KIPS Tr. Comp. and Comm. Sys.
Vol.7, No.2 pp.59~66 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2018.7.2.59

Permission Management System for Secure IoT Devices in Android-Based IoT Environment

In Kyu Park† · Jin Kwak††

ABSTRACT

Android Things is an Android-based platform running in Google's IoT environment. Android smartphones require permissions from application users to use certain features, but in the case of Android Things, there is no display to send request notifications to users. Therefore Does not make a request to use the permissions and automatically accepts the permissions from the system. If the privilege is used indiscriminately, malicious behavior such as system failure or leakage of personal information can be performed by a function which is not related to the function originally. Therefore, By monitoring the privileges that a device uses in an Android-based IoT system, users can proactively respond to security threats that can arise through unauthorized use of the IoT system. This paper proposes a system that manages the rights currently being used by IoT devices in the Android Things based IoT environment, so that Android-based IoT devices can cope with irrelevant use of rights.

Keywords : Android-Things, IoT, Android Based IoT, Application Management, Permissions Management

안드로이드 기반 IoT 환경에서 안전한 IoT 디바이스를 위한 권한 관리 시스템

박 인 규† · 곽 진††

요 약

Android Things는 구글에서 발표한 IoT 환경에서 동작하는 안드로이드 기반 플랫폼이다. 이전 버전과는 다르게 자바 언어, 안드로이드 API, 구글 서비스 등 기존 서비스를 제공하며 더욱 쉽게 접근할 수 있도록 하였다. 안드로이드 스마트폰의 경우 특정 기능을 사용하기 위해 애플리케이션 사용자에게 권한을 요청하지만 Android Things의 경우 사용자에게 요청 알림을 보낼 수 있는 디스플레이가 존재하지 않은 경우도 있어 애플리케이션 개발시 특정 권한을 애플리케이션 내에 선언하지만 사용자에게 권한 사용에 대한 요청을 하지 않으며 시스템에서 권한을 자동 수락한다. 권한이 무분별하게 사용될 경우 본래 기능과 상관없는 기능으로 시스템 장애나 개인정보 유출 등의 악성행위를 수행할 수 있다. 따라서 안드로이드 기반 IoT 시스템에서 디바이스가 사용하는 권한을 사용자가 모니터링함으로써 IoT 시스템에서 무분별한 권한 사용을 통해 발생할 수 있는 보안위협에 대해 사전에 대응할 수 있다. 본 논문에서는 Android Things 기반 IoT 환경에서 IoT 디바이스가 현재 사용 중인 권한을 관리하는 시스템을 제안하여 안드로이드 기반 IoT 디바이스가 무분별한 권한 사용에 대해 대응할 수 있도록 한다.

키워드 : Android-Things, IoT, Android Based IoT, Application Management, Permissions Management

1. 서 론

Android Things는 안드로이드 기반 IoT 플랫폼으로 기존 안드로이드 플랫폼 기반으로 개발되었으며 2016년 12월 13일 개발자 프리뷰로 처음 공개되었다. Android Things의 이전 버전인 브릴로(Brillo) 버전의 경우 C/C++ 기반으로 개발되었지만 Android Things는 안드로이드 플랫폼으로 개발되어 기존에 안드로이드 탬플릿, 스마트폰 등에서 사용되던 구글 서비스, 안드로이드 API 등을 모두 사용할 수 있도록 지원함으로써 개발자들의 접근성을 높였다[1-2, 14].

Android Things의 경우 안드로이드 플랫폼을 이용하기 때문에 기존 안드로이드 디바이스에서 발생할 수 있는 보안 위협이 동일하게 발생할 가능성이 높다. 한 가지 예로 안드로이드 사용하는 권한을 Android Things도 동일하게 사용하기 때문에 애플리케이션이 본래의 기능과 무관한 권한을 요구하여 무분별하게 사용할 경우 데이터 유출 등의 피해를 입을 수 있다[3-6].

※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이며(No.NRF-2017R1E1A1A01075110), 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2015-0-00403).

† 준 회 원 : 아주대학교 컴퓨터공학과 석사과정
†† 종신회원 : 아주대학교 사이버보안학과 부교수

Manuscript Received : November 21, 2017
First Revision : December 15, 2017
Accepted : January 14, 2018

* Corresponding Author : Jin Kwak(security@ajou.ac.kr)

안드로이드 스마트폰은 애플리케이션이 설치될 때 사용자에게 애플리케이션이 사용하는 권한을 요청한다. IoT 디바이스는 디바이스의 상태를 표시하는 디스플레이가 없는 경우가 존재하기 때문에 Android Things는 특정 권한 사용 시 사용자에게 권한을 요청하지 않는다[5]. Android Things도 안드로이드 스마트폰과 마찬가지로 특정 기능을 사용하기 위해 필수적으로 AndroidManifest 파일에 권한을 선언하지만 IoT 디바이스에서 자동 승인하여 애플리케이션이 권한을 사용할 수 있도록 한다[4, 5].

애플리케이션이 요구한 권한을 자동 승인하여 애플리케이션을 설치할 경우 Fig. 1과 같이 무분별하게 권한이 사용될 수 있으며 이러한 경우 의도하지 않은 기능 수행으로 인해 개인정보 유출, 시스템 무력화 등의 심각한 피해를 초래할 수 있다[2].

또한, 안드로이드 플랫폼은 애플리케이션 제작, 배포 등을 쉽게 할 수 있는 구조로 되어 있다. 대표적인 예로 애플리케이션 리패키징은 기존에 알려진 애플리케이션의 소스코드를 일부 수정하여 패키징하는 기법을 말하며, 이후 오픈마켓, 블랙마켓 등을 통해 배포한다[4, 5].

본 논문에서는 Android Things 기반 IoT 환경에서 디바이스가 사용 중인 권한을 관리 및 모니터링하는 서버를 구축하고 서버에 연결되어 있는 모든 Android Things 디바이스를 분석하여 사용자에게 디바이스의 권한 정보를 제공한다. 사용자에게 현재 연결되어 있는 디바이스에 대한 정보를 제공함으로써 현재 디바이스가 어떠한 행위를 수행할 수 있는지 확인하여 추후 발생할 수 있는 악성행위에 대비할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서 제안하는 시스템과 관련하여 Android Things와 권한에 대해 살펴보고, 3장에서 안드로이드 플랫폼에서 발생할 수 있는 보안 위협을 분석한다. 4장에서는 제안하는 시스템을 설명하고 5장에서 구현된 시스템 결과를 확인하며 6장에서 제안한 시스템을 분석한다. 마지막으로 7장에서 결론을 맺는다.

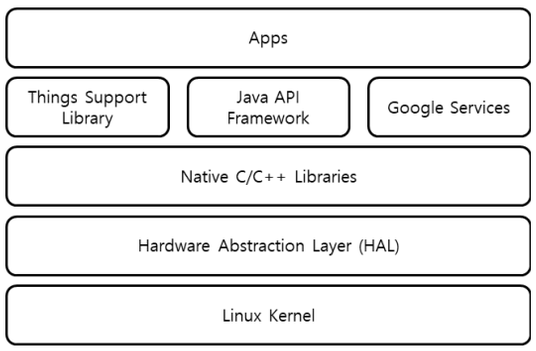

Fig. 2. Android Things Platform

2. 관련 연구

2.1 Android-Tthings

Android Things는 구글에서 발표한 안드로이드 기반 IoT 전용 운영체제이다. 구글에서 이전에 발표한 IoT 플랫폼과는 다르게 Android Things는 기존 안드로이드 서비스, API 등 다양한 기술을 사용할 수 있도록 하여 사용자의 접근성을 높였으며 Doorbell, Bluetooth 등 사용자가 쉽게 사용할 수 있는 샘플 코드를 제공함으로써 개발을 더욱 쉽게 할 수 있도록 지원하였다[1-3]. Android Things는 개발자 키트로 NXP Pico, Intel Edison, Raspberry Pi 3 등을 지원하며 Android Things 플랫폼은 Fig. 2와 같다[14].

2.2 Android Things 권한

Android Things는 기존 안드로이드 플랫폼과 동일한 형태를 갖고 있기 때문에 Android Things도 마찬가지로 애플리케이션 내부 AndroidManifest.xml에 특정 기능을 수행하기 위한 권한이 선언되어 있으며 권한을 선언하지 않을 경우 기능을

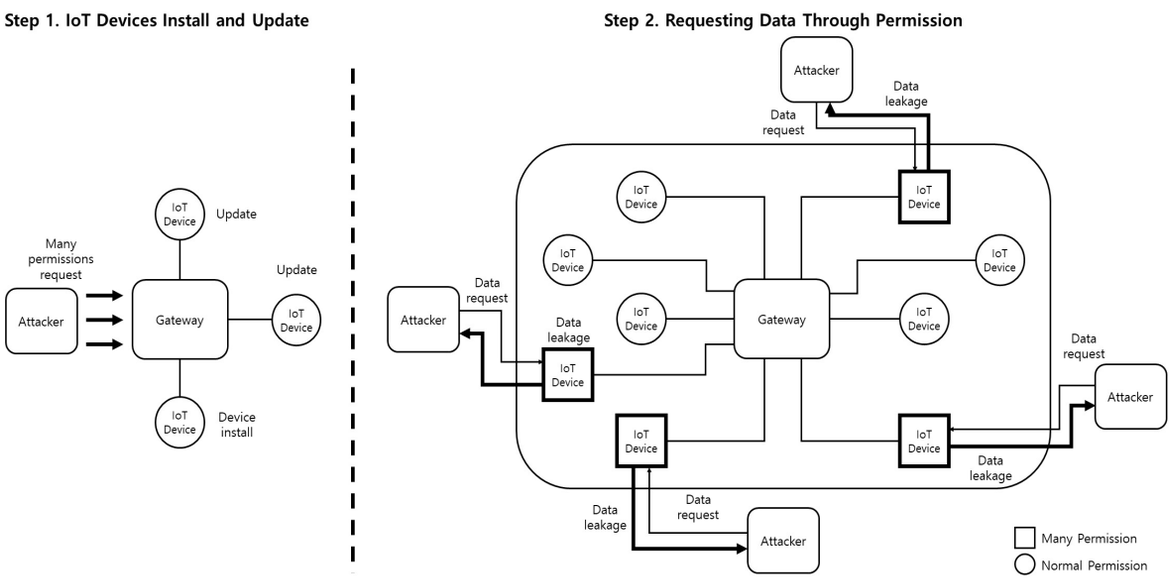

Fig. 1. Security Threat to Permissions Request

사용할 수 없다[5-8].

안드로이드 스마트폰의 경우 권한을 선언할 경우 사용자의 동의가 있어야 애플리케이션 설치가 가능하며 동의를 얻은 후에 특정 기능을 수행할 수 있다[4].

IoT 디바이스 특성상 디스플레이가 존재하지 않을 수 있기 때문에 Android Things 디바이스는 사용자에게 권한 요구와 같은 대화 상자를 요청하지 못한다. 사용자에게 권한 요청을 할 수 없기 때문에 개발자가 선언한 권한을 시스템에서 자동 승인한다. 자동 승인된 후 IoT 디바이스는 재부팅하고 재부팅 후에 선언된 권한이 적용된다[1, 4, 7].

Android Things 기반 애플리케이션에 있는 AndroidManifest.xml 파일의 경우 기존 안드로이드 스마트폰에서 사용되는 AndroidManifest.xml과 동일한 형태로 구성되어 있다. 안드로이드 스마트폰과 Android Things는 AndroidManifest.xml 파일에 Table 1과 같은 형태로 사용할 권한, 시작 액티비티 등의 정보를 갖는다[3, 6-9].

Table 1. AndroidManifest.file

```
<manifest package
  <uses-permission android:name="per_name"/>
  <uses-permission android:name="per_name"/>
  <application android:label="@string/app_name
    <action android:name="Activity"/>
    <category android:name="Main"/>
    <category android:name="LAUNCHER"/>
    <activity android:name=".StartActivity">
  </application>
</manifest>
```

다음 Fig. 3은 Android Things에서 제공하는 Doorbell 애플리케이션 샘플에서 사용하는 AndroidManifest.xml 파일이다. 사용 권한, 라이브러리, 액티비티 등 현재 안드로이드 스마트폰 애플리케이션에서 사용되는 것과 거의 유사하다[14].

```
<manifest package="com.example.androidthings.doorbell"
xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.INTERNET"/>
  - <application android:label="@string/app_name"
    android:icon="@android:drawable/sym_def_app_icon"
    android:allowBackup="true">
        <uses-library android:name="com.google.android.things"/>
      - <activity android:name=".DoorbellActivity">
          - <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category
                    android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
                <!-- Launch activity automatically on boot -->
          - <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category
                    android:name="android.intent.category.IOT_LAUNCHER"/>
                <category
                    android:name="android.intent.category.DEFAULT"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Fig. 3. Android Things AndroidManifest.xml File Example

3. 보안 위협

본 장에서는 Android Things 디바이스에서 권한 요구, 애플리케이션 유포로 인해 발생할 수 있는 보안 위협을 분석한다.

3.1 무분별한 권한 요청

안드로이드 스마트폰의 경우 애플리케이션 설치 시 개발자가 요구한 모든 권한에 대해 수락해야하며 권한 요청에 대해 거부하면 애플리케이션 설치는 취소된다. 이러한 이유로 사용자는 애플리케이션 설치를 위해 애플리케이션이 요구한 권한을 허용한다[3-5, 9-10].

애플리케이션이 과도한 권한을 요구할 경우 애플리케이션을 통해 본래 목적과 다른 기능을 통해 악성행위를 수행할 수 있다. 과도하게 권한이 사용될 경우 안드로이드 플랫폼이 권한을 통해 디바이스의 자원을 보호하려는 목적과는 다르게 권한이 사용될 수 있다[5-6, 10-11]. 예를 들어 블루투스의 기능을 수행하는 애플리케이션이 GPS, 문자 전송 등의 다른 기능을 수행할 수 있다.

Android Things 디바이스는 기존 안드로이드 플랫폼과 동일하게 권한을 사용하여 디바이스에서 과도한 권한을 요구할 경우 안드로이드 스마트폰과 동일한 보안 위협이 발생할 수 있다[1-2]. IoT 디바이스는 디바이스의 상태를 표시할 수 있는 디스플레이가 존재하지 않을 수 있다. 따라서 Android Things는 안드로이드 스마트폰과는 다르게 애플리케이션에서 요구하는 권한을 사용자에게 요청하지 않고 시스템에서 권한을 자동 허용한다[3, 14]. 이러한 경우 안드로이드 스마트폰에서 권한으로 인해 발생하는 보안 위협이 Android Things에서도 발생할 수 있으며 권한을 자동 허용하기 때문에 더 큰 위협이 될 수 있다.

Fig. 2는 무분별한 권한 요청으로 발생할 수 보안 위협 시나리오이며, 자세한 과정은 다음과 같다[2-3].

Step 1. IoT Devices Install and Update

Android Things가 새로 설치되거나 업데이트 될 때 공격자가 신규 애플리케이션에 대해 많은 권한을 요청하여 위치서비스, 블루투스, 네트워크 등 IoT 디바이스 기능을 많이 사용할 수 있도록 한다.

Step 2. Requesting Data Through Permission

많은 권한을 갖는 Android Things 디바이스는 본래의 목적과 다르게 다양한 기능을 수행한다. 예를 들어 GPS 서비스 기능을 제공하는 Android Things가 블루투스나 네트워크 권한을 가질 경우 GPS 서비스 외에 블루투스, 네트워크 기능까지 같이 수행하면서 접근할 수 있는 데이터의 양이 많아진다. 이러한 경우 IoT 디바이스에 있는 많은 양의 사용자 개인정보가 유출될 수 있다.

3.2 애플리케이션 유포

Android Things는 기존에 안드로이드 스마트폰과 같은 안드로이드 플랫폼 기반 디바이스와 동일한 플랫폼으로 사용되

기 때문에 기존 안드로이드에서 발생할 수 있는 보안 위협이 동일하게 발생할 수 있다[1, 3].

안드로이드 애플리케이션은 리패키징(Re-Packaging) 기법으로 인해 유포하기 용이하다. 리패키징은 기존 애플리케이션에 일부 소스코드를 수정하여 다시 패키징하는 기법으로 일반적으로 공격자는 유명 애플리케이션에 악성코드를 삽입하여 리패키징 후 배포한다. 리패키징 기법으로 인해 안드로이드 환경에서 악성 애플리케이션은 많이 유포되고 있으며 사용자 디바이스에 설치된 악성 애플리케이션은 사용자의 개인 정보를 유출시키거나 사용자 디바이스를 무력화시킨다[9, 11, 12].

Android Things도 마찬가지로 안드로이드 플랫폼 기반이기 때문에 안드로이드 스마트폰과 동일하게 리패키징으로 인한 보안 위협이 발생할 수 있다. 리패키징된 악성 애플리케이션이 Android Things 디바이스에 설치될 경우 디바이스는 본래의 기능과 관련 없는 권한이 선언되고 이를 통해 사용자의 개인정보 유출, 디바이스 오작동 등의 피해를 입을 수 있다[3, 6, 13].

4. 제안 시스템

본 장에서는 Android Things 플랫폼 기반 IoT 환경에서 디바이스가 특정 기능 수행을 위해 사용하는 모니터링하고 관리할 수 있는 시스템을 제안한다.

Android Things가 특정 행위를 수행할 경우 안드로이드 스마트폰과 마찬가지로 권한을 사용하는데 Android Things의 경우 애플리케이션이 패치될 때 안드로이드 스마트폰과는 다르게 사용자에게 권한 사용에 대한 동의를 얻지 않고 시스템에서 Android Things에 있는 권한을 자동 수락한다. 권한을 자동수락 할 경우 본래의 기능과 관련 없는 기능으로 데이터 유출 등의 악성행위를 수행할 수 있다. 따라서 현재 연결되어 있는 Android Thing 플랫폼 기반 IoT 디바이스가 사용하는 권한을 관리한다.

제안하는 시스템은 Android Things 플랫폼 기반 애플리케이션이 설치되거나 해당 디바이스에서 사용하는 애플리케이션이 수정될 때 현재 서버에 연결되어 있는 모든 디바이스가 사용하는 권한을 분석한다. 또한 분석된 권한을 각 디바이스 별로 관리하며 각 Android Things 디바이스 별로 사용중인 권한이 무엇인지 관리하여 사용자의 모바일로 확인할 수 있도록 한다.

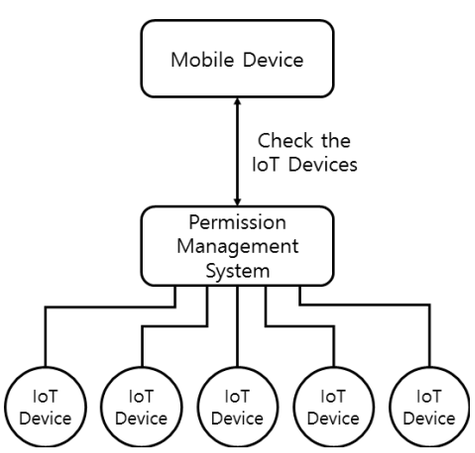

Fig. 4. Permission Management System

Android Things 디바이스에 대한 분석과 디바이스에 대한 정보 관리는 리눅스 서버를 통해 이루어지며 디바이스에 대한 정보 제공은 아파치 웹 서버를 통해 사용자에게 실시간으로 제공된다.

다음 Fig. 4는 제안하는 Android Things 플랫폼 기반 IoT 디바이스 권한 관리 및 모니터링 시스템이다. Android Things 디바이스는 권한 관리 시스템에 연결되어 있으며 권한 관리 시스템에 의해 Android Things 디바이스 분석, 관리가 수행되며 관리되는 데이터를 사용자에게 제공하는 기능도 수행한다.

4.1 권한 관리

IoT 디바이스가 사용하는 권한은 안드로이드 애플리케이션과 동일하게 IoT 디바이스 애플리케이션 내부에 있는 AndroidManifest.xml 파일을 통해 확인한다.

IoT 디바이스는 하드웨어 특성상 사용자에게 제공하는 기능이 크게 변하지 않기 때문에 애플리케이션이 업데이트 된다고 하더라도 요구하는 권한이 크게 변하지 않는다. 하지만 악성행위를 수행하기 위해 IoT 디바이스가 제공하는 권한과 관련 없는 권한을 요구할 수 있기 때문에 현재 버전 외에 이전 버전에 대한 권한 정보도 관리한다.

IoT 디바이스가 처음 설치될 때나 IoT 디바이스 업데이트 서버로부터 애플리케이션에 대한 변화가 있을 때 권한에 대한 정보를 수집한다.

권한 관리 시스템은 입력된 애플리케이션에 대한 정보를 바탕으로 AndroidManifest.xml 파일을 검색한다. 파일이 검색될 경우 해당 파일로부터 디바이스가 사용하는 권한 목록을 수집한다. 권한 목록 수집이 완료되면 디바이스 정보와 함께 권한 정보를 저장한다. 이후 권한 관리 시스템은 사용자가 모바일 디바이스로 확인할 수 있도록 해당 정보를 업데이트한다.

해당 IoT 디바이스가 사용하는 권한에 대한 정보를 수집하기 위해 AndroidManifest.xml 파일 검색 후 해당 파일에 대한 분석을 수행한다.

권한 관리를 위해 수행되는 단계는 Fig. 5와 같이 총 5단계이며 Android Things 디바이스 정보 입력, AndroidManifest.xml 파일 검색, 디바이스에서 사용중인 권한 리스트 수집, 권한 정보 저장, 웹 서버 업데이트 순으로 진행된다.

1) Android Things 애플리케이션 입력

IoT 환경에서 새로운 IoT 디바이스가 설치되거나 IoT에 대한 패치가 수행되면 해당 애플리케이션을 분석하기 위해 애플리케이션 정보를 수집한다. 추후 여기에서 입력된 정보를 통해 IoT 디바이스에 대한 분석이 수행된다.

2) AndroidManifest.xml 검색

일반적으로 Android Things 플랫폼 기반으로 만들어진 IoT의 경우 루트 폴더 하위에 app 폴더에 AndroidManifest.xml 파일 있다. 하지만 항상 동일한 경로에 존재하지 않을 수 있기 때문에 애플리케이션 내에 모든 경로에 대해 검색을 수행한다.

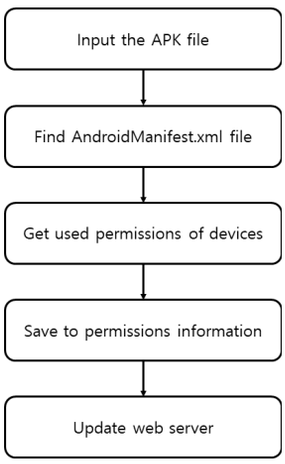

Fig. 5. Permissions Management Process

3) AndroidManifest.xml 분석

IoT 디바이스가 사용하는 권한에 대한 분석을 수행하기 위해 이전에 검색했던 AndroidManifest.xml 파일을 분석한다. Android Things도 기존 안드로이드 애플리케이션과 동일하게 <uses-permission android:name="권한명"/> 형태로 선언이 되어있다. 따라서, AndroidManifest.xml 파일 내부에 존재하는 사항 중 권한 선언 형태를 갖는 선언문이 있을 경우 해당 정보를 수집한다.

Android Things 플랫폼 기반 애플리케이션은 개발자를 위해 Doorbell, Bluetooth 등 샘플 애플리케이션을 제공하여 개발을 쉽게 할 수 있도록 제공한다. 샘플 코드에는 AndroidManifest.xml 파일이 CompanionApp 형태로 두 개가 있는 경우가 있기 때문에 권한 분석시 모든 경로에 있는 AndroidManifest.xml 파일에 대해 분석을 수행한다.

4) IoT 디바이스 및 권한 저장

IoT 디바이스가 사용하는 권한에 대한 분석이 완료될 경우 사용자에게 정보 제공을 하기 위한 데이터 저장을 수행한다. 제공되는 정보는 디바이스 이름, 수정된 날짜, 디바이스가 사용중인 권한 정보를 제공한다.

5) 웹 서버 업데이트

저장된 Android Things 플랫폼 기반 IoT 디바이스 정보를 모바일 디바이스를 통해 사용자에게 제공하기 위해 웹 서버에 업데이트 한다. IoT 디바이스에 대한 정보는 리눅스 서버의 웹 서버를 통해 제공하며, IoT 디바이스에 대한 정보는 사용자가 각 디바이스별로 확인할 수 있도록 제공한다.

4.2 IoT 디바이스 정보 확인

사용자는 현재 IoT에 접속되어 있는 IoT 환경에 연결되어 있는 IoT 디바이스에 대한 정보를 확인하기 위해 IoT 디바이스를 관리 및 모니터링하는 웹 서버에 접속한다. 웹 서버 접속 시 사전에 등록된 사용자만 접속이 가능하며 서버에 접속할 경우 IoT 환경에 존재하는 권한, 수정날짜 등 각 디바이스에 대한 정보를 확인할 수 있다.

5. 시스템 구현 결과

구현된 시스템은 Android Things 플랫폼 기반 IoT 디바이스가 연결된 서버에서 IoT 디바이스에 대한 권한 정보를 제공한다. IoT 디바이스를 관리하는 서버는 IoT 디바이스를 실시간으로 모니터링하며 IoT 디바이스가 서버에 연결되거나 애플리케이션의 패치로 인해 애플리케이션이 업데이트 됐을 경우 IoT 디바이스에 대한 분석을 수행한다.

5.1 권한 관리 서버

서버는 IoT 디바이스에 대한 변화가 생길 경우 변화가 생긴 IoT 디바이스의 정보를 수집한다. IoT 디바이스가 사용하는 권한 수집을 위해 안드로이드 플랫폼에서 권한 요청시 사용하는 AndroidManifest.xml 파일을 검색한다. Android Things 애플리케이션에서 AndroidManifest.xml 파일을 찾기 위해 다음 Table 2와 같이 IoT 애플리케이션 내부에 존재하는 AndroidManifest.xml 파일을 검색한 후 사용 권한 목록을 수집하는 형태의 수도코드로 프로그램을 작성하였다.

Table 2. Experimental Results

No.	Permission	Permission Information
1	BLUETOOTH	Bluetooth communication access
2	CAMERA	Camera devices access
3	INTERNET	Application network access
4	READ_EXTERNAL_STORAGE	External stroage access
5	RECORD_AUDIO	Audio receive and record access
8	…	…

파일에 대한 검색이 끝날 경우 IoT 디바이스 내에 존재하는 AndroidManifest.xml 파일을 모두 검색하여 이후 검색된 AndroidManifest.xml 파일에 대한 경로는 분석 수행을 위해 절대 경로 형태로 데이터가 수집된다. Fig. 6은 경로 검색이 끝난 후 수집된 데이터이다.

```
AndroidManifest >> /home/isaa/androidthings/sample-butt
on-master/app/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-nati
vepio-master/speaker/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-nati
vepio-master/button/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-nati
vepio-master/blink/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-simp
lepio-master/button/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-simp
lepio-master/blink/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-simp
lepio-master/pwm/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/bluetooth-a
udio/audio-sink/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sample-usbe
num-master/app/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/sensorhub-c
loud-iot-master/app/src/main/AndroidManifest.xml
AndroidManifest >> /home/isaa/androidthings/doorbell-ma
ster/app/src/main/AndroidManifest.xml
```

Fig 6. AndroidManifest search result

5.2 데이터 가공

IoT 디바이스가 사용하는 권한 분석이 끝나면 사용자에게 IoT 디바이스에 대한 정보를 제공하기 위해 데이터를 가공한다. 일반적으로 AndroidManifest.xml 파일 분석을 통해 안드로이드 플랫폼에서 사용하는 권한을 수집할 경우 “android.permission.permisson_name” 형태를 나타낸다. 이러한 경우 수집한 데이터를 그대로 사용자에게 제공할 경우 사용자는 수집된 권한 목록이 어떠한 행위를 수행하는 권한인지 알 수가 없다. 따라서 권한 수집이 완료될 경우 사용자에게 정보를 제공하기 위한 데이터 가공을 수행한다.

Android Things의 경우 안드로이드 플랫폼에서 제공하는 권한과 동일한 권한을 사용한다. 따라서 디바이스관리 및 모니터링 서버는 테이블 형태로 권한과 권한별 의미를 저장한 뒤 IoT 디바이스로부터 수집된 권한 목록과 매칭하여 권한별로 어떠한 기능을 하는지 식별한다. 권한별로 수행하는 기능을 분석하기 위해 다음 Table 3과 같은 수도코드를 통해 프로그램을 작성하였다.

Table 3. Permission Information Search Pseudo Code

```
Algorithm Search_Permission
get Permission list
while(Size(Permission list)):
  if Permission==table(Permission)
    Find permission information
  end if
end while
return result_permission
```

데이터 가공을 위해 서버에 저장되어 있는 권한 정보 테이블은 다음 Table 4와 같은 형태를 나타내며 일부 권한에 대한 정보이다.

Table 4. Experimental Results

No.	Permission	Permission Information
1	BLUETOOTH	Bluetooth communication access
2	CAMERA	Camera devices access
3	INTERNET	Application network access
4	READ_EXTERNAL_STORAGE	External stroage access
5	RECORD_AUDIO	Audio receive and record access
8	…	…

5.3 서버 업데이트

서버 업데이트는 사용자에게 현재 연결되어 있는 Android Things 디바이스에 대한 정보를 제공하기 위해 수행하는 단계이다. 수집된 데이터에 대한 가공이 완료되면 서버는 각 디바이스별로 사용자에게 서비스를 제공하기 위해 어떤 권한을 사용하는지 정리하여 저장한다. 사용자는 해당 서버에 접근할 경우 다음 Fig. 7과 같이 사전에 등록된 ID와 Password를 통해 권한 정보에 접근할 수 있다.저장된 데이터는 최상위 폴더인 result 폴더에 저장되며 result 폴더 하위에 각 IoT 디바이스 명으로 권한 정보가 담겨있는 파일이 생성하여 사용자가 실시간으로 쉽게 접근할 수 있도록 한다.

Fig. 7. User Authentication

result 폴더 하위에 IoT 디바이스 명으로 저장된 형태는 다음 Fig. 8과 같으며 각 파일 내부에 디바이스가 사용중인 권한 명과 권한에 수행하는 기능이 명시되어 있다

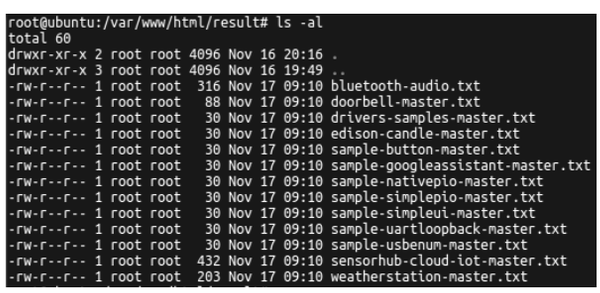

Fig. 8. Android Things List

5.4 사용자 서버 접근

사용자가 서버에 저장된 데이터에 접근하기 위해 서버의 result 폴더에 접근한다. result 폴더에 접근하면 현재 서버에 연결되어 있는 모든 Android Things 디바이스 목록을 확인할 수 있다.

현재 서버에 연결되어 있는 Android Things 디바이스는 Fig. 9와 같으며 각 디바이스별로 사용 중인 권한을 확인할 수 있다.

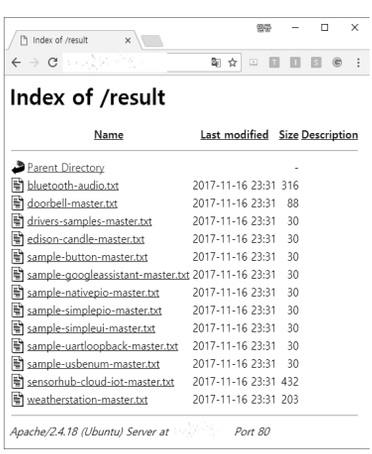

Fig. 9. List of Connected Devices

Android Things 디바이스 목록에서 디바이스가 사용중인 권한 목록을 확인할 경우 “android.permission.permission_name” 형태의 내용과 해당 권한이 갖는 의미를 나타낸다. 현재 서버에 연결된 Android Things 디바이스가 사용중인 권한이 없을 경우에는 사용중인 권한이 없다는 문구가 표시된다.

다음 Fig. 10은 사용자가 Android Things 디바이스에서 사용중인 권한을 확인했을 때 나타나는 데이터이다.

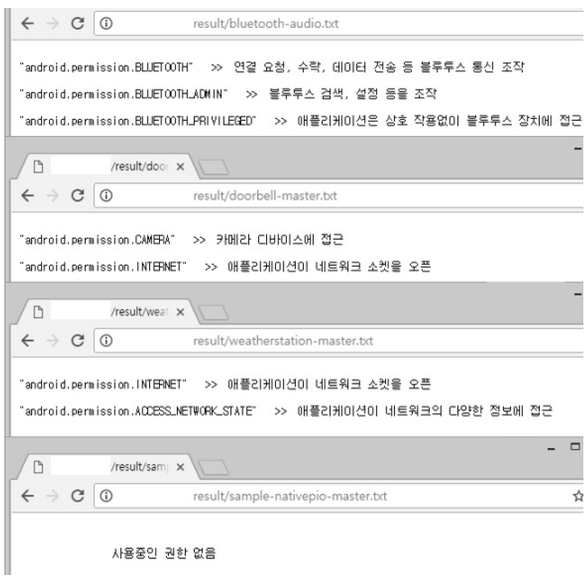

Fig. 10. Permission Result

6. 시스템 분석

본 장에서는 제안하는 시스템 도입 시 안드로이드 기반 IoT 디바이스가 무분별하게 권한을 사용할 경우에 대해 사용자가 대응할 수 있는 방안을 분석한다. IoT 시스템에 연결된 디바이스 목록을 통해 각각의 디바이스가 갖는 권한 목록을 확인하여 디바이스가 무분별하게 권한을 사용하는 것을 방지한다.

Table 5. Experimental results

Devices	Permission	Normal Permission	Indiscriminate permission
D1	P1	✓	
	P3	✓	
	P7		✓
D2	P2	✓	
	P3	✓	
	P4	✓	
D3	P2		✓
	P6	✓	
D4	P2		✓
	P5		✓
	P6	✓	
	P7	✓	
D5	…	…	…

서버에서 제공하는 IoT 디바이스 목록과 각각의 디바이스가 사용하는 권한목록을 확인하고 해당 권한을 통해 수행할 수 있는 행위를 파악한다. 권한을 통해 수행할 수 있는 행위를 파악할 경우 Table 5와 같이 IoT 디바이스가 불필요하게 사용하고 있는 권한을 확인할 수 있다. 또한 이를 바탕으로 IoT 디바이스가 개인정보 유출, 디바이스 제어 등의 악성행위 수행을 방지할 수 있다.

7. 결 론

안드로이드 기반 IoT 플랫폼인 Android Things는 이전 안드로이드 플랫폼과 동일하게 특정 기능 수행을 위해 권한을 선언하는 구조로 되어있다. 안드로이드 스마트폰은 애플리케이션에 권한을 선언할 경우 애플리케이션 설치 시 사용자에게 권한 사용에 대한 동의를 구한다. 하지만 Android Things는 IoT 디바이스 특성상 디스플레이가 존재하지 않는 경우도 있어 사용자에게 동의를 구하지 않고 시스템에서 애플리케이션이 기능 제공을 위해 요구하는 권한을 자동으로 승인한다.

Android Things는 안드로이드 플랫폼 기반이기 때문에 안드로이드 스마트폰에서 발생하는 보안 위협이 동일하게 IoT 환경에서도 발생할 수 있다. 안드로이드 플랫폼 기반의 디바이스에서 발생할 수 있는 대표적인 보안 위협은 무분별한 권한 사용으로 인한 데이터 유출이다. 안드로이드 환경에서는 1~2개의 권한으로도 많은 API 사용이 가능하여 무분별한 권한 사용은 큰 피해로 이어질 수 있다.

본 논문에서는 Android Things 플랫폼 기반 IoT 환경에서 서버에 연결되어 있는 디바이스를 관리할 수 있는 시스템을 제안하고 서버를 구축하였다. 구축된 시스템을 통해 서버에 연결되어 있는 Android-Thing 디바이스가 현재 사용중인 권한에 대해 분석하여 관리함으로써 사용자가 실시간으로 현재 IoT 디바이스의 사용권한에 대해 확인할 수 있도록 하였다.

향후 본 논문에서 제안 및 구현한 시스템을 통해 보다 안전한 Android Things 기반 IoT 환경을 제공하고 Android Things에 관련된 연구를 진행하는데 기여될 것으로 예상된다.

References

[1] Htaejoo Cho, Hyunki Kim, and Jeong Hyun Yi, “Security Assessment of Code Obfuscation Based on Dynamic Monitoring in Android Things,” *Special Section: Security and Privacy in Applications and Services for Future Internet of Things,* IEEE Acess, 2017.

[2] Mahdi Amiri-Kordestani and Hadi Bourdoucen, “A Survey on Embedded Open Source System Software for the Internet of Things,” *Free and Open Source Software Conference 2017 (FOSSC'17)*, 2017.

[3] W. J. Okello, Q. Liu, F. A. Siddiqui, and C. Zhang, “A survey of the current state of lightweight cryptography for the Internet of things,” In *Computer, Information and Telecommunication Systems (CITS), 2017 International Conference on.* IEEE, pp.292-296, 2017.

[4] H. Wang, Y. Guo, Z. Tang, G. Bai, and X. Chen, “Reevaluating Android Permission Gaps with Static and Dynamic Analysis,” *in Global Communications Conference (GLOBECOM), 2015 IEEE,* pp.1-6, 2015.

[5] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, “Android Permissions: A Perspective Combining Risks and Benefits,” *in SACMAT '12 Proceedings of the 17th ACM symposium on Access Control Models and Technologies,* pp.13-22. 2015.

[6] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, “Android permission: user attention, comprehension, and behavior,” *in SOUPS '12 Proceedings of the Eighth Symposium on Usable Privacy and Security,* Article No. 3, 2012.

[7] Jiaojiao Fu, Yangfan Zhou, Huan Liu, Yu Kang, Xin Wang, “Perman: Fine-grained Permission Management for Android Applications”, *in Software Reliability Engineering (ISSRE), 2017 IEEE 28th International Symposium on, 2017.*

[8] Ajay Kumar Jha, Seungmin Lee, Woo Jin Lee, “Permission-based Security in Android Application - From Policy Expert to End User”, *in RACS Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems*, 2015.

[9] K. Tam, A. Feizollah, N. B. Anuar, R. Salleh, and L. Cavallaro, “The Evolution of Android Malware and Android Analysis Techniques,” *in Computing Survey,* Vol.49, No.4, Issue 4, 2017.

[10] C. Da, Z. Hongmei, and Z. Xiangli, “Detection of Android malware security on system calls,” *in CAdvanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC),* pp.974-978, 2016.

[11] J. Wu, S. Liu, S. Ji, M. Yang, T. Luo, Y. Wu, and Y. Wang, “Exception Beyond Exception: Crashing Android System by Trapping in “uncaughtException”,” *in Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), 2017 IEEE/ACM 39th International Conference on,* pp.283-292, 2017.

[12] D. Wang, H. Yao, Y. Li, H. Jin, D. Zou, and R. H. Deng, “A Secure, Usable, and Transparent Middleware for Permission Managers on Android,” *in IEEE Transactions on Dependable and Secure Computing, 2017 IEEE 28th International Symposium on,* Vol.14, No.4, 2017.

[13] A. Jain and Prachi, “Android Security : Permission Based Attacks,” *in Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on,* pp. 2754-2759, 2016.

[14] Android Things [Internet], https://developer.android.com/things/.

박 인 규

http://orcid.org/0000-0002-8922-5198

e-mail : ikpark.isaa@gmail.com

2016년 순천향대학교 정보보호학과(학사)

2016년~현 재 아주대학교 컴퓨터공학과 석사과정

관심분야 : 모바일 보안, IoT 보안

곽 진

http://orcid.org/0000-0001-6931-2705

e-mail : security@ajou.ac.kr

2000년 성균관대학교 생물기전공학(공학사)

2003년 성균관대학교 컴퓨터공학(공학석사)

2006년 성균관대학교 컴퓨터공학(공학박사)

2006년 일본 큐슈대학교 방문연구원

2006년 일본 큐슈시스템정보기술연구소 특별연구원

2006년~2007년 정보통신부 정보보호기획단 개인정보보호팀 통신사무관

2007년~2015년 순천향대학교 정보보호학과 교수

2008년~현 재 한국정보보호학회 상임이사

2011년~현 재 한국정보처리학회 이사

2015년~현 재 아주대학교 사이버보안학과 교수

관심분야 : 자동차 보안, 암호프로토콜, 응용시스템보안, 클라우드 컴퓨팅 보안, 개인정보보호, 정보보호제품평가