안드로이드 운영체제의 보안 취약점에 관한 연구

조희훈[1] · 김종배[2*]

[1,2*]숭실대학교 SW특성화대학원

A Study on the Security Vulnerability for Android Operating System

Hee-Hoon Cho[1] · Jong-Bae Kim[2*]

[1,2*]Graduate School of Software, Soongsil University

E-mail : [1]heehoonc@naver.com, [2*]kjb123@ssu.ac.kr

요 약

최근 안드로이드 기반의 모바일기기 보급이 증가하고 있다. 이에 따라 안드로이드 운영체제의 시장점유율도 늘어나고 있다. 하지만 안드로이드 운영체제의 경우에는 다른 폐쇄적 운영체제와는 달리 상대적으로 많은 보안 취약점을 지니고 있다. 대부분의 안드로이드 응용프로그램은 과도한 권한 승인이나 모바일 기기의 식별정보를 요구한다. 이러한 정보들은 보안 위협요인이 될 수 있다. 또한 응용프로그램의 설치과정에서 사용자의 동의를 얻기 때문에 보안문제가 발생할 경우 사용자가 책임을 지게 된다. 이러한 문제점이 지속될 경우, 사용자는 안드로이드 응용프로그램 사용 시 거부감을 느낄 수 있을 뿐만 아니라 운영체제의 신뢰성을 잃을 수 있다. 때문에 위와 같은 보안 취약점에 대해서는 조속한 대응책이 마련되어야 한다. 따라서 본 논문에서는 안드로이드 운영체제에서의 보안 취약점을 조사하고 대응방안을 제안하고자 한다.

ABSTRACT

Recently, Android-based mobile devices has increased. Thus increasing market share of the Android operating system. However, in the case of the Android operating system, it has the relatively large number of security vulnerabilities Unlike other closed operating systems. Most Android application requires the identity of the mobile device or over-authorization approval. This information can be a security threat. In addition, in the event of a security problem because obtaining the user's consent during the installation of the application is the user responsible. If these problems persist, loss of reliability of the user operating system, as well as to feel a resistance when using an Android application. In this paper, we investigate a security vulnerability in the Android operating system, and proposed countermeasures.

키워드

Security, Mobile Device, Mobile Operation System, Android

Ⅰ. 서 론

안드로이드 기반의 스마트폰 보급으로 인하여 안드로이드 운영체제의 확산이 날로 증가하고 있다. 안드로이드는 2007년 구글(Google)이 공개한 리눅스(Linux) 기반의 개방형 모바일 운영체제이며 안드로이드의 구조는 아래 그림과 같이 응용프로그램, 응용프로그램 프레임워크, 라이브러리, 안드로이드 런타임, 리눅스 커널로 구성되며 총 5개의 계층으로 분류되어 있다[1][2].

오픈 소스기반의 안드로이드의 경우 개방된 환경으로 개발자가 부담 없이 개발을 할 수 있고 접근이 용이하다. 하지만 접근이 쉽고 개발에 있어 제약조건이 적다는 점이 보안에 있어서는 취약점으로 작용하고 있다. 권한부여 측면에서의 경우 안드로이드 운영체제는 IOS와 같은 폐쇄적인 운영체제에 비해 과도한 권한을 요청하여 보안문제에 있어 문제점을 야기할 뿐만 아니라 사용자

를 식별하기 위해 개인 식별자 정보를 요구한다. 이러한 보안 취약점은 사용자에게 응용프로그램 사용 시에 불안감과 거부감을 들게 한다. 이러한 문제점이 지속될 경우 사용자는 운영체제에 대해 신뢰하지 못하고 다른 운영체제를 사용할 가능성이 높아지게 된다. 따라서 본 논문에서는 안드로이드 운영체제의 보안 취약점을 조사하고 이에 따른 대응책을 제안하고자 한다.

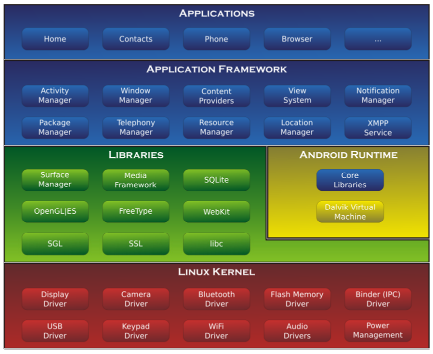

그림 1. 안드로이드 구조

II. 안드로이드 운영체제의 보안 취약점

1. 과도한 권한 요청 문제점

안드로이드 플랫폼은 응용프로그램 권한을 이용해 시스템의 자원이나 다른 응용프로그램의 컴포넌트 사용을 제어하는 보안 모델을 사용하고 있다. 그러나 권한 기반의 보안 모델은 사용자의 이해부족으로 인해 권한에 대한 자세한 지식이 없거나 악의적인 행위를 가진 개발자의 필요이상의 과도한 요청으로 인한 취약점이 존재한다[3]. 이러한 권한 요청 문제는 불필요한 권한을 확인하여 이를 승인하지 않을 경우 응용프로그램의 설치를 진행하지 못하는 경우가 발생한다. 때문에 사용자는 응용프로그램의 사용을 위해서는 권한 요청에 동의할 수밖에 없다.

2. 식별자 기반 인증방식의 문제점

대부분의 안드로이드 응용프로그램의 경우 사용자를 구분하기 위해 API(Application Programming Interface)기능에서 Device ID, MAC Address와 같은 식별자 정보를 얻는다. 이러한 식별자 정보를 사용하여 사용자를 구분짓고 인증방식을 거치게 된다. 하지만 식별자 정보의 경우 위·변조가 가능하기 때문에 보안 문제에 있어 취약점을 지닌다. 또한 인증을 거부할 경우 응용프로그램의 기능을 제한받게 되므로 사용자의 입장에서는 보안 위험에 노출 될 수밖에 없다.

3. 역분석 문제점

자바 가상 머신(JAVA Virtual Machine, JVM)에서 실행되는 안드로이드 응용프로그램은 바이트 코드(Byte code)로 이루어져 있다[4]. 이로 인해서 역컴파일(Decompile)이 쉽게 이루어 지는 특징을 가지고 있다. 이런한 특징은 제3자가 작성한 코드에 대해 역분석을 하여 정보를 얻는데 이용이 가능하다. 최근에는 이러한 점을 이용하여 다른 사람이 작성한 코드에 대한 정보를 쉽게 얻고 얻어낸 코드정보로 토대로 만들어진 악의적인 행위를 하는 악성 응용프로그램의 출현이 활발해지고 있다[5][6]. 이러한 악성 응용프로그램의 경우 개발자가 정상 응용프로그램과 유사하게 만들 경우 사용자의 입장에서 구분이 쉽지 않다.

III. 대응 방안

1. 과도한 권한 요청 대응 방안

권한 문제점의 경우 사용자의 필요에 따라 권한 부여의 분할이 가능하게 하는 방법을 제시한다. IOS의 경우 안드로이드 응용프로그램과는 달리 설치할 경우 모든 권한에 대해 사용자 동의를 얻는 것이 아니라 설치 후 필요한 기능에 대해서만 권한을 부여하고 나머지 기능에 대한 권한 요청을 승인하지 않더라도 응용프로그램을 사용할 수 있게 한다. 따라서 현재 안드로이드와 같이 사용자에게 권한 부여에 관한 공지만 이루어 질뿐 사용자의 부분적인 권한사용에 관한 선택의 여지를 주지 않는 시스템 방식은 사용자의 선택권을 제한하는 방법이라고 본다.

2. 식별자 기반 인증방식의 대응 방안

치명적인 식별자가 아닌 별도의 인증방식을 제안한다. 식별자 기반 인증방식의 응용프로그램 사용시 사용자는 식별자를 통한 인증 절차를 거치게 된다. 이러한 경우 식별자 정보 유출시에 사용자의 입장에서 보안문제를 안게 된다. 따라서 식별자를 인증 수단으로 사용하지 않는 것이 최선의 방법이며 UDID, E-mail인증과 같은 방법을 통하여 유출시 치명적인 식별자를 노출하지 않고 사용자를 구분하는 방법을 제안한다.

3. 역분석 대응 방안

JVM에서의 역분석을 통해 작성한 코드 정보를 얻을 수 있다. 이러한 문제점은 코드 난독화 기법을 통하여 역분석시에 이루어지는 코드 정보를 어렵게 바꾸는 것이 가능하다. 하지만 이러한 난독화 기법은 보통의 개발자들이 대체적으로 사용하지 않는다. 따라서 역분석을 통한 보안문제의 피해를 막기 위해서는 개발 시 난독화 기법을 의

무화 사용하는 것이 가장 좋은 대응방안이다.

Ⅳ. 결 론

안드로이드 운영체제의 시장이 증가하고 있지만 아직까지 다른 폐쇄적 운영체제에 비해 상대적으로 많은 보안취약점이 존재하고 있다.

본 논문에서는 안드로이드 운영체제의 문제점을 조사하고 이에 따른 대응 방안을 제시하였다. 안드로이드 응용프로그램의 권한 요청방식은 IOS와 같이 부분적인 권한을 요청하고 이를 통해 사용자는 불필요한 권한을 승인하지 않는 방식을 제안하였다. 식별자 기반의 인증 방식은 유출 시 치명적으로 작용하는 식별자를 사용하는 대신다른 방법(E-mail,UDID)을 통하여 인증하는 방식을 제안 하였다. 마지막으로 역분석의 문제점이 조사되었는데 이는 역분석된 코드를 복잡하게 만드는 난독화 기법을 의무화하여 코드의 보안성을 높이는 방법을 제안하였다. 향후 이러한 문제에 대한 정책이나 기준을 제안하고 이를 적용시킨 안드로이드 응용프로그램을 개발하여 안드로이드 응용프로그램 보안성을 높이는데 활용할 수 있을 것을 기대한다.

참고문헌

[1] 우종정, 부기동, 안드로이드 프로그래밍의 이해, 생능출판, 2012

[2] http://en.wikipedia.org/wiki/Android

[3] 김영동, 김익환, 김태현, 안드로이드 권한과 브로드캐스트 인텐트 매커니즘의 사용 현황 및 보안 취약성 분석,

[4] 이병용 최용수 “Obfuscation 기술의현황및 분석과향후개발방향” 보안공학연구논문지 5(3), pp.219-228 ,2008년6월

[5] Yuxue Piao, 정진혁, 이정현, 프로가드 난독화 도구 구조 및 기능 분석한국통신학회논문지 제38권 제8호(네트워크 및 서비스), 2013.8, 654-662 (9 pages)

[6] 이상호, 주다영, “안드로이드 기반 SNS 어플리케이션의 코드 변조를 통한취약점 분석 및 보안 기법 연구” 정보보호학회논문지 제23권 제2호, 2013.4, 213-221 (9 pages)