브로드캐스트 인텐트 메시지의 유출 가능성이 있을 경우 사용자에게 공지하도록 한다. 이와 같은 작업을 설치 과정에서 진행되도록 하는 이유는 악성 프로그램의 런타임 행위를 모니터링하는 작업과 비교할 때 실제 수행 중에 성능 저하를 유발하지 않으며 작동 중에 악성 응용프로그램을 검사하는 기법에 비하여 플랫폼의 수정이 비교적 용이하기 때문이다.

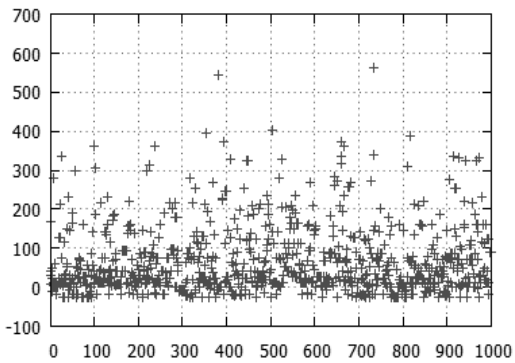
설치되는 응용프로그램이 악성 응용프로그램일 가능성이 높은지 여부를 탐지, 분류하는데 사용되는 기준을 마련하기 위하여 본 연구에서는 〔10〕에 명시된 96개 악성 응용프로그램들에서 주로 요구되는 권한 23개와 브로드캐스트 리시버 2개를 악성 응용프로그램의 특성 집합으로 잡았다. 이렇게 만들어진 특성 집합을 이용해 실제 안드로이드 마켓에 등재된 1,000개의 일반 응용프로그램과 96개 악성 응용프로그램의 매니페스트 파일 분석을 통해 특성 집합에 속한 권한과 브로드캐스트 리시버의 사용 현황을 조사하고, 그 결과를 〔표 3〕에 제시하였다. 〔표 3〕에서 항목 (A)와 (B)는 악성 응용프로그램 특성 집합에 속하는 각 권한과 브로드캐스트 리시버가 조사 대상인 전체 악성 응용프로그램과 일반 응용프로그램에서 얼마나 나타

〔표 3〕 주요 권한 및 브로드캐스트 리시버와 연관된 위험도 지수 산정

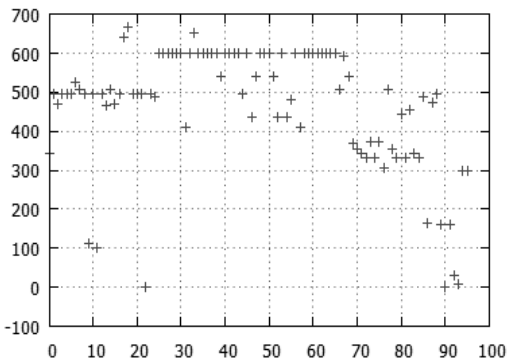
종 류		악성응용 현황(A) (%)	일반응용 현황(B) (%)	위험도지수 (A)-(B)
요구된 권한	INTERNET	96.88	96.90	-0.02
	READ_PHONE_STATE	95.83	66.20	29.63
	SEND_SMS	90.63	3.40	87.23
	WRITE_EXTERNAL_STORAGE	84.38	77.50	6.88
	READ_SMS	75.00	2.90	72.10
	READ_CONTACTS	72.92	15.40	57.52
	SET_WALLPAPER	68.75	4.00	64.75
	CALL_PHONE	66.67	19.20	47.47
	RECEIVE_SMS	64.58	8.20	56.38
	ACCESS_NETWORK_STATE	63.54	88.90	-25.36
	VIBRATE	55.21	42.80	12.41
	WRITE_SMS	52.08	0.90	51.18
	WRITE_APN_SETTINGS	46.88	0.30	46.58
	RECEIVE_MMS	45.83	0.40	45.43
	READ_HISTORY_BOOKMARKS	43.75	5.30	38.45
	WRITE_HISTORY_BOOKMARKS	73.75	5.20	38.55
	RECEIVE_WAP_PUSH	42.71	0.20	42.51
	GET_TASKS	40.63	23.20	17.43
	ACCESS_COARSE_LOCATION	33.33	23.60	9.73
	WRITE_CONTACTS	32.29	5.20	27.09
	ACCESS_FINE_LOCATION	31.25	32.40	-1.15
	MOUNT_UNMOUNT_FILESYSTEMS	31.25	2.30	28.95
	INSTALL_PACKAGES	16.67	3.40	13.27
등록된 브로드캐스트 리시버	BOOT_COMPLETED	72.92	20.00	52.92
	SMS_RECEIVED	21.88	4.70	17.18

났는지를 백분율로 나타낸 것이다. 이 조사 내용을 바탕으로 출현빈도의 차이인 (A-B)로 결정되는 상수값을 일종의 "위험도 지수"로 정의하고 이 위험도 지수가 높을수록 악성 응용프로그램일 확률이 높은 것으로 산정하였다. 그리고 각각의 응용프로그램에 대해서는 특정 권한과 브로드캐스트 리시버가 등록되었을 경우 이와 연관된 위험도 지수를 다 합하여 해당 프로그램의 위험도 지수를 산출하였다.

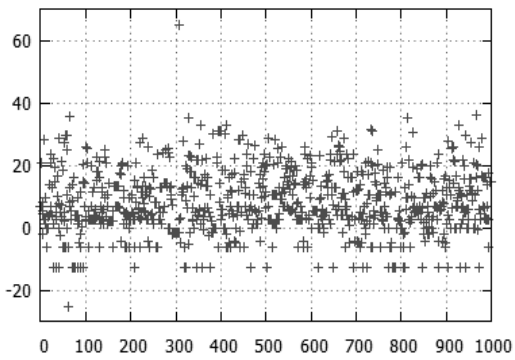
〔그림 10〕의 (a)와 (b)는 각각 일반 응용프로그램과 악성 응용프로그램에서의 위험도 지수 산출 결과를 나타낸 것으로 악성 응용프로그램에서 산출된 위험도 지수가 대체로 일반 응용프로그램에 비하여 높지만, 악성 응용프로그램 집합 내의 각 프로그램별 편차가 크게 나타나서 이 값을 그대로 악성 응용프로그램 여부 판단에 사용하기에는 무리가 있다. 이러한 결과가 발생한 원인은 비교적 적은 수의 권한을 요구하는 악성 응용프로그램의 경우 이들이 요구하는 권한 중 상당수가 실제 일반 응용프로그램에서도 다양한 기능 구현을 위해 많이 요구하는 것들로 각 권한에 대한 위험도 지수 자체가 낮고 이에 따른 합계도 역시 낮은 값을 가지기 때문으로 볼 수 있다. 이러한 한계점을 해결하기 위해 애초에 각 항목의 위험도 지수 합으로 산출하였던 응용프로그램의 위험도 지수를 위험도 지수 산정에 사용된 권한의 수로 나눈 평균으로 변경하여 다시 산출해 보았다. 〔그림 10〕의 (c)와 (d)는 그 결과를 나타낸 것으로 〔그림 10〕의 (a)와 (b)에 나타낸 결과에 비해 위험도 지수 측면으로 볼 때 일반 응용프로그램과 악성 응용프로그램의 특성 차이가 더 뚜렷해지는 것을 알 수 있다. 평균값을 이용한 악성 응용프로그램의 위험도 지수는 평균 32.1로 나타났으며, 이 값을 기준으로 하여 다시 앞에서 조사 대상으로 삼은 일반 응용프로그램과 악성 응용프로그램 집합에 대하여 분류 작업을 수행하였다. 그 결과 실제 악성 응용프로그램이 악성 프로그램일 확률이 높은 것으로 검출된 비율이 72.92%, 일반 응용프로그램이 검출된 결과는 0.8%로 나타났다. 실제 악성 응용프로그램이 위험도 지수 검사에서 탐지되지 않은 경우는 〔표 3〕에 제시된 항목 중 일반 응용프로그램과 악성 응용프로그



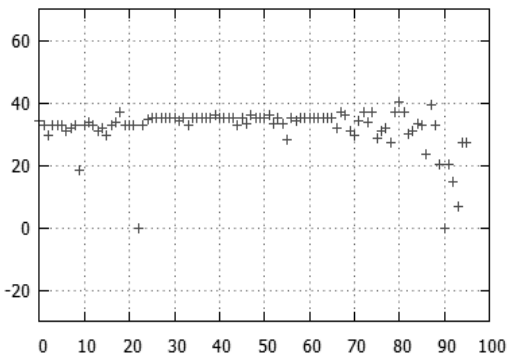
(a) 일반 응용프로그램의 위험도 지수 (합계 사용)



(b) 악성 응용프로그램의 위험도 지수 (합계 사용)



(c) 일반 응용프로그램의 위험도 지수 (평균값 사용)



(d) 악성 응용프로그램의 위험도 지수 (평균값 사용)

〔그림 10〕 일반 응용프로그램과 악성 응용프로그램의 위험도 지수 산출 결과

램에서 정의한 비율이 거의 유사하여 평균 위험도 지수 산정을 하더라도 낮은 점수 값을 가질 때이며, 단순하게 개별 권한 혹은 브로드캐스트 리시버 등록 사항의 위험도 지수로 산정하는 방식으로는 위험도 높은 권한들이 집합으로 결합하여 위험도가 증가하는 상황을 인지하지 못 하기 때문이다.

앞선 첫 번째 방법의 한계를 극복하기 위한 방법으로 본 연구에서는 악성 응용프로그램의 동작 특성을 구분하고 각각에 대한 분석을 통하여 실제 악성 응용프로그램과의 유사도를 통하여 검출하는 두 번째 방법을 제안한다. 이를 위해 〔10〕에 등록된 96개의 악성 응용프로그램의 동작 특성에 따른 종류 중 가장 많은 비율을 차지하는 Geinimi, Kmin, Pjapps&ADRD 세 가지 종류에 속하는 프로그램들에서 주로 요구하는 권한과 등록된 브로드캐스트 리시버를 조사한 결과를 〔표 4〕에 제시하였다. 〔표 4〕를 기반으로 검증을 시행할 때, 단 한 번이라도 악성 프로그램에 정의된 내용까지 다 포함할 경우에는 실제로는 문제가 없는 응용프로그램까지 악성 응용프로그램으로 분류되는 부작용이 있을 수 있으므로 "악성 코드 유사도"에 따른 사전 분석을 추가로 수행하고 그 결과를 〔표 5〕에 제시하였다. 검증 척도로 사용한 "악성 코드 유사도"는 〔표 4〕에 제시된 악성 프로그램의 특성 집합에 나타난 권한 요청사항과 브로드캐스트 리시버 등록사항 중 실제로 몇 개의 내용이 점검 대상 응용프로그램에 나타났는지를 백분율로 나타낸 것이다.

〔표 5〕의 결과를 보면 악성 코드 유사도를 사용한 두 번째 방법에서는 첫 번째 가중치 점수 평균을 통한 검출 방법의 결과인 정탐 비율 72.92%에 비하여 유사도 90% 기준 11% 이상 증가하여 상대적으로 악성 응용프로그램 검출 비율이 높은 것으로 나타났다. 또한 유사도 90%에서는 마켓의 일반 응용프로그램이 악성 프로그램으로 분류되는 비율이 1,000개 중 0.1%로 낮게 나타났다. 이 범주에 속하는 일반 응용프로그램은 명시적으로 악성 행위를 하지 않더라도 상대적으로 많은 권한을 요구하고 있다는 점에서 잠재적

〔표 4〕 악성 응용프로그램 동작 특성에 따른 요구된 권한 및 브로드캐스트 인텐트 요구

종 류		Geinimi	Kmin	Pjapps &ADRD
요구된 권한	INTERNET	○	○	○
	READ_PHONE_STATE	○	○	○
	SEND_SMS	○	○	○
	WRITE_EXTERNAL_STORAGE	○		○
	READ_SMS	○	○	
	READ_CONTACTS	○		
	SET_WALLPAPER	○	○	
	CALL_PHONE	○		
	RECEIVE_SMS		○	○
	ACCESS_NETWORK_STATE		○	
	VIBRATE		○	
	WRITE_SMS		○	
	WRITE_APN_SETTINGS		○	
	RECEIVE_MMS		○	
	READ_HISTORY_BOOKMARKS	○		○
	WRITE_HISTORY_BOOKMARKS	○		○
	RECEIVE_WAP_PUSH		○	
	GET_TASKS		○	
	ACCESS_COARSE_LOCATION	○		
	WRITE_CONTACTS	○		
	ACCESS_FINE_LOCATION	○		
	MOUNT_UNMOUNT_FILESYSTEMS	○		
	INSTALL_PACKAGES			○
등록된 브로드캐스트 리시버	BOOT_COMPLETED	○		
	SMS_RECEIVED			○

보안 취약성을 가지고 있다고 볼 수 있다. 따라서 이렇게 문제가 될 수 있는 일반 응용프로그램에 대해서도 사용자 공지가 이루어져야 한다.

본 연구에서는 보다 많은 악성 응용프로그램 검출을 위하여 두 번째 유사도를 통한 검출 방법을 앞선 첫 번째 평균 위험도 지수를 이용한 검출 방법과 함께 적용시켜 보았다. 실험은 1차로 평균 위험도 지수를 이용하여 응용 프로그램을 분류한 후, 1차 검사를 통과한 응용프로그램만을 대상으로 악성 코드 유사도를 이용한 2차 검사를 시행하는 방식으로 이루어졌으며 그 결과는 〔표 6〕과 같다. 통합 검사 결과 악성 응용프로그램 검출결과가 거의 90%의 검출 비율을 보였으며 이에 검출되지 않은 악성 응용프로그램을 분석한 결과 대체적으로 소수의 권한만으로 악의적 행위를 하는 형태이므로 제안한 방법으로 탐지가 불가능함을 알 수 있었다. 일반 응용프로그램의 경우 2차 검사에서 유사도 70%를 적용할 경우 전체의 3%가 위험성이 높은 응용프로그램으로 분류되었는데 그 이유는 이들 응용프로그램에서 요구하는 권한 사항과 브로드캐스트 리시버가 정당하게 사용될 경우에는 유용한 기능을 제공하지만 악의적으로 사용되거나 플랫폼의 결함으로 인해 오용될 수 있을 소지가 큰 사항들을 많이 요

〔표 5〕 악성 응용프로그램 집합과의 유사도에 따른 검출 결과

유사도	구 분	G	K	P	G∩K	G∩P	K∩P	G∩K∩P	Total (G∪K∪P)
70%	일반 응용	1.3%	0.4%	1.4%	0.2%	0.5%	0.2%	0.2%	2.4%
	악성 응용	28.1%	44.8%	43.8%	1.0%	28.1%	1.0%	1.0%	87.5%
80%	일반 응용	0.3%	0.3%	0.3%	0.1%	0.0%	0.0%	0.0%	0.8%
	악성 응용	27.1%	42.7%	18.8%	0.0%	3.1%	0.0%	0.0%	85.4%
90%	일반 응용	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%
	악성 응용	27.1%	41.7%	18.8%	0.0%	3.1%	0.0%	0.0%	84.4%

〔 G:Geinimi, K:Kmin, P:Pjapps&ADRD 〕

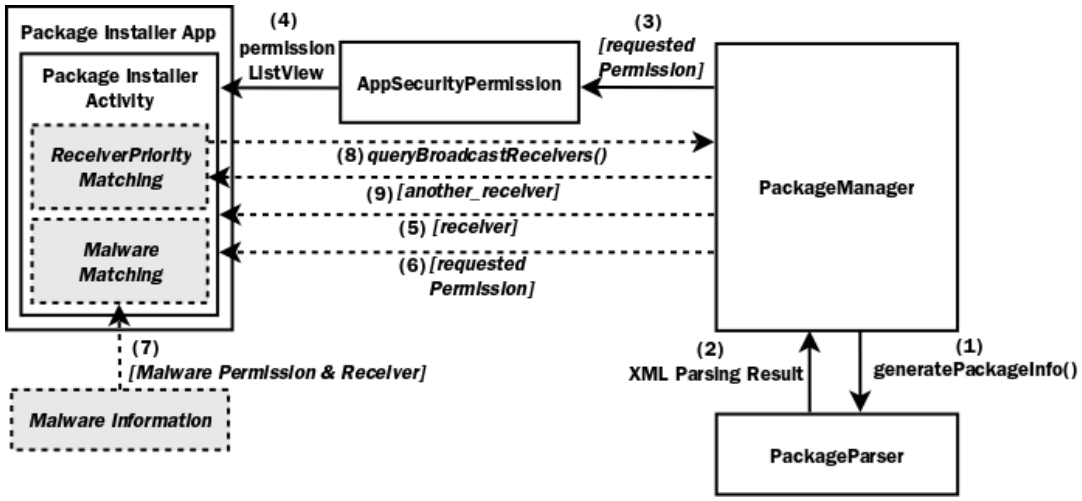
〔표 6〕 위험도 지수와 악성 코드 유사도를 함께 적용할 때의 검출 결과

	1차 위험도 지수를 이용한 검출	2차 악성 코드 유사도를 이용한 검출		최종 검출결과
일반 응용 프로그램	0.8 %	70% 유사도	2.20% 증가	3.00%
		80% 유사도	0.80% 증가	1.60%
		90% 유사도	0.10% 증가	0.90%
악성 응용 프로그램	72.92 %	70% 유사도	16.66% 증가	89.58%
		80% 유사도	15.62% 증가	88.54%
		90% 유사도	14.58% 증가	87.50%

구하고 있기 때문이다. 예를 들어 ACCESS_NETWORK_STATE 와 ACCESS_WIFI_STATE 권한의 경우 단순히 네트워크 설정 사항이나 WiFi 네트워크 접속을 위한 설정 사항을 접근할 수 있는 권한으로 되어 있어 일반 응용프로그램에서도 많이 사용되고 있지만 2011년에 실제 스마트폰 기기에서 ACCESS_WIFI_STATE 권한을 보유할 경우 사용자가 설정한 WiFi 접속 비밀번호까지 유출될 수 있는 결함이 발견된 경우가 있었다[13].

위에서 제안된 분석 결과를 바탕으로 플랫폼을 수정하기 위해서는 패키지 설치 프로그램의 수정을 필요로 하는데 이를 위한 설계안은 〔그림 11〕과 같다. 〔그림 11〕에서 (1)~(4)에 해당하는 과정은 기존의 안드로이드 플랫폼에서 응용프로그램이 설치되기에 앞서 사용자에게 권한 허가 여부를 묻는 과정이다. 설치 과정에서 PackageManager에 의해 AndroidManifest.xml 의 파싱이 이루어지고, 응용프로그램이 요청한 권한 자료에 관한 자료구조를 AppSecurityPermission에 전달하고 이에 대한 리스트 뷰를 PackageInstaller가 제공하는 형태이다.

본 연구에서는 3.2절에서 기술한 보안 침해 시나리오를 예방하기 위하여, 기존에 요청된 권한 리스트 뷰를 사용자에게 제공하여 사용자 설치 동의 여부를 묻는 PackageInstaller의 역할을 확장하였다. 첫 번째는 악성응용프로그램에 대한 정보를 바탕으로 설치가 요청된 응용프로그램의 권한과 브로드캐스트 리시버를 비교하여 사용자에게 공지를 해주는 과정이다. 이러한 수정을 위해서는 〔그림 11〕의 (5)~(6) 과정과 같이 요청된 권한과 브로드캐스트 리시버의 자료구조가 PackageInstaller에도 추가 전달되어야 하며 (7)에서와 같이 PackageInstaller 외부에 악성 응용프로그램의 정보를 자료구조 형태로 저장, 이러한 정보를 바탕으로 악성 응용프로그램 비교 작업을 수행하도록 수정되어야 한다. 악성 응용프로그램의 정보를 PackageInstaller 내부가 아닌 외부의 별도 자료구조로 보관하는 이유는 수시로 변하는 악성 응용프로그램 정보의 업데이트를 용이하게 하기 위함이다. 두 번째는 브로드캐스트 리시버가 우선순위가 있을 경우, 해당 브로드캐스트를 수신할 수 있는 다른 브로드캐스트 리시버를 확인하여 보다 낮은 우선순위의 브로드캐



〔그림 11〕 패키지 설치 프로그램 수정 설계안

스트 리시버일 경우, 요청된 권한과 함께 해당 브로드캐스트의 전달이 되지 않을 수 있다는 주의를 하는 형태이다. 이는 (5)에서 전달받은 브로드캐스트 리시버가 우선순위가 등록된 경우, (8)~(9)의 과정을 통하여 해당 브로드캐스트 리시버와 같은 브로드캐스트를 수신하는 다른 브로드캐스트 리시버의 리스트를 전달받도록 수정한 후, 이 리스트를 바탕으로 설치 요청된 응용프로그램과의 우선순위 비교를 통하여 사용자에게 공지하는 작업을 수행하는 부분의 수정이 PackageInstaller에서 이루어져야 한다.

이와 같은 해결방안은 안드로이드 플랫폼 중 응용계층에 해당하는 PackageInstaller를 수정함으로서 전체적인 안드로이드 플랫폼의 수정을 최소화한 형태로 이루어진다. 또한 런타임 행위를 일일이 모니터링하는 동적 방법에 비해 보다 적은 오버헤드로 응용프로그램의 악성여부를 사용자에게 공지할 수 있다.

IV. 결 론

본 논문에서는 안드로이드 플랫폼의 권한기반 보안 모델과 브로드캐스트 인텐트 매커니즘을 사용할 때 개인정보 유출과 관련된 잠재적 보안 침해 시나리오를 제시하고 이를 실제 안드로이드 스마트폰에서 확인하였다. 또한, 국내외 안드로이드 마켓의 응용프로그램과 악성 응용프로그램을 이용해 권한과 브로드캐스트 인텐트의 사용 현황을 조사하였으며, 이 결과를 바탕으로 응용프로그램 설치 과정에서 악성 응용프로그램과의 유사성 검사를 통해 악성 프로그램일 가능성이 높은 응용프로그램에 대해서는 사용자에게 자세한 정보를 공지하여 설치 여부를 신중하게 결정하는 데 도움이 되는 플랫폼 수정 설계안을 제시하였다. 제안한 기법의 유효성 검증을 위해 유사도에 따른 검사 정확도를 미리 확인하였으며, 그 결과 제안한 기법이 현재 알려진 주요 악성 응용프로그램 유형에 따라 유용한 일반 응용프로그램과 악성 프로그램을 효과적으로 분류할 수 있음을 확인하였다. 향후 연구로는 더 많은 악성 응용프로그램들을 대상으로 추출한 특성집합을 사용하여 제시된 설계안을 실제 구현하는 방안과 응용프로그램 정적 분석 기능을 통해 좀 더 견고한 판단기준을 수립하고 보안성을 향상시키고자 한다.

참고문헌

〔1〕 IDC, "Worldwide Quarterly Mobile Phone Tracker." Aug. 8. 2012.

〔2〕 Juniper Networks, "2011 Mobile Threats Report," Feb. 2012.

〔3〕 AhnLab, "ASEC 2011년 12월 Report," 2011년 12월.

〔4〕 한국인터넷진흥원, 안드로이드 기반 모바일 운영체제 보안기능 분석, KISA-WP-2010-0011, 2010년 8월.

〔5〕 김익환, 김태현, "안드로이드 플랫폼에서 유연한 응용프로그램 권한관리 기법 설계 및 구현," 정보처리학회 논문지C, 18-C(3), pp. 151-156, 2011년 6월.

〔6〕 Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," Proceeding of the 2012 IEEE Symposium on Security and Privacy, pp. 95-109, May. 2012.

〔7〕 B. Sarma, N. Li, C. Gates, R. Potharaju, and C. Nita-Rotaru, "Android permissions: a perspective combining risks and benefits," Proceeding of the 17th ACM symposium on Access Control Models and Technologies, pp. 13-22, Jun. 2012.

〔8〕 Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, You, Get off of My Market: Detecting Malicious Apps in Official and Alternative Android Markets," Proceedings of the 19th Annual Symposium on Network and Distributed System Security, Feb. 2012.

〔9〕 E. Chin, A. Felt, K. Greenwood, and D. Wagner, "Analyzing Inter-Application Communication in Android," Proceedings of the 9th international conference on Mobile systems, applications, and services, pp.239-252, Jun. 2011.

〔10〕 Contagio Malware Dump, "Take a sample, leave a sample. Mobile malware mini-dump - July 8 Update", Jul. 2011. 〔Online〕: http://contagiodump.blogspot.kr/2011/03/take-sample-leave-sample-mobile-malware.html

〔11〕 A. P. Felt, E. Ha, S. Egelman, A. Haney,

E. Chin, and D. Wagner, "Android Permissions: User Attention, Comprehension and Behavior," UCB/EECS-2012-26, University of California at Berkeley. 2012.
〔12〕 Google Android Developers Official Site: Dev Gude, "〈permission〉", 〔Online〕: http://developer.android.com/guide/topics/security/permissions.html
〔13〕 National Institute of Standards and Technology, "National Vulnerability Database, CVE-2011-4872", Feb. 2012. 〔Online〕: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2011-4872

〈著者紹介〉

김 영 동 (Young-dong Kim) 학생회원
2006년 3월~현재: 서울시립대학교 기계정보공학과 학사과정
〈관심분야〉 정보보호, 임베디드 시스템, 실시간 시스템

김 익 환 (Ikhwan Kim) 학생회원
2008년 2월: 서울시립대학교 기계정보공학과 학사
2011년 2월: 서울시립대학교 기계정보공학과 석사
2011년 3월~현재: 서울시립대학교 기계정보공학과 박사과정
〈관심분야〉 실시간 시스템, 임베디드 시스템, 모바일 시스템 보안

김 태 현 (Taehyoun Kim) 정회원
1994년 2월: 서울대학교 컴퓨터공학과 학사
1996년 2월: 서울대학교 컴퓨터공학과 석사
2001년 2월: 서울대학교 전기컴퓨터공학부 박사
2001년 5월~2005년 7월: ㈜지씨티 리써치 책임연구원
2005년 8월~현재: 서울시립대학교 기계정보공학과 부교수
〈관심분야〉 실시간 시스템, 임베디드 시스템, 모바일 시스템 보안