안드로이드 플랫폼에서 정보 유출 탐지를 위한 지능형 접근 제어 모델 설계

A Design of Intelligent Access Control Model for Information Leakage Detection in Android Platform

2014년 8월 25일

조선대학교 대학원

컴퓨터공학과

성 운

안드로이드 플랫폼에서 정보 유출 탐지를 위한 지능형 접근 제어 모델 설계

지도교수 김 판 구

이 논문을 공학석사학위신청 논문으로 제출함.

2014년 4월

조선대학교 대학원

컴퓨터공학과

성 운

성운의 석사학위논문을 인준함

위원장 조선대학교 교수 정 일 응 (인)

위 원 조선대학교 교수 이 상 응 (인)

위 원 조선대학교 교수 김 판 구 (인)

2014년 5월

조선대학교 대학원

목 차

표 목 차

그 림 목 차

ABSTRACT

A Design of Intelligent Access Control Model for Information Leakage Detection in Android Platform

Woon Sung
Advisor : Prof. Pankoo Kim, Ph.D
Department of Computer Engineering
Graduate School of Chosun University

As smartphones become increasingly popular, attacks on the open-source platform Android are also on the rise. A user's personal information is stored on a smartphone and when the user activates an application which includes malicious code, that user's information can be leaked, with accompanying damage. Therefore a method of detecting and controlling access to a user's personal data within a smartphone is needed.

In this dissertation, categories of applications those operating on the device to detect theft of personal information using smartphone applications and to control access to prevent malicious behavior sold on the Android Market were classified based on package information. An ontology was constructed based on joint permissions of applications in each category as classified here and a security policy defined to detect use exceeding permissions.

The proposed technique was actualized on the Android platform and simulations and results were deduced. It was confirmed to have a lower error false positive rate than previous techniques based on experimental measurements of detection rate and error rate. Since the proposed technique compares application permissions and resource requirements, it has the advantage of being able to quickly detect and control access when new malicious behavior occurs.

Ⅰ. 서론

A. 연구 배경 및 목적

과거 낮은 연산 능력을 지닌 폐쇄형 플랫폼인 피쳐(Feature)폰은 제조사별 OS를 탑재하였기 때문에 서드 파티(third party) 애플리케이션 활용이 제한되었지만 스마트(Smart)폰이 보편화되고, 컴퓨팅파워 기술이 발전함에 따라 범용 OS를 탑재하여 앱 스토어를 통하여 웹 검색, 쇼핑, 메일, 소셜 네트워크 서비스(SNS), 인터넷 뱅킹 등과 같은 다양한 서드 파티 애플리케이션을 제공 받고, 사용자들은 3G, WiFi 등의 통신망을 이용하여 모바일 환경에서 이용할 수 있게 되었다. 이를 통해 스마트폰 사용자는 시공간의 제약 없이 사용자 간 연락이나 문서 작업, 은행 업무를 수행할 수 있게 되었지만 이는 무선 네트워크를 통하여 접속하고 있기 때문에 보안 위협에 상시 노출되고 있다고 할 수 있다.

특히, 스마트폰의 운영체제 중 하나인 안드로이드 플랫폼은 개방성 특징으로 인해 다양한 경로를 통하여 모바일 악성코드가 유입되어 2차적인 보안 문제를 야기한다. 이는 스마트폰의 휴대성으로 인한 기기 내부의 주요 정보가 저장되고, 단말기의 정보 및 위치 정보, 연락처, 메시지, 문서 등의 개인정보를 내포하고 있어 스마트폰의 취약점을 공격하여 개인정보 유출과 같은 보안 위협이 항상 공존하고 있기 때문에 보안 위협이 제거 된 안전한 서비스를 배포 하고, 악성 행위를 적절하게 탐지하기 위한 방안이 필요하다.

최근 스마트폰의 보안 위협에 대응하기 위해 시그니쳐 탐지, 모바일 가상화, 모바일 클라우드 등의 서비스를 이용하고 있으며, 상황정보 기반의 인증 및 권한 관리의 중요성을 강조한 연구가 활발히 진행 중이다.

본 논문에서는 안드로이드 환경에서 기기 및 위치정보, 주소록, 메시지 등의 개인 정보를 유출하는 악성 행위를 탐지하기 위하여 애플리케이션 서비스의 카테고리별 접근 권한(Permission)을 학습하고, 이를 통하여 개인 정보를 외부로 유출하는 API 리소스 정보와 접근 권한의 상황정보 온톨로지 추론을 통하여 신규 악성코드를 탐지하여 기존의 안드로이드 파일 시스템의 임의 접근제어 방식의 리소스 접근에 대한 문제점을 해소하고, 능동적인 보안 수준에 대한 접근이 가능하도록 하

는 지능형 접근제어 모델을 제시하고자 한다. 지능형 접근제어 모델의 제안을 통해 기존의 악성코드의 탐지 기술보다 정확성을 높이고, 접근제어 모델의 실효성을 검증하고자 한다.

B. 논문의 구성

안드로이드 플랫폼에서 개인정보유출 탐지를 위한 지능형 접근제어 모델을 제시하기 위해 본 논문은 다음과 같은 구성으로 작성되었다.

1장 서론에서는 연구의 배경 및 목적에 대해 간략하게 기술하고, 2장 관련연구에서는 스마트폰의 보안 현황 및 스마트폰 악성 행위 분석, 접근제어 모델의 사례와 설계 방법, 그리고 온톨로지 추론 방법에 대해 살펴본다.

3장에서는 개인정보 유출 탐지를 위한 온톨로지 기반의 접근제어 모델을 설계하고 정책에 대하여 기술한다.

4장에서는 기존의 개인정보 접근제어 기법과 비교 평가하고, 제안한 지능형 접근제어 모델의 시나리오를 통하여 성능을 검증한다.

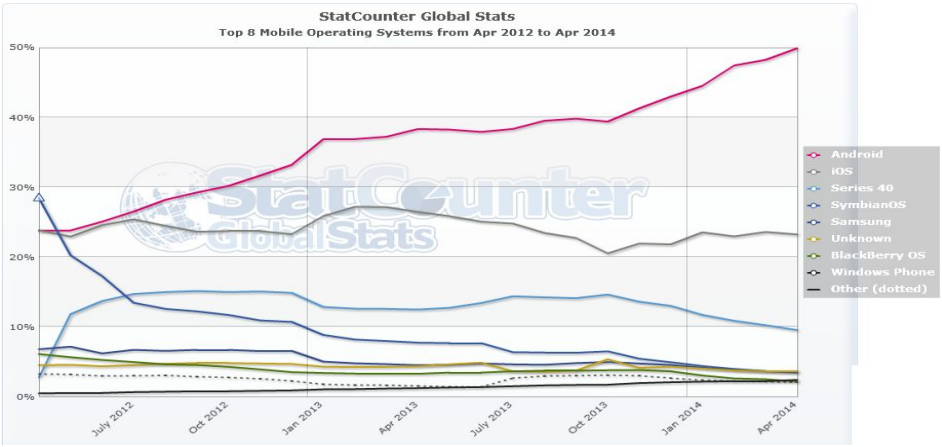
마지막으로 5장에서는 본 연구에 대한 전체적인 결과를 요약하고, 향후 연구 방향을 제시한다.

Ⅱ. 관련 연구

A. 스마트폰 보안 현황

본 논문에서 제안하는 안드로이드 환경에서 개인 정보를 유출하는 행위 탐지를 위한 접근제어 모델을 설명하기에 앞서 본 장에서는 스마트폰의 보안 현황 및 악성 행위 분석 방법에 대해 설명한다.

1. 국내외 보안 동향

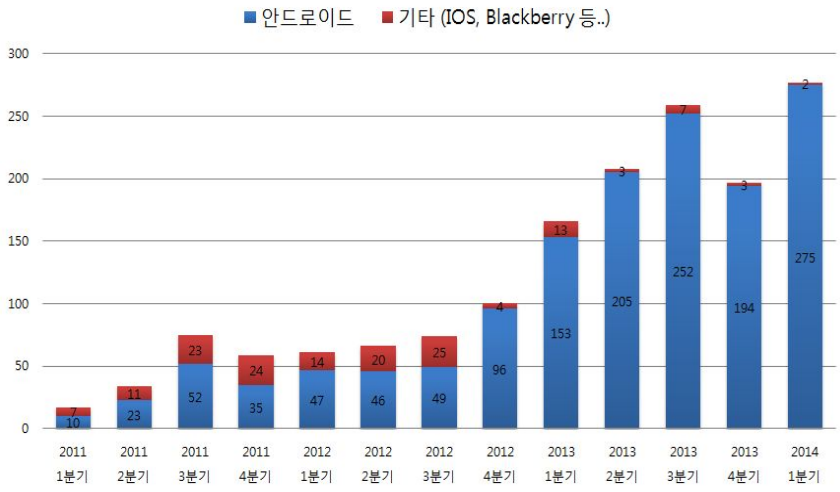


[그림 2-1] 전 세계 스마트폰 OS 점유율[55]

모바일 기기의 컴퓨팅 파워(Computing Power) 능력이 발전함에 따라 전 세계적으로 스마트폰의 보급은 보편화 되었다. [그림 2-1]은 2012년부터 2014년까지의 전 세계 스마트폰 플랫폼의 점유율을 조사한 것이다. 2012년 2/4분기에는 IOS와 안드로이드의 점유율은 비슷한 양상을 보였으나, 안드로이드는 개방형 플랫폼으로 오픈소스를 제공하고 있어 각 제조사들이 독자적인 기능을 커스터마이징하여 개발 및 탑재가 용이하다는 이점을 가지고 있다. 때문에 안드로이드 플랫폼을 이용하는 제조사가 늘어나게 됨에 따라 3/4분기 이후에는 안드로이드 플랫폼의 점유율이 지속적으로 증가하여 현재 국내 90.7%, 전 세계 49.92%로 가장 높은 점유율을 보이고

있으며, 그 뒤를 이어 IOS 23.25%, 기타 26.83%의 플랫폼이 사용되고 있는 것으로 조사되었다[55].

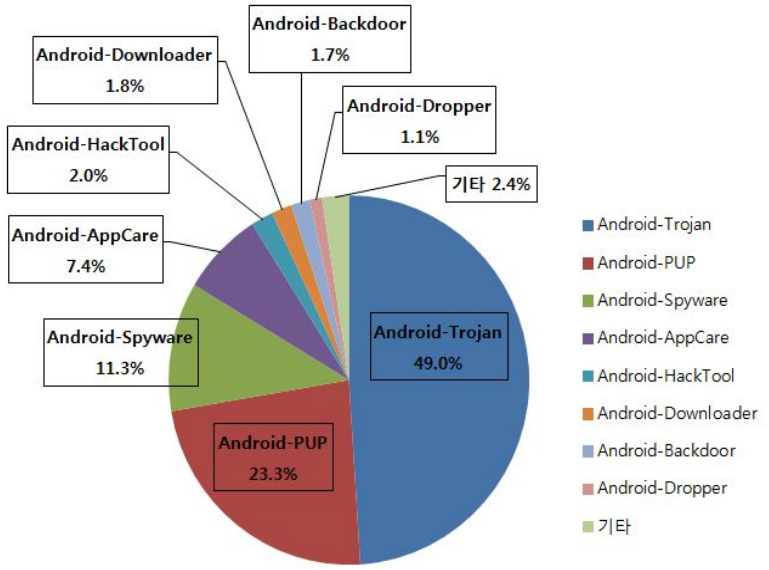
안드로이드의 가장 큰 특징인 개방성은 시장 점유율 확대에 절대적인 공헌을 하고 있지만 안드로이드는 기본적으로 플랫폼이 오픈소스로 공개되어 있어 애플리케이션에 대한 검증 절차가 없고, 멀티태스킹을 지원하기 때문에 보안 문제를 야기한다. [그림 2-2]는 2014년 1분기까지의 스마트폰 신종 악성코드 유형 발견 수를 나타낸다. 2012년 분기별 약 50건 정도였던 악성코드는 2013년에 들어서 약 4~5배 증가하였고, 대부분의 악성코드들이 안드로이드에서 발생하는 것으로 조사되었다[48]. 이를 악용하여 최근에 SMS를 이용한 스미싱(Smishing)이나 악성 애플리케이션을 이용한 개인정보 유출 등 시스템의 취약점을 이용한 보안 사례가 증가하고 있다.



[그림 3-2] 신종 스마트폰 악성코드 유형 발견 수

안드로이드 플랫폼의 서비스 사용 증가는 다수의 사용자 정보를 노리는 악성 행위로 이어졌다. 안드로이드를 타깃으로 하는 악성코드의 배포는 다양한 경로를 통하여 유입 되지만 그 중에서 정식 앱 스토어의 업로드가 아닌 블랙마켓을 통하여 애플리케이션을 이용 시 가장 많은 것으로 조사되었다. [그림2-3]은 AhnLab에서 2014년 1분기 스마트폰 악성코드 동향을 보여준다[49].

악성코드의 분포는 크게 사용자의 스마트폰에 숨어 개인정보를 유출시키고 단말기 장애, 요금 발생 등의 악성 행위를 하는 트로이목마 악성코드가 44.8%를 차지했으며, 사용자가 인지한 프로그램의 설치 목적과 관계없거나 필요하지 않은 프로그램을 설치하는 PUP(Potentially Unwanted Program, 유해 가능 프로그램)이 23.5%, 사용자 정보를 수집하는 스파이웨어가 11.3%를 차지하고 있다.



[그림 2-4] 2014년 1분기 모바일 악성코드 유형별 분포[49]

2. 스마트폰 보안 유형

본 절에서는 스마트폰 환경에서 발생하는 유형 별 정보 유출 경로와 위협 사례에 대해 기술한다.

다음 [표 2-1]은 정보 유출의 경로 및 위협 사례를 나타낸다. 국내외 스마트폰 정보 유출 사례를 보면 대부분의 개인 정보 유출은 악성코드에 의해 발생하고 있으며, 악성코드는 다양한 경로를 통해 유입되고 있다. 대표적인 방법으로 스미싱 또는 사회 공학적 기법을 통해 유입되고 있으며, 탐지 기술이 발전함에 따라 신종 악성코드 기술도 지능화되고 있어 탐지하기는 매우 어려운 입장이다.[17]

[표 2-1] 정보 유출 경로 및 위협

유형	경로	위협 사례
스미싱 (SMS+Phishing)	- SMS을 이용한 URL 접속 유도	- 악성코드 감염 - 요금 유발 - 스팸 광고 노출
사회공학적 기법	- 이메일, SMS 등을 이용하여 정상적인 애플리케이션을 위장한 악성프로그램 배포	- 주소록 정보 유출 - 미디어 정보 유출 - 메시지 정보 유출 - 이동 저장 매체 정보 유출
물리적 요인	- 모바일 기기 도난 및 분실	- 이동 저장 매체의 내부 개인/기업 정보 유출 - 2차 보안 위협 (금융결제, 사기 등)
무선 인터넷	- 악의적인 무선 AP설치 - 내부에 존재하는 취약한 무선 AP 해킹	- 무선망을 이용한 악성코드 감염 및 개인 정보 수집

2.1 악성코드 보안 위협

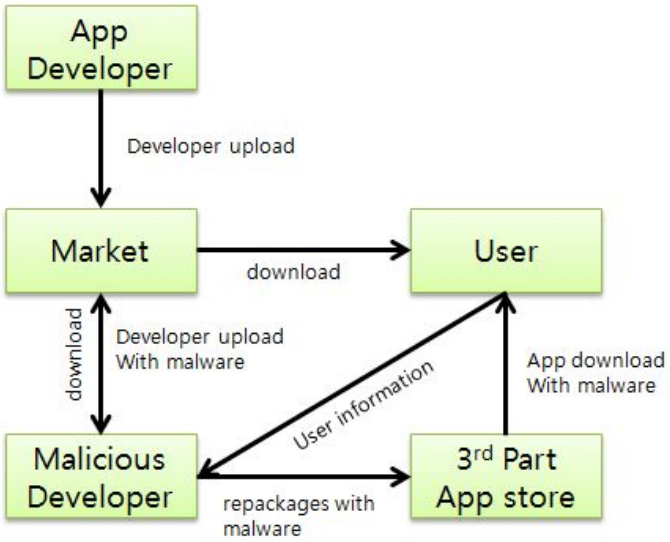
AhnLab의 2013년 4분기 모바일 악성코드 동향 보고서에 따르면 악성코드를 탐지 및 수집한 결과는 다음 [표 2-2]와 같다. 대부분의 악성코드는 안드로이드 플랫폼에서 발견되고 있으며, 트로이 목마와 PUP 유형의 악성코드가 가장 높은 비율을 차지하고 있는 것을 확인할 수 있다.

대표적으로 트로이목마 유형인 Android-Trojan/FakeInst는 기존의 애플리케이션을 사칭하여 단말기의 IMEI 정보를 수집하여 SMS 수신을 통하여 개인정보를 유출하며, Android-PUP/Airpush는 광고 권한을 탑재한 애플리케이션이 실행되지 않았음에도 불구하고 어느 애플리케이션에서 광고를 노출하는지 알 수 없게 백그라운드 서비스로 동작하며 지속적으로 광고를 노출하는 행위를 한다[50].

[표 2-2] 스마트폰 악성코드 TOP 10 [50]

순위	진단명	건수	비율
1	Android-Trojan/FakeInst	158,663	24%
2	Android-PUP/Airpush	90,218	13%
3	Andorid-Trojan/Opfake	49,309	7%
4	Android-PUP/Kuguo	33,730	5%
5	Andorid-PUP/Wapsx	32,890	5%
6	Android-Exploit/Rootor	28,000	4%
7	Android-POP/Plankton	23,329	3%
8	Android-PUP/Leadbolt	22,028	3%
9	Andorid-PUP/Admogo	18,842	3%
10	Android-Trojan/GinMaster	18,214	3%

대부분의 악성코드는 다음 [그림 2-4]의 리패키징(Repacking) 방법을 사용하여 배포를 하고 있으며, 악성코드 제작자는 구글 마켓을 통하여 정상적인 애플리케이션을 다운로드 받고, 이것을 디컴파일(Decompile)을 통하여 악성 행위를 유발하는 코드를 삽입 후 리패키징하여 제3의 마켓을 통하여 재등록함으로써 본래의 애플리케이션의 동작을 수행하면서 백그라운드 상태에서 개인정보를 유출하는 행위를 한다[45,47].

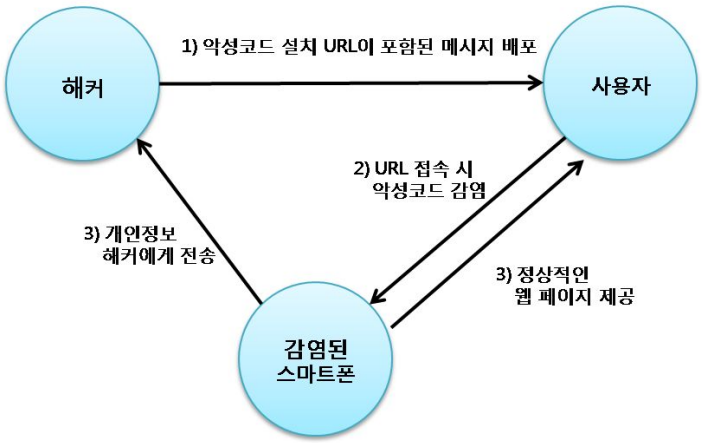


[그림 2-4] 리패키징 방법을 이용한 악성코드 배포과정

리패키징하여 악성코드를 삽입한 애플리케이션의 APK 파일을 분석해보면 기존의 정상적인 애플리케이션의 퍼미션 권한에서 필요하지 않는 이상 권한들을 요청하고 있는 것을 확인할 수 있으며, 이를 통하여 개인 정보를 유출한다. 하지만 기존의 악성행위 탐지 기술들은 악성행위를 유발하는 퍼미션의 존재 여부를 판단하여 악성 애플리케이션을 탐색하기 때문에 정확성이 낮은 문제점을 가지고 있다.

2.2 스미싱(Smishing)을 통한 악성코드 유포

SMS와 피싱(Phishing)의 결합어로 휴대폰 문자메시지를 이용하여 피싱하는 방법이다. 악성코드 유형인 체스트(Android-Trojan/Chest)는 다음[그림2-5]과 같은 과정을 통하여 개인 정보 유출 사례를 발생하고 있다. 스미싱은 [그림 2-4]의 리패키징을 통하여 생성된 애플리케이션이나 제작한 악성 프로그램을 URL 접속을 통하여 사용자 스마트폰에 감염시키는 방식을 사용한다. 이를 통하여 사용자의 스마트폰 내부의 정보를 해커에게 전송하는 역할을 한다[50].



[그림 2-6] 스미싱을 이용한 악성코드 배포 과정

AhnLab의 2013년 4분기 모바일 악성코드 동향 보고서에 따르면, 스미싱을 이용한 악성코드 배포 유형은 다음 [표 2-3]과 같다[50].

[표 2-3] 스미싱을 통한 악성코드 유형

순위	악성코드명	순위	악성코드명
1	Android-Trojan/Chest	6	Android-Spyware/PhoneSpy
2	Android-Trojan/SMSstealer	7	Andorid-Spyware/Msgintercept
3	Android-Trojan/Bankun	8	Android-Spyware/Langya
4	Android-Downloader/Bankun	9	Andorid-Spyware/Tmphone
5	Android-Trojan/Meteor	10	Android-Trojan/KorTalk

B. 스마트폰 악성행위 분석

본 절에서는 기존의 정적 분석을 통하여 악성 행위 분석 방법에 대한 설명을 하고, 개인 정보 유출을 하는 악성코드의 특징을 살펴보고, 접근제어 정책을 정의하기 위한 악성코드 특징을 기술한다.

1. 악성코드 분석 기법

악성코드 분석은 악성코드 탐지의 기준이 되는 특성을 추출하는 과정으로 분석 기법은 크게 정적 분석(Static Code Analysis)과 동적 분석(Dynamic Analysis)으로 구분된다[40,45,47].

[표 2-4] 악성코드 분석 기법 비교

분류	정적 분석	동적 분석
장점	- 프로그램 전역에 대한 분석 가능 - 낮은 유지비용 - 요구 권한과 악성코드 은닉여부에 따른 악성행위 검증 가능	- 실제 실행결과를 통한 악성행위에 대한 정확한 탐지
단점	- 허위탐지와 미탐지 한계 - 코드 난독화로 인한 분석 어려움	- 시간 및 하드웨어 자원 등에 대한 높은 유지비용 - 테스트 케이스의 한계

1.1 정적분석(Static Code Analysis) 기법을 통한 분석

정적 분석 기법은 안성코드를 실행 시키지 않고 코드 분석으로만 악성코드 여부를 탐지하는 기법이다. 다음 악성코드 샘플[52]를 이용하여 온라인 정적분석[53] 도구를 사용하여 분석한 방법 및 결과는 다음과 같다.

① 악성코드 은닉 여부 검사

악성코드는 이미지 파일이나 XML 파일로 가장하여 포함되어 있으며, URL을 통한 악성코드 감염 또는 유출 행위가 발생할 수 있기 때문에 URL 검사를 통하여 악성코드 은닉 여부를 검사한다.

[표 2-5] 악성코드 URL 분석 결과

Urls
http://www.adview.cn/agent/agent1_android.php?appid=%s&appver=%d&client=0
http://www.adviw.cn/agent/agent3.php?appid=%s&nid=%s&type=%d&uuid=%s&country_code=%s&appver=%d&client=0
http://beta.vpon.com/api/api/webviewAdReq
http://cn.ad.adon.vpon.com/api/webviewAdClick
http://www.adview.cn/agent/agent2.php?appid=%s&nid=%s&type=%uuid=%country_code=%s&appver=%d&client=0
http://wap.casee.cn/mo/siteAd.ad?b=1&m=
http://lebar.gicp.net/more.aspx?pid=9973
http://androidsdk.ads.mp.mydas.mobi/getAd.php5?
http://d.wiyun.com/adv/s?
http://maps.google

② 애플리케이션에 대한 요구권한 분석

안드로이드 환경에서 권한 요청 시 all-or-nothing 정책을 사용하고 있으며, 애플리케이션을 설치 시에 사용자에게 단 한번의 권한 요청을 하게 되며, 요청을 수락하게 되면 애플리케이션을 삭제 할 때 까지 유지된다. 다음 권한의 기본 보호 수준은 [표 2-6]과 같이 네 가지의 권한 수준으로 구성되어 있다[51].

[표 2-6] 안드로이드 권한 보호 수준

수준	분류	설명
0	Normal	최소한 위험 수준으로 명시적으로 권한 부여 요청 없이 자동으로 허용된다.
1	Dangerous	사용자에게 부정적인 영향을 미칠 수 있는 개인데이터나 디바이스의 제어에 대한 접근 요청을 부여하는 권한이다.
2	Signature	이전에 권한이 부여된 애플리케이션에서 사용한 동일한 인증서를 통해 권한 요청을 하는 경우에 권한을 허용한다.
3	Signature|system	시스템은 안드로이드 시스템 이미지 패키지나 동일한 인정서로 서명된 경우 권한을 허용한다.

악성코드 샘플을 이용하여 AndroidManifest.xml 파일을 분석한 결과는 다음 [표 2-7]과 같다. 총 사용한 퍼미션의 개수는 15개의 권한을 사용하고 있으며, 퍼미션 권한을 분석해보면 0-수준(Normal)의 권한 3개와 개인정보의 리소스 접근을 요청하는 1~3 수준의 퍼미션으로 구성되어 있다. 이러한 분석을 통하여 퍼미션 요청 권한을 기준으로 악성행위 여부를 판단하여 분류할 수 있다. 하지만 정상적인 애플리케이션이 악성코드와 유사한 권한을 요청하는 경우 이를 판단하는 모호성의 문제가 발생한다. 이를 더욱 정확하게 분류하고, 분석하기 위하여 [표 2-8]에서의 개인정보에 민감한 정보를 요청하는 API 함수의 호출 여부를 분석하여 판단한다.

[표 2-7] 악성코드 퍼미션 분석 결과

permission	위험 수준	설명
android.permission.ACCESS_FINE_LOCATION	1	GPS 위치 정보
android.permission.SEND_SMS	1	SMS 메시지 보내기
android.permission.RECEIVE_BOOT_COMPLETED	0	부팅시 자동실행
android.permission.INTERNET	1	인터넷 엑세스
android.permission.INSTALL_PACKAGES	1	패키지 설치
android.permission.PROCESS_COUTGOING_CALLS	1	브로드케스팅 수신
android.permission.ACCESS_WIFI_STATE	0	WiFI 상태 확인
android.permission.ACCESS_COARSE_LOCATION	1	GPS 위치 정보
android.permission.CALL_PHONE	1	전화 걸기
android.permission.DELETE_PACKAGES	1	패키지 삭제
android.permission.ACCESS_NETWORK_STATE	0	네트워크 상태 정보
android.permission.READ_PHONE_STATE	1	폰 상태 읽기
android.permission.READ_SMS	1	SMS 메시지 읽기
android.permission.WRITE_EXTERANL_STORAGE	1	저장 매체 저장
android.permission.RECEIVE_SMS	1	SMS 메시지 받기

악성코드의 분석을 통하여 악성코드에서 주로 포함되어 있는 퍼미션 권한을 살펴보면 다음 [표 2-8]과 같이 정의할 수 있다. 이 도표를 기준으로 악성 행위를 유발할 수 있는 퍼미션 조합 패턴을 통해 카테고리별로 학습하고, 접근제어 정책을 정의 한다[33,37,40].

[표 2-8] 개인정보 유출 관련 퍼미션

권한 수준	Permission	설 명
1	INTERNET	인터넷 접근
1	ACCESS_NETWORK_STATE	네트워크 상태 보기
0	ACCESS_WIFI_STATE	WiFi 상태 접근
1	CHANGE_WIFI_STATE	WiFi 상태 변경
1	READ_SMS	SMS 메시지 읽기
1	WRITE_SMS	SMS 메시지 쓰기
1	SEND_SMS	SMS 메시지 보내기
1	RECEIVE_SMS	SMS 메시지 받기
1	READ_PHONE_STATE	통화 상태 읽어오기
1	CALL_PHONE	전화 걸기
1	WRITE_EXTERNAL_STORAGE	외부 저장 매체 쓰기
0	READ_EXTERNAL_STORAGE	외부 저장 매체 읽기
0	RECEIVE_BOOT_COMPLETED	부팅 시 자동시작
1	ACCESS_COARSE_LOCATION	기지국 정보를 통한 위치 확인
1	ACCESS_FINE_LOCATION	WiFi를 통한 위치 확인
1	READ_CONTACTS	주소록 읽어오기
0	WAKE_LOCK	절전모드에서 WakeLock 허용
1	WRITE_CONTACTS	주소록 쓰기
3	WRITE_APN_SETTINGS	APN설정 쓰기
0	RESTART_PACKAGES	패키지 리스타트
1	INSTALL_PACKAGES	패키지 인스톨
1	DISABLE_KEYGUARD	락 화면 제거
1	GET_TASKS	태스크 정보
3	READ_LOGS	로그 읽어오기
1	READ_HISTORY_BOOKMARKS	웹 즐겨찾기 등 권한
1	WRITE_HISTORY_BOOKMARKS	웹 즐겨찾기 쓰기 등 권한
3	MOUNT_UNMOUNT_FILESYSTEMS	파일시스템 편집
1	INSTALL_SHORTCUT	홈 스크린에 아이콘 생성 권한

③ 소스코드 분석을 통한 악성행위 검증

APK 파일의 Class파일 분석을 통해 악성행위를 유발할 수 있는 API 조합으로 구성된 패턴의 사용 여부를 확인하여 악성행위를 검증한다. 다음 [표 2-9]는 악성코드 샘플을 이용하여 분석하여 개인정보를 유출하는 API 리스트를 추출한 결과이다. 악성코드들은 퍼미션 접근권한을 수락 시 스마트폰의 단말기 또는 전화번호 정보를 얻어와 URL이나 SMS를 통하여 개인 정보를 유출하는 다수의 민감한 정보를 이용하는 API 함수가 포함되어 있다[37].

[표 2-9] 악성코드의 정보유출 관련 API

API	설명
android/telephony/TelephonyManager;->getSimSerialNumber	Get SIM Serial Number
android/telephony/SmsManager;->sendTextMessage	Send Text Message
java/net/HttpURLConnection;->connect	HTTP Connect
java/net/URL;->openConnection	Open URL Connection
android/content/ContentResoiver;->query	Qquery Database of Contacts, SMS, etc
android/appNotificaitonManager;->notity	Notification
java/net/URLConneciton;->connect	URL Connect
java/lang/Rutime;->exec	Execute Command
android/telephony/TelephonyManger;->getDeviced	Get Device IMEI and Cellphone Number
android/telephony/TelephonyManager;->getLine1Number	Get Cellphone Number
httpClient;->execute	Request Remote Server
DefaultHttpClient;-.execute	Send HTTP Request

개인 정보를 요청하는 리소스를 통하여 상황 정보별 분류를 통해 [표 2-10]의 API Blacklist를 정의하고 온톨로지 모델링을 통해 개인 정보에 접근을 허가하는 권한을 사용 시 API Blacklist와 비교를 통하여 개인 정보를 유출하는 API가 포함 여부를 검사한다. 이 비교를 통하여 악성행위의 유발 가능성을 검증한다[22,37].

[표 2-10] 개인정보 유출 API 특성 예제

분류	함수	기능
단말기 정보	getLine1Number	주소록 조회
	getSubscriberId	IMSI(가입자 식별자 번호) 정보
	getDeviceId	IMEI(단말기 고유 번호) 정보
	getNetworkOperator	망 사업자코드 조회
	getNetworkOperatorName	망 사업자명 조회
	getSimSerialNumber	SIM카드 시리얼 번호 조회
	getPackageManager	애플리케이션 설치 정보
	getInstalledApplications	설치된 애플리케이션 정보
네트워크 정보	isWifiEnabled, getWifiState	WIFI 상태 확인 및 활성화
	setWifiEnabled, setWifiState	WIFI의 상태를 켜짐/커짐으로 변경
	url.openConnection	URL간의 통신 링크.
	openStream, InputStream.read	URL로부터 읽은 파일 다운로드
	getNetworkInfo	네트워크 종류 정보
	HttpURLConnection	웹 페이지 접근
SMS	sms.all.CONTENT_URI	SMS 전체 데이터에 접근
	sms.inbox.CONTENT_URI	SMS 수신 데이터에 접근
	sms.sent.CONTENT_URI	SMS 발신 데이터에 접근
	sms.draft.CONTENT_URI	SMS 미완성 저장함 데이터에 접근
	sms.outbox.CONTENT_URI	SMS 미완성 보관함 데이터에 접근
	sendTextMessage	SMS 전송하는 함수
	getMessageBody	저장된 SMS를 읽어오는 함수
위치정보	getLatitude. getLongitude	현재 위치 위도, 경도 정보
	getFromLocation getFromLocationName	GeoCoder 사용한 위치 정보

C. 안드로이드 환경에서 접근제어모델(ACM)

1. 접근제어모델(Access Control Model) 정의

[그림 2-6] 접근제어 구성 요소

스마트폰 내부의 저장되는 데이터의 증가에 따라 민감한 정보들이 권한이 없는 사용자들에 의해 유출, 변조될 위험성이 증가하였다. 이로써 역할에 따라 권한에 맞는 정보에 접근하기 위한 접근제어의 필요성이 요구되고 있다.

접근제어(Access Control)란 사용자나 프로세서 등의 "Subject"가 사용자의 개인 정보를 포함하고 있는 데이터베이스 또는 파일 등을 포함하고 있는 "Object"의 정보를 읽기(Reading), 쓰기(Writing), 실행(Execution)의 기능을 접근 여부를 허가하거나 거부하는 기능을 말한다[22,24,25].

2. 임의적 접근제어(Discretionary Access Control : DAC)

안드로이드 환경에서는 파일 시스템 접근을 DAC(Discretionary Access Control) 방법을 사용하고 있으며, 사용자의 식별에 기초하여 사용자의 접근 권한을 임의 적으로 추가 또는 삭제 할 수가 있다. 이를 통해 사용자는 임의로 파일의 읽기, 쓰기, 실행 권한을 부여할 수 있기 때문에 악의적인 사용자가 DAC 권한을 획득하면 데이터 파일 및 리소스에 무단으로 접근을 할 수 있기 때문에 보안에 취약하다.

임의 접근제어 정책에서는 접근 제한의 설정이 주체와 객체 단위로 설정할 수 있으며, 객체를 소유하고 있는 사용자에 의해서 접근 제한 변경이 가능하다. 하지만 임의 접근제어 기반의 환경에서는 커널 리소스를 접근제어하기 때문에 특정 애플리케이션이 실행되지 않을 수 있는 문제점이 발생한다.

3. 상황정보를 이용한 접근제어

기존의 역할 기반 접근제어는 상황 정보에 근거한 접근제어를 수행 할 수 없기 때문에 이를 보완하고자 개선된 역할 기반 접근제어 방법으로 일반 역할기반 접근제어(GRBAC) 모델[3]과 상황 제약 역할기반 접근제어(xoRBAC) 모델이 제안되었다[29].

GRBAC 모델은 기존의 역할 기반의 접근제어를 개선하여, 사용자 역할(Subject Role), 객체 역할(Object Role), 환경 역할(Environment Role)을 사용하고 있다. 이 역할을 구조화함으로써 정책 기술의 단순함과 융통성을 제공하고 있다[3]. 이를 통하여 보안 관리자는 접근 권한의 정책 정보 역할(Subject Role), 객체 역할(Object Role), 환경 역할(Environment Role), 연산(operation), 접근기호(sign)의 5가지의 요소를 통하여 정의한다.

```
<<SRole,ORole,ERole,op>,sign>
<<app_category,infor_sms,music,read>,->
```

예를 들어, 위와 같은 표현에서 app_category의 주체 역할을 할당받은 사용자가 music이라는 애플리케이션에서 infor_sms, 즉 SMS의 열람을 할 수 없음을 나타낸다. 또한, 역할 계층 구조에서 발생하는 비명시적인 권한 부여 문제를 해결하기 위하여 역할 계층구조에서의 권한 상속 개념을 사용하고 있으며, 권한 상속의 타입은 Standard, Strict, Lenient로 구분된다. GRBAC 모델은 상황 정보를 추가하여 환경 역할로 정의하고, 접근제어 정책을 정의한다. 하지만 접근권한의 전달로 권한들 사이에 충돌에 대한 문제점이 발생한다. 이러한 문제점을 GRBAC 모델에서는 해결 방안을 제시하고 있지 않기 때문에 사용자가 고려사항에 대한 환경 역할로 정의함에 따라 많은 계층 구조가 발생하여, 설계와 관리의 어려움이 따른다[3,29].

4. 강제 접근 정책 기반의 안드로이드 보안

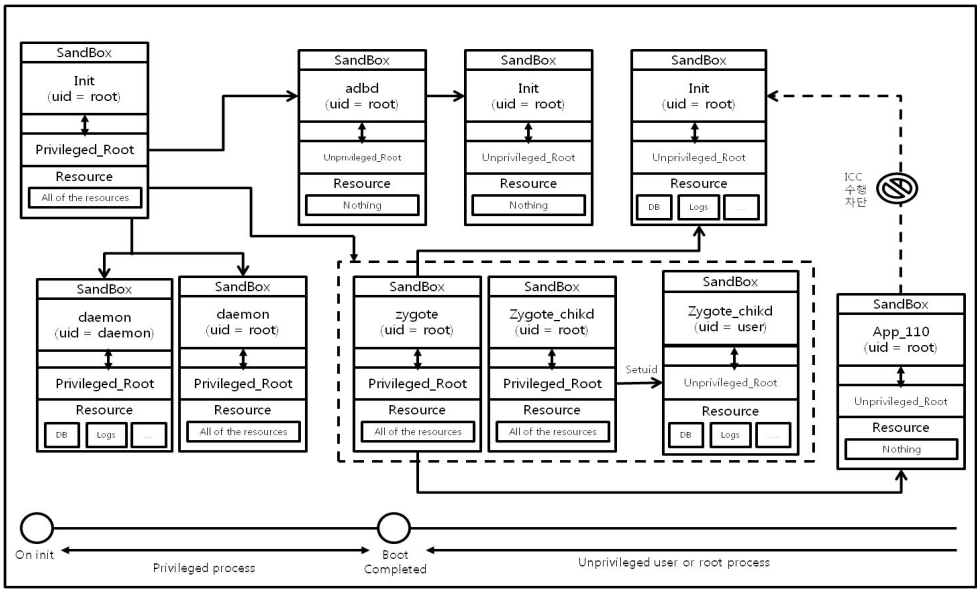
기존의 임의 접근제어(Discretionary Access Control) 기반의 안드로이드 파일 시스템의 접근제어 프레임워크를 보완하기 위해 강제 접근정책을 적용한 방법이다.

① Privileged Root(PR) : 시스템 부팅 완료 시점 이전의 합법적인 모든 루트 권한을 가진 프로세스

② Unprivileged Root(UR) : 부팅 시점 이후에 간헐적으로 실행되는 adbd와 같은 불법적으로 루트 권한을 가진 것으로 판단되는 프로세스

③ Unprivileged User(UU) : 루트 권한이 없는 일반 애플리케이션 프로세스

접근 제어 테이블의 객체는 해당 주체가 접근하는 애플리케이션 파일들로 나타내고 있으며, [그림 2-7]은 DAC 기반의 안드로이드 보안 프레임워크 상에서의 PR, UR, UU, 프로세스들로 분류하는 시나리오를 보여주고 있다.



[그림 2-7] 강제 접근 제어를 적용하기 위한 안드로이드[3]

[그림 2-7]에서 분류된 프로세서들은 다음의 접근제어 규칙을 이용하여 접근제어 테이블로 표현한다.

첫째, PR(Privileged Root) 프로세스는 파일 시스템의 모든 데이터베이스 파일을 접근을 허용한다.

둘째, UR(Unprivileged Root) 프로세스는 모든 파일에 대하여 접근을 거부한다.

셋째, UU(Unprivileged User) 프로세스는 사용자의 임의 접근 제어의 퍼미션 권한 요청을 통하여 허가할 경우 모든 데이터를 접근한다.

하지만 이러한 미리 일괄적으로 설정해 놓은 강제 접근제어 정책은 신규 애플리케이션에 대해 고정적인 정책 테이블을 사용하기 때문에 일치하지 않는 정책이 적용되어 서비스를 이용할 수 없는 문제점이 발생한다[3].

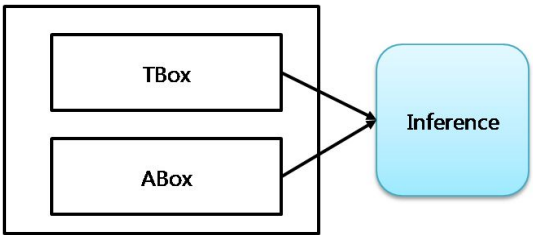
D. 온톨로지 추론 기술

1. 논리표현 및 추론

추론이란 사전에서 이미 알고 있는 사실들을 기반으로 결론에 도달하기 위한 행위 또는 프로세스를 논리적으로 유도하는 과정이라 명시한다. 온톨로지 기반 추론의 종류의 하나인 일차논리(First Order Logic : FOL)은 명제논리(Propositional Logic)의 제약을 보완하기 위한 방법이다.

1.1 Description Logic(DL)

지식의 표현과 공유를 위하여 개발되어온 서술논리는 실세계 환경의 도메인에 대한 지식에 대하여 표현하기 위한 언어로서 일반적인 지식 표현과 개념의 관계를 정의할 수 있다. 따라서 현실에서의 대용량의 데이터를 사람을 대신하여 기계가 처리 가능하도록 하기 위해 추론을 통하여 새로운 의미 정보를 도출할 수 있다. 서술논리를 이용한 지식베이스는 [그림 2-8]과 같이 크게 두 개의 구성 요소로 이루어진다. 개념 및 용어에 관한 지식을 포함하는 용어적 공리 TBOX와 개념에 포함 될 수 있는 직접적인 실세계의 데이터 개체들의 선언에 관한 지식을 표현하고 있는 선언적 공리 ABOX로 구성된다. 이러한 구조를 바탕으로 구성된 지식베이스는 서술논리 언어를 이용하여 표현되며, 추론엔진은 명시적(Explicit)으로 표현된 지식으로부터 추론(Reasoning)을 통하여 새로운 지식 정보를 도출 할 수 있다[58].



[그림 2-9] 서술 논리 구성

1.2 혼 논리(Hron Logic)

혼 논리(Hron Logic)은 혼 문장(Horn sentence)에 의해 논리가 표현되는 것으로, 혼문장은 ¬이 선행되지 않은 모든 원소들(Positice atom)이 ∧로 연결되어진 문장이다.

[표 2-11] 혼 논리 표현

$P_1 \wedge P_2 \cdots \wedge P_n \Rightarrow Q$
(where P and Q are nonnegated atoms)

혼 문장은 두 가지 특별한 경우가 있다. 첫째, Q가 상수 “False”일 때, $\neg P_1 \vee \neg P_2 \vee \cdots \vee \neg P_n$과 동등한 문장을 얻을 수 있다. 둘째, n=1이고 P1=true일 때 True ⇒ Q라는 기본문장 Q와 동일한 문장을 얻을 수 있다. 모든 지식표현이 혼 문장으로 표현 될 수 없지만, 혼 문장은 추론과정을 간단하게 해주는 장점이 있다[58].

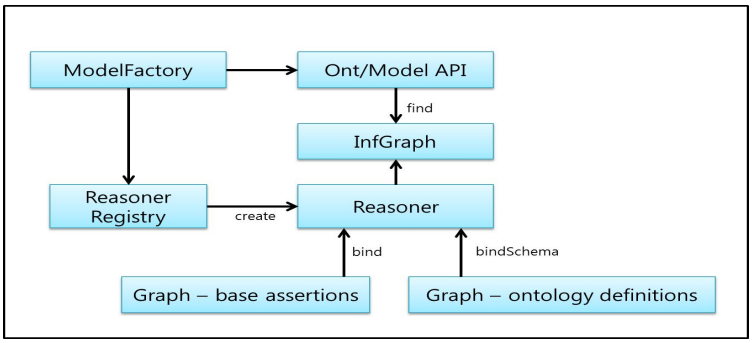
2. 온톨로지 추론 엔진

추론엔진은 온톨로지로부터 얻은 지식 정보의 관계를 통하여 새로운 사실을 유추하는 기능을 실행한다. 이는 추론엔진을 통해 지식 정보에 대한 질의를 통해 추론엔진은 질의어와 관계를 파악하여 관련 정보를 전달하는 방식을 사용하고 있다. 대표적인 추론엔진 도구는 Hoolet, F-OWL, Jena 등을 관련 연구에서 많이 활용하고 있다. 본 논문에서는 Protoge를 사용하여 온톨로지를 모델링하여 OWL 데이터를 사용하였으며, 온톨로지 도구 비교분석[57]에 따르면 Protege와 가장 호환성이 높고, OWL데이터에 대한 간단한 물리적 스키마 구조로 구성되어 있어 Jena 추론엔진을 선택하였다. [그림 2-9]는 Jena 추론엔진의 구성요소를 나타낸다. Jena는 Java 기반의 프레임워크를 사용하고 있으며, 단순한 물리적 스미카 구조의 형태로 되어 있어 많은 간단한 방법으로 구축할 수 있기 때문에 상황정보 시스템의 연구에서 많이 사용되고 있다.

[표 2-12] Jena를 이용한 추론 예제

```
SmartPhone(?a) ∧ Company(?b) ∧ Product(?c)
∧ isManufacturerOf(?b, ?a) ∧ isItemOf(?b, ?c)
→ isProductOf(?c, ?a)
```

[표 2-12]는 스마트폰를 Jena RDF API를 이용하여 정의한 상황 추론 예제이다. 각 클래스는 인스턴스의 관계를 통해 Product(c)는 SmartPhone(a)의 isProductOF(c, a)라는 추론의 결과를 보여준다.



[그림 2-10] Jena API Inference [56]

Ⅲ. 온톨로지 기반 접근제어 모델 설계

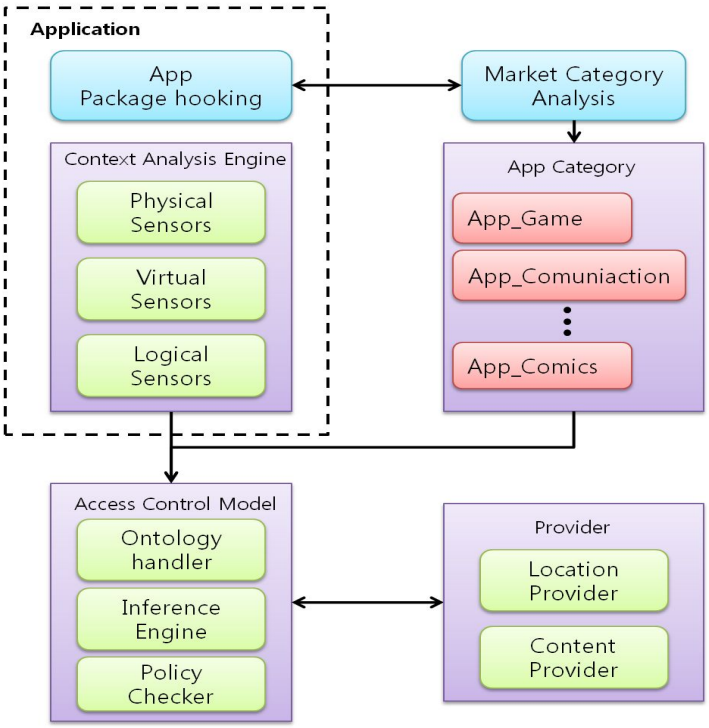
스마트 디바이스 발전으로 사용자는 다양한 애플리케이션 서비스를 제공 받을 수 있게 되었다. 그로인해 제공되는 서비스의 종류가 크게 증가한 만큼 악성행위의 종류와 방법도 기하급수적으로 증가하고 있다. 이에 기존의 안드로이드 환경에서 권한을 부여하는 퍼미션 시스템에서는 실행 중에 할당된 퍼미션을 동적으로 회수하는 것이 불가능하며, 사용자가 모든 퍼미션을 허가하여 설치를 허용하거나 모두 거절하여 설치를 거부하는 all-or-nothing decision의 방식을 사용한다[11].

스마트폰 환경에서는 개인정보 데이터베이스에 접근하는 권한에 대하여 보다 세밀하고 동적인 접근제어를 위해 다음과 같은 조건들이 만족해야 한다. 접근제어를 위한 조건은 사용자 권한 접근으로 인한 권한 위임 시에 동적으로 부분적으로 위임을 할 수 있어야 하며, 보다 동적인 접근제어를 위해 허가 역할의 제약을 고려할 수 있어야 한다. 또한 개인정보 데이터베이스의 정보 보호를 위하여 데이터 접근에 대한 조건, 의무사항들을 고려하여 필요에 따라서 접근이 허용하거나 거절 될 수 있어야 한다[13].

본 논문에서는 이러한 안드로이드 환경에서 개인정보 데이터베이스 접근에 관한 권한 부여에 따른 지능형 접근제어 방식을 이용하여 악성행위에 대하여 탐지를 제시한다. 구글 마켓의 애플리케이션의 카테고리별 정보를 통하여 실행되고 있는 애플리케이션을 분류하고, 카테고리별 사용하는 특정 권한에 대하여 관계를 설정함으로써 악성행위를 분류 하고 탐지하는데 효과적인 것을 확인하였다. 이를 통해 새로운 변종 악성행위에 대한 애플리케이션 탐지의 오탐율을 줄일 수 있는 방안에 대하여 제시한다.

A. 지능형 접근제어를 위한 프레임워크 설계

본 장에서는 개인정보 유출 탐지를 위한 온톨로지 기반의 접근제어 모델의 프레임워크와 수행과정에 대해 설명한다.



[그림 3-1] 지능형 접근제어 프레임워크

전체적인 프레임워크의 구성은[그림 3-1]과 같이 구성되어 있다. 먼저 애플리케이션 카테고리별 퍼미션 분석을 통하여 특정 권한에 대한 온톨로지 모델링을 하고, 실행중인 애플리케이션의 카테고리 분류를 하기 위하여 실행중인 서비스의 패키지명을 추출하여 구글 마켓의 정보를 통해 애플리케이션 카테고리별 분류를 한다. 이는 카테고리를 분류하는 모듈과 애플리케이션에서 요청하는 상황정보를 수집하고 관리하는 Context Analysis Engine, 스마트폰 환경에서 사용자 인증, 상황정보에 따른 접근제어 등을 통해 보안 서비스를 관리하는 Access Control Module로 구성되어 있다[7].

① Category Analysis

카테고리별 온톨로지 추론을 위하여 스마트폰 환경 내의 실행되고 있는 애플리케이션의 포어그라운드 서비스의 패키지명을 추출하고, 카테고리를 분류하기 위하여 추출한 패키지명의 정보와 구글 마켓에 등록되어 있는 애플리케이션의 카테고리 정보를 이용하여 현재 실행되고 있는 서비스의 애플리케이션을 25개의 카테고리로 분류한다.

② Context Analysis Engine

스마트폰 내의 환경 상황정보를 수집 및 관리하며 Virtual Sensor는 SNS 및 사용자 일정 등과 같이 가상의 공간에 저장되어 있는 각종 상황 정보를 수집하는 역할을 하고, Physical Sensor는 온도, 습도, 속도와 같은 물리적인 상황정보를 수집하고, Logical Sensor는 다른 센서들로부터 수집된 상황정보를 가공하여 새롭게 유추된 상황정보를 수집하는 역할을 한다. 상황 정보에 따른 리소스 사용 요청을 받으면 상황 정보 분석 단계에 상황정보의 요청을 받으면, 정보 수집을 위한 질의 메시지를 생성하여 센서 네트워크에 전송한다. 그리고 수집한 상황정보를 보안 정책에 사용할 수 있는 정보로 통합, 가공한다.

③ Access Control Module

Access Control Module은 스마트폰 환경에서 상황 정보에 따른 사용자 인증 및 접근제어의 보안 서비스를 관리한다. 관리 도메인 내에서 애플리케이션이 리소스에 대한 사용 요청이 발생하면, ACM에서는 Context Analysis Engine을 통해 애플리케이션에서 요청한 리소스 정보에 따른 보안정책과 추론을 통하여 접근제어를 한다.

④ Inference Engine

Inference Engine은 권한, 인증 서비스, 상황 온톨로지 저장소로 구성되어 있다. 각 카테고리별 보안정책을 관리하고, 사용자의 리소스 접근 상황 역할을 추론하여 정보 자원 접근 요청에 따른 접근 제어를 수행한다.

⑤ Ontology Handler

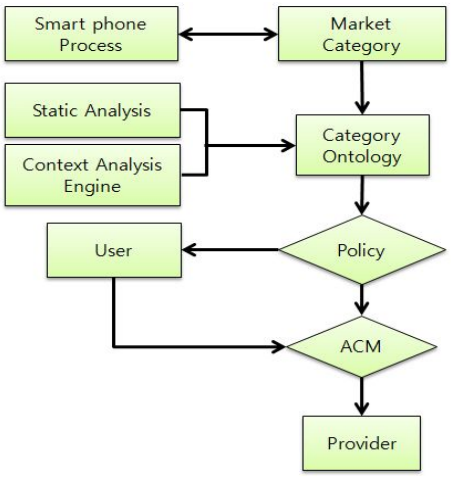
Ontology Handler는 사용자가 접근하려는 카테고리별 권한 서비스, 리소스 서비스로부터 데이터 처리를 통하여 분석한 상황 정보를 저장하는 상황 온톨로지를 관리한다. 상황 온톨로지는 접근요청이 포함된 트랜잭션(Transaction) 리스트와 각각 트랜잭션의 허가 여부를 나타내는 승인 정보 정책을 저장하고 있다. 상황정보 온톨로지는 사용자의 상황 정보의 수집 및 분석을 위해 OWL 파일 형태로 기술하고, 추론 엔진을 통해 가장 적합한 접근 제어 정책을 추론한다.

⑥ Policy Checker

지능형 접근제어 시스템에 접근하려는 주체에 대한 권한 확인과 주체의 리소스 정보 관리 및 처리를 담당한다. 또한 리소스에 접근하기 위해 주체의 카테고리별 권한과 리소스 요소 등에 대한 추가 정보를 획득 및 정책의 분석을 통하여 애플리케이션의 역할을 동적으로 할당하는 서비스를 제공한다.

1. 접근제어 과정

본 논문에서 제안하는 정보유출 탐지를 위한 지능형 접근제어의 수행과정은 다음[그림 3-2]와 같다.



[그림 3-2] 지능형 접근제어 흐름도

안드로이드 시스템은 내부 데이터베이스에 대하여 애플리케이션 간의 접근 가능한 보안성과 유연성을 제공하기 위하여 각 데이터베이스별 Provider를 제공 하고 있다. 이에 Provider의 접근 여부를 판단하기 위하여 본 논문에서는 다음 [그림 3-2]의 수행과정을 걸치게 된다.

먼저 현재 안드로이드 환경에서 실행 중인 포어그라운드 서비스의 패키지명을 추출하고, 추출한 애플리케이션의 패키지명을 이용하여 구글 마켓에 등록되어 있는 카테고리 정보를 파싱하여 해당 실행 중인 애플리케이션을 카테고리를 분류 한다.

분류된 애플리케이션 카테고리의 권한 특성 패턴을 학습하기 위하여 정적 분석을 통하여 해당 카테고리별 요청하는 주요 권한 정보를 분석하고, 상황 정보를 이용한 카테고리별 온톨로지를 구축하여 정책을 정의 한다. 이때, 사용자가 애플리케이션을 통하여 내부 데이터베이스를 요청하게 되면 애플리케이션의 카테고리별 요청하는 권한의 정보를 바탕으로 악성 행위를 유발할 수 있는 퍼미션과 악성API의 추론을 통하여 보안 정책에 위반하는 경우 사용자에게 확인 요청을 통하여 리소스에 대한 접근을 허용하거나 차단한다.

2. 카테고리 분석

다음 [표 3-1]과 [표 3-2]는 안드로이드 환경에서 실행 중인 애플리케이션 서비스의 패키지 ID값을 추출하여 구글 마켓의 등록되어 있는 애플리케이션 카테고리의 정보를 파싱하여 카테고리를 분석하는 Java 코드이다.

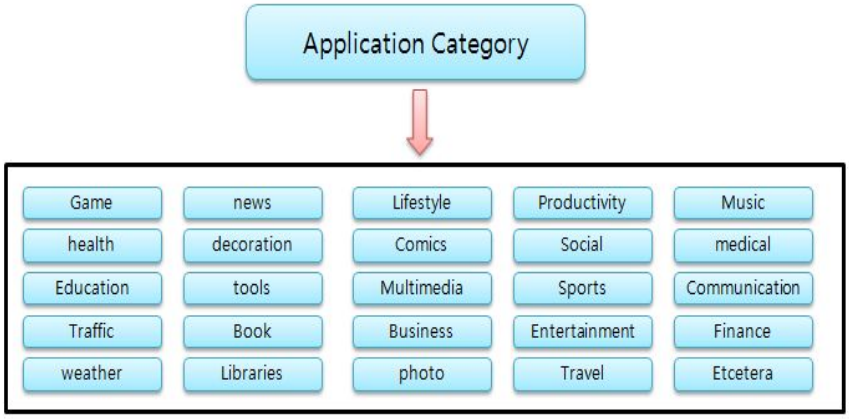
[표 3-1] 실행중인 서비스 패키지명 추출

```
protected void onPostExecute(Void result) {
    TextView txttitle = (TextView) findViedById(R.id.titletxt);
    ActivityManager am = (ActivitiyManger)getApplicationContext()
                        .getSystemService(Context.ACTIVITY_SERVICE);
List<ActivityManager.RunningServiceInfo> rs = am.getRunningServices(50);
  for(int i=0;i<rs.size();i++){
      ActivityManager.RunningServiceInfo rsi = rs.get(i);
      txttitle.setText(rsi.service.getPackageName());
      mProgressDialog.dismiss();
      Log.d("run service", "Package name : " + rsi.service.getPackageName());
  }
}
```

[표 3-2] 구글 마켓을 통한 카테고리 추출

```
protected void doInbackground(Void... params){
    try{
        Document document = Jsoup.connect(url).get();
        Elements desription = document.select(".category");
        desc = description.attr("href");
        } catch (IOException e){
          e.printStackTrace();
     } return null;
}
```

다음 [그림 3-3]은 애플리케이션의 카테고리 분류를 보여준다. 애플리케이션의 카테고리별 주요 Permission 정보를 분석하기 위하여 구글 마켓을 통하여 카테고리별 인기 무료 상위 50개씩 총 1200개의 APK 파일을 분석하여 퍼미션 항목을 측정하여 분류한 결과는 다음 [표 3-3], [표 3-4], [표 3-5]에서 카테고리별 주요 권한 리스트를 확인 할 수 있다. 대부분의 애플리케이션의 권한 요청에서 90%이상 인터넷 접근 허가를 요청하는 INTERNET 퍼미션 정보와 네트워크 상태를 확인하는 ACCESS_NETWORK_STATE 퍼미션 정보를 포함하고 있으며, 각 해당 애플리케이션 카테고리별 60~70%의 공통적으로 요청하는 퍼미션 패턴 정보를 바탕으로 카테고리별 특성 퍼미션 권한을 학습 시킨다.



[그림 3-3] 애플리케이션별 카테고리 분류

[표 3-3] 카테고리별 퍼미션 요청 권한(1)

	게임	건강	교육	교통	금융	날씨	뉴스	데코레이션
ACCESS_COARSE_LOCATION	14	26	8	39	17	37	18	17
ACCESS_FINE_LOCATION	6	15	7	38	22	48	47	11
ACCESS_GPS	-	-	-	27	-	-	-	-
ACCESS_LOCATION	-	-	-	28	-	-	-	-
ACCESS_LOCATION_EXTRA_COMMANDS	10	-	-	-	8	-	-	-
ACCESS_NETWORK_STATE	50	50	42	50	45	50		45
ACCESS_WIFI_STATE	36	19	7	27	13	36	14	18
AUTHENTICATE_ACCOUNTS	-	-	-	-	7	8	-	-
CALL_PHONE	-	-	-	9	8	-	9	-
CAMERA	-	8	9	-	14	-	8	4
CHANGE_NETWORK_STATE	22	-	-	-	5	-	-	-
CHANGE_WIFI_STATE	24	-	-	26	24	-	-	-
GET_ACCOUNTS	32	30	5	5	20	20	40	20
GET_TASKS	25	7	-	12	11	12	17	-
INTERNET	50	48	47	49	50	48	50	38
MANAGE_ACCOUNTS	-	-	-	-	7	8	-	-
MODIFY_AUDIO_SETTINGS	-	13	-	-	12	-	-	-
MOUNT_UNMOUNT_FILESYSTEMS	13	8	-	-	-	-	-	-
READ_CALL_LOG	10	-	-	-	-	-	-	5
READ_CONTACTS	15	5	-	-	10	-	-	-
READ_EXTERNAL_STORAGE	45	45	35	30	40	45	45	35
READ_LOGS	8	-	-	-	-	-	-	-
READ_OWNER_DATA	7	-	-	-	-	-	-	-
READ_PHONE_STATE	43	13	8	22	18	23	35	15
READ_SYNC_SETTINGS	-	-	7	-	-	8	-	-
RECEIVE_BOOT_COMPLETED	8	13	5	-	4	42	6	33
RECORD_AUDIO	10	5	13	-	18	-	7	-
RESTART_PACKAGES	7	-	-	8	6	-	-	-
UPDATE_DEVICE_STATS	18	-	-	-	-	-	-	-
USE_CREDENTIALS	13	12	-	-	7	11	17	-
VIBRATE	32	17	8	37	32	27	25	24
WAKE_LOCK	43	23	12	13	27	23	47	16
WRITE_CALL_LOG	-	-	-	-	-	-	5	5
WRITE_EXTERNAL_STORAGE	44	43	32	28	43	43	42	27
WRITE_SETTINGS	-	-	-	6	-	-	-	8

[표 3-4] 카테고리별 퍼미션 요청 권한(2)

	도구	도서	라이브러리	일상	만화	미디어	사진	생산성
ACCESS_COARSE_LOCATION	13	6	-	12	-	15	24	10
ACCESS_FINE_LOCATION	-	-	-	14	-	5	28	10
ACCESS_NETWORK_STATE	43	49	13	48	50	-	48	43
ACCESS_WIFI_STATE	13	32	14	25	10	-	47	22
AUTHENTICATE_ACCOUNTS	-	-	-	5	-	-	-	24
BLUETOOTH	-	-	12	-	-	-	3	6
BLUETOOTH_ADMIN	-	-	8	-	-	-	4	7
BROADCAST_STICKY	16	-	-	-	-	-	-	-
CALL_PHONE	-	-	-	6	-	-	-	5
CAMERA	12	-	-	27	-	-	45	13
CHANGE_NETWORK_STATE	6	6	-	-	-	-	7	-
CHANGE_WIFI_STATE	-	7	-	14	-	-	10	5
DISABLE_KEYGUARD	-	-	-	13	-	-	-	-
FLASHLIGHT	8	-	-	-	-	-	5	-
GET_ACCOUNTS	22	24	7	15	-	-	20	30
GET_TASKS	14	13	-	14	-	-	15	15
INTERNET	48	49	13	38	50	-	49	48
MANAGE_ACCOUNTS	7	-	-	5	5	-	-	32
MOUNT_UNMOUNT_FILESYSTEMS	8	-	-	-	-	-	9	-
NFC	-	-	-	-	-	-	8	-
READ_CALL_LOG	-	-	4	-	-	-	-	13
READ_CALENDAR	-	-	-	-	-	-	6	17
READ_CONTACTS	-	-	5	-	-	-	48	23
READ_EXTERNAL_STORAGE	22	47	16	30	33	-	7	48
READ_LOGS	7	-	7	-	-	-	-	-
READ_PHONE_STATE	4	26	13	17	7	-	33	22
READ_SYNC_SETTINGS	-	7	-	-	-	-	-	21
RECEIVE_BOOT_COMPLETED	-	6	-	14	-	-	12	17
RECORD_AUDIO	12	-	-	-	8	-	-	7
RESTART_PACKAGES	5	-	-	12	-	-	-	8
SYSTEM_ALERT_WINDOW	10	-	-	-	-	-	16	-
USE_CREDENTIALS	7	-	-	6	-	-	-	33
VIBRATE	26	11	-	36	4	-	32	37
WAKE_LOCK	18	23	17	27	6	-	31	34
WRITE_EXTERNAL_STORAGE	23	42	18	32	32	-	13	11
WRITE_SETTINGS	6	7	-	6	-	-	12	17
WRITE_OWNER_DATA	-	5	-	7	-	-	-	23
WRITE_SYNC_SETTINGS	-	6	-	-	-	-	-	13

[표 3-5] 카테고리별 퍼미션 요청 권한(3)

	쇼핑	소셜	여행	음악	스포츠	비즈니스	엔터테인먼트	커뮤니케이션
ACCESS_COARSE_LOCATION	22	32	43	-	26	13	22	30
ACCESS_FINE_LOCATION	21	38	42	7	24	16	14	35
ACCESS_LOCATION_EXTRA_COMMANDS	8	16	13	-	25	13	-	-
ACCESS_NETWORK_STATE	42	43	46	42	46	45	45	50
ACCESS_WIFI_STATE	32	28	27	26	35	47	38	36
AUTHENTICATE_ACCOUNTS	-	13	-	7	7	6	-	17
BLUETOOTH	8	-	-	4	11	8	3	-
BLUETOOTH_ADMIN	-	-	-	8	13	3	9	-
BROADCAST_STICKY	-	12	-	-	7	13	4	-
CALL_PHONE	12	18	6	-	-	17	-	20
CAMERA	28	36	-	-	-	24	7	28
CHANGE_NETWORK_STATE	-	-	-	-	6	12	8	7
CHANGE_WIFI_STATE	5	13	17	-	7	23	-	28
GET_ACCOUNTS	43	37	14	16	23	24	23	37
GET_TASKS	36	22	-	-	8	17	13	23
INTERNET	50	48	48	46	46	47	43	46
KILL_BACKGROUND_PROCESSES	7	7	-	7	8	12	-	8
MANAGE_ACCOUNTS	6	5	-	-	6	9	8	17
MODIFY_AUDIO_SETTINGS	-	6	13	-	-	6	-	-
READ_CALENDAR	-	8	-	-	-	-	-	4
READ_CONTACTS	14	28	12	-	7	15	-	18
READ_EXTERNAL_STORAGE	42	29	43	-	42	45	45	48
READ_LOGS	18	7	7	-	-	7	-	13
READ_OWNER_DATA	-	12	-	-	-	4	-	-
READ_PHONE_STATE	27	37	37	27	-	49	45	37
READ_SYNC_SETTINGS	-	17	-	5	4	8	-	-
RECEIVE_BOOT_COMPLETED	8	-	17	-	8	6	23	-
RECORD_AUDIO	-	22	-	13	-	5	17	23
RESTART_PACKAGES	-	17	-	-	-	10	6	7
RECEIVE_SMS	23	13	8	-	-	-	13	18
SEND_SMS	-	7	6	-	-	-	-	8
USE_CREDENTIALS	22	13		8	13	10	-	34
VIBRATE	48	42	13	11	37	32	23	37
WAKE_LOCK	41	38	24	34	47	29	46	31
WRITE_CALEDAR	-	7	-	-	-	-	-	8
WRITE_CONTACTS	-	7	-	4	-	18	-	17
WRITE_EXTERNAL_STORAGE	43	43	42	39	44	44	41	46
WRITE_SETTINGS	-	7	8	18	10	-	13	14
WRITE_SYNC_SETTINGS	-	-	-	6	6	-	-	12

B. 상황정보 수집

지능형 접근제어 기술에서 요구되는 상황정보의 수집 요구 사항은 다음과 같다.

1. 상황정보 수집 요구사항

① 카테고리의 식별자를 수집할 수 있어야 한다.
- 지능형 접근제어 보안적용은 기본적으로 각 애플리케이션의 권한 정보를 분석하여 차별적으로 보안을 제공하는 기술이므로 애플리케이션을 구분할 수 있는 식별자가 수집되어야 한다.

② 퍼미션 요청 권한 정보를 수집할 수 있어야 한다.
- 카테고리별 특정 이상 권한을 포함 여부를 판단하고, 보안 적용이 요구 되므로 실행 중인 애플리케이션 서비스의 퍼미션 정보를 수집할 수 있어야 한다.

③ 리소스 접근을 요청하는 API 정보를 수집할 수 있어야 한다.
- 정산적인 애플리케이션에서도 개인정보를 요청하는 권한을 요청하는 경우가 발생하기 때문에 리소스에 접근하는 API 정보를 수집할 수 있어야 한다.

④ 단말기의 컴퓨팅파워 정보를 수집할 수 있어야 한다.
- 스마트폰 기기의 종류가 다양하며 각 디바이스별 컴퓨팅능력 및 버전 정보가 다르므로, 각 단말의 성능에 적합한 보안 정책의 적용을 위해 단말기의 정보 수집이 요구된다.

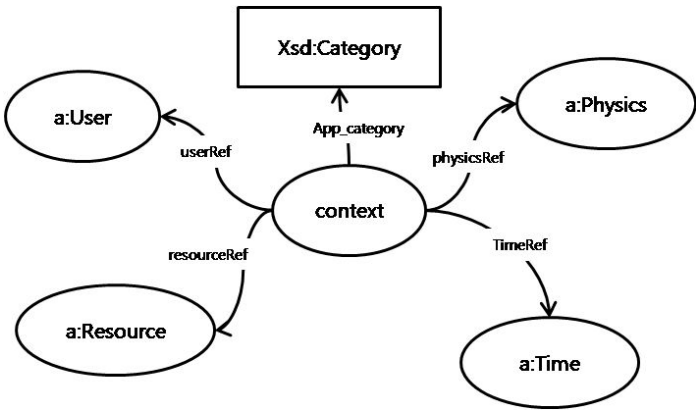
2. 상황정보 데이터

본 논문에서의 상황 클래스는 신원 정보, 물리 정보, 메시지 정보, 자원 정보, 카테고리 정보로 구성한다. 상세한 내용은 다음 [표 3-5]와 같다[7].

[표 3-6] 상황 정보 분류

상황 정보	설 명
신원 상황 정보	사용자 권한
물리 상황 정보	단말기 위치, 단말기 정보 등
시간 상황 정보	접속 시간, 종료 시간, 사용 시간 등
자원 상황 정보	리소스 자원 접근 권한, 저장 매체 정보 등
카테고리 상황 정보	게임, 음악, 교육 정보 등

지능형 접근제어 모델은 정의된 상황 클래스 정보를 이용하여 [표 3-5]와 같이 OWL 정보를 이용하여 신원정보, 물리정보, 메시지 정보, 리소스 정보, 카테고리 정보 등을 정의한다. 정의 된 상황 정보를 분류한 상황 온톨로지의 클래스와 속성으로 표현하면 다음 [그림 3-4]와 같다.



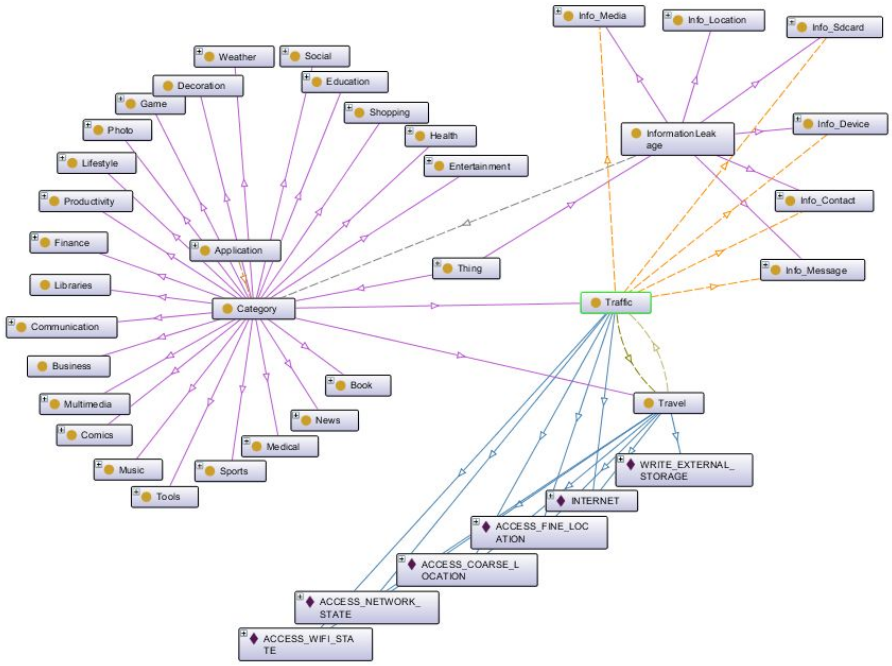
[그림 3-4] 상황 정보 온톨로지 클래스와 속성

C. 지능형 접근제어 정책

본 장에서는 지능형 접근제어를 위한 애플리케이션의 카테고리별 온톨로지를 구축하고 정책을 정의하여 지능형 접근제어 모델을 서술한다.

1. 온톨로지 모델링

다음 [그림 3-5]는 애플리케이션 카테고리별 상황정보 온톨로지 모델링이다. 3.2절에서 애플리케이션의 카테고리별 퍼미션 분석을 통하여 각 카테고리별 주요 특성 퍼미션 요청 권한과 정보 유출에 권한에 대한 관계를 학습하고, Protege을 사용하여 온톨로지 모델링을 구축한다.



[그림 3-5] 애플리케이션 카테고리 별 온톨로지 모델링

앞서 구축한 온톨로지 모델링을 통하여 실행중인 애플리케이션이 요구하는 리소스 권한과 애플리케이션 카테고리 별 요구하는 리소스 접근 권한의 관계를 통하여 이상 접근 권한에 대하여 개인정보 유출하는 클래스와 관계의 추론을 통하여 악성 행위를 일으킬 수 있는 조합을 판단하고, 사용자에게 접근제어모델 통하여 이상 권한 사용 시 사용자에게 접근 제어를 요청한다. 다음 [표 3-7]는 악성코드에서 개인정보를 유출하는 퍼미션 조합을 나타낸다. 이를 온톨로지 추론을 이용하여 신종 악성코드가 발생 시 해당 애플리케이션의 카테고리 별 이상 권한에 대한 추론을 통하여 신종 악성코드를 탐지 할 수 있다.

[표 3-7] 개인정보를 유출하는 퍼미션 조합

분류	권한	설명
Info_Device	INTERNET READ_PHONE_STATE	핸드폰 정보를 읽어와 인터넷을 통하여 전송
Info_Device	SEND_SMS + INTERNET READ_PHONE_STATE	핸드폰 정보를 읽어 SMS, 인터넷을 통하여 전송
Info_Location	INTERNET ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION	위치정보를 읽어서 전송
Info_Location	SEND_SMS ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION	위치정보를 읽어서SMS로 전송
Info_Massage	SEND_SMS READ_PHONE_STATE	핸드폰 정보를 읽어 SMS로 전송
Info_Location	INTERNET + SEND_SMS ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION	위치정보를 읽어와 SMS, 인터넷을 통하여 전송

[표 3-8] 온톨로지 기반 상황 정보 온톨로지 OWL 코드

```
<owl:Class rdf:ID="Context">
        <rdfs:hasUser rdf:resource="#User"/>
        <rdfs:hasPhysics rdf:resource="#Physics"/>
        <rdfs:hasTime rdf:resource="#time"/>
        <rdfs:hasResource rdf:resource="#Resource"/>
        <rdfs:hasCategory rdf:resource="#Category"/>
</owl:Class>

<owl:Class rdf:ID="App_Category">
        <rdfs:subClassOf rdf:resource="#Application"/>
</owl:Class>

        <owl:Class rdf:ID="App_Category">
                <rdfs:subClassOf rdf:resource="#Game"/>
                <rdfs:hasGame rdf:resource="#Permission1"/>
                <rdfs:subClassOf resource="#book"/>
                <rdfs:hasBook rdf:resource="#Permission2"/>
                <rdfs:subClassOf rdf:resource="#Health"/>
                <rdfs:hasHealth rdf:resource="#Permission3"/>
                                        ⋮
                <rdfs:subClassOF rdf:resource="#Travel"/>
                <rdfs:hasTravel rdf:resource="#Permission N"/>

        </owl:Class>

        <owl:Class rdf:ID="InformationLeakage">
                <rdfs:subClassOf rdf:resource="#Info_Sdcard"/>
                <rdfs:subClassOf rdf:resource="#Info_Location"/>
                <rdfs:subClassOf rdf:resource="#Info_Media"/>
                <rdfs:subClassOf rdf:resource="#Info_Contact"/>
                <rdfs:subClassOf rdf:resource="#Info_Message"/>
                <rdfs:subClassOf rdf:resource="#Info_Device"/>
        </owl:Class>

        <owl:ObjectProperty rdf:ID="hasBook">
                <rdfs:domain rdf:resource="#Application"/>
                <rdfs:range rdf:resource="#Rook"/>
        </owl:ObjectProperty>

        <owl:ObjectProperty rdf:ID="User-Create">
                <rdfs:domain rdf:resource="#Application"/>
                <rdfs:range rdf:resource="#Game"/>
        </owl:ObjectProperty>
                                        ⋮
```

Ⅳ. 지능형 접근제어 모델 적용 및 평가

본 장에서는 지능형 접근제어 모델에서의 개인정보 유출 탐지의 정확성을 검증하기 위해, 개인정보 유출 탐지를 위한 접근제어 모델을 구현하여 결과를 확인한다. 지능형 접근제어 모델은 개인정보 유출에 따른 권한 관리 중요성이 요구되고, 각 카테고리별 요청하는 권한과 API 함수의 관계를 설정을 통하여 스마트폰 환경에서 제안된 지능형 접근제어 모델을 적용한다.

A. 지능형 접근제어 모델 적용 환경

개인정보 유출 탐지 시스템은 안드로이드 환경 내의 애플리케이션으로 인한 단말기 정보, 주소록 정보, 문자메시지 정보, 통화기록 정보 등의 개인정보 유출을 방지하기 위한 접근제어를 제공한다. 지능형 접근제어의 수행 능력을 검증하기 위해 본 실험에서는 접근제어 모델의 보안 요구 사항을 카테고리별 상황을 고려하여 [표 4-1]과 같이 정의하여 카테고리별 권한과 상황에 따른 능동적인 접근제어의 수행 결과를 확인한다. 안드로이드 내의 보안 도메인은 주체 객체, 상황 정보가 있으며 각 도메인은 다음과 같이 정의 하였다.

주체 도메인은 개인정보 유출 탐지 시스템의 보안 정책을 정의하기 위하여 사용자의 실행 환경에 따라 카테고리로 구분된다. 주체 도메인은 [표 4-2]와 같다. 예로 들면 대부분의 A라는 카테고리 애플리케이션은 A그룹의 퍼미션 권한을 요청하고 있으며, B라는 카테고리 그룹은 B라는 퍼미션 권한을 제공 받는다.

[표 4-1] 개인정보 유출 접근제어 보안 정책

주체	상황	보안 요구 사항
애플리케이션 카테고리	단말기 정보	1. 단말기 정보 열람 2. 단말기 정보 전송
	메시지 정보	1. 단말기 내부 SMS 메시지 정보 열람 2. SMS 메시지 쓰기 3. SMS 메시지 송신 4. SMS 메시지 수신
	미디어 정보	1. 단말기 미디어 정보 열람 2. 단말기 미디어 정보 전송 3. 단말기 미디어 정보 기록
	통화 정보	1. 단말기 통화 상태 열람 2. 단말기 주소록 정보 기록 3. 단말기 주소록 정보 연락
	위치 정보	1. 단말기 위치 정보 열람 2. 단말기 위치 정보 보내기
	저장매체 정보	1. 외부 저장 매체 쓰기 2. 외부 저장 매체 읽기

[표 4-2] 주체 도메인 정의

주체	ID	주체	ID
게임	pak_game	비즈니스	pak_business
건강	pak_health	사진	pak_photo
교육	pak_education	생산성	pak_productivity
교통	pak_traffic	소셜	pak_social
날씨	pak_weather	스포츠	pak_sports
뉴스	pak_news	엔터테인먼트	pak_entertainment
데코레이션	pak_decoration	여행	pak_travel
도구	pak_tools	음악	pak_music
도서	pak_book	미디어	pak_medical
라이브러리	pak_libraries	커뮤니케이션	pak_communication
라이프스타일	pak_lifestyle	금융	pak_finance
만화	pak_comics	기타	pak_etcetera
멀티미디어	pak_multimedia		

권한 도메인은 주체들에서 발생될 수 있는 개인정보 유출과 관련된 퍼미션 권한과 상황에 따라 발생할 수 있는 객체로 정의하였다. 개인정보 유출 관련 객체는 “메시지 정보”, “주소록 정보”, “단말기 정보”, “통화기록 정보”, “이미지 정보”에서 발생 할 수 있는 도메인 권한은 [표 4-3]과 같다.

[표 4-3] 도메인 권한

Permission	ID
INTERNET	Per_Int1
ACCESS_NETWORK_STATE	Per_Int2
ACCESS_WIFI_STATE	Per_Int3
CHANGE_WIFI_STATE	Per_Int4
READ_SMS	Per_Sms1
WRITE_SMS	Per_Sms2
SEND_SMS	Per_Sms3
RECEIVE_SMS	Per_Sms4
READ_PHONE_STATE	Per_Call1
CALL_PHONE	Per_Call2
READ_CONTACTS	Per_Call3
WRITE_CONTACTS	Per_Call4
READ_EXTERNAL_STORAGE	Per_card1
WRITE_EXTERNAL_STORAGE	Per_card2
ACCESS_COARSE_LOCATION	Per_loc1
ACCESS_FINE_LOCATION	Per_loc2
RECEIVE_BOOT_COMPLETED	Per_dev1
RESTART_PACKAGES	Per_restart
INSTALL_PACKAGES	Per_install
WRITE_APN_SETTINGS	Per_apn
WAKE_LOCK	Per_lock
DISABLE_KEYGUARD	Per_key
GET_TASKS	Per_task
READ_LOGS	Per_log
READ_HISTORY_BOOKMARKS	Per_bookmark1
WRITE_HISTORY_BOOKMARKS	Per_bookmark2
SET_WALLPAPER	Per_wallpaper
MOUNT_UNMOUNT_FILESYSTEMS	Per_file
INSTALL_SHORTCUT	Per_short

상황 정보 도메인은 애플리케이션의 카테고리별 권한 사용에 따른 개인정보 유출 상황을 정의하였다. 또한 카테고리별 접근제어 수행 결과를 알아보기 위해 개인정보가 유출되는 상황을 정의하여 단말기에서 카테고리별 개인정보가 유출되는 상황에 따라 탐지하고 접근제어를 통하여 개인정보 유출을 보호하는 결과를 얻을 수 있도록 하였다. 다음 [표 4-4]은 상황 정보 도메인에 대한 정의이다.

[표 4-4] 상황정보 도메인 정의

상황 정보	ContextInformation
단말기 정보 접근	Info_Device
메시지 정보 접근	Info_Message
주소록 정보 접근	Info_Contact
미디어 정보 접근	Info_Media
위치 정보 접근	Info_Location
저장매체 정보 접근	Info_Sdcard

객체 도메인은 지능형 접근제어 모델의 수행 객체들로 서비스를 제공하기 위해 개인 정보를 정보 저장소에서 개인정보를 호출하는 객체, 개인정보를 외부로 유출하는 객체들로 구성된다. 객체 도메인은 다음[표 4-5]과 같이 정의하였다.

[표 4-5] 객체 도메인 정의

상황정보 분류	함수 명	Object_ID
단말기 정보	getLine1Number	Contact_Number
	getSubscriberId	Device_Id
	getDeviceId	Device_DeviceId
	getNetworkOperator	Network_Operator
	getNetworkOperatorName	Network_OperatorName
	getSimSerialNumber	Device_SIM
	getPackageManager	App_Pack
	getInstalledApplications	App_Install
네트워크 정보	isWifiEnabled, getWifiState	Info_Wifi_A
	setWifiEnabled, setWifiState	Info_Wifi_B
	url.openConnection	Url.openConnection
	openStream, InputStream.read	Network_file
	getNetworkInfo	Info_Network
	HttpURLConnection	URL_Connection
SMS 정보	sms.all.CONTENT_URI	Sms_all
	sms.inbox.CONTENT_URI	Sms_inbox
	sms.sent.CONTENT_URI	Sms_sent
	sms.draft.CONTENT_URI	Sms_draft
	sms.outbox.CONTENT_URI	Sms_outbox
	sendTextMessage	Sms_send
	getMessageBody	Sms_Body
위치정보	getLatitude. getLongitude	User_location
	getFromLocation getFromLocationName	

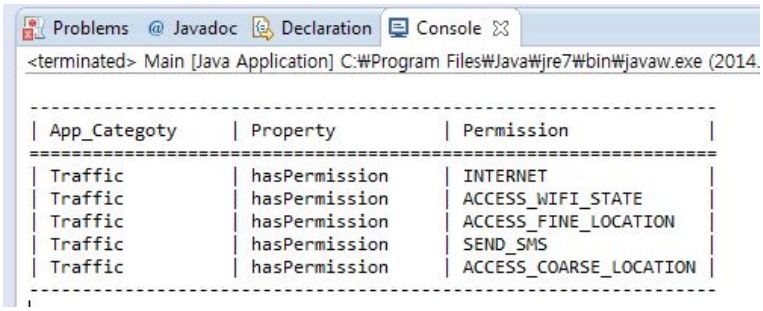
B. 실험 환경 및 실험 시나리오 평가

본 절에서는 지능형 접근제어 모델을 상황정보에 따른 온톨로지 추론을 통해 동적 접근제어 서비스가 가능하도록 정보유출 행위의 리소스 접근을 탐지하기 위하여 시나리오를 통해 수행성을 평가하고, 검증하고자 한다. 본 실험은 [표 4-6]와 같은 환경에서 다음과 같은 방법으로 진행하였다. 첫 번째로 구글 마켓의 애플리케이션 카테고리별 무료 애플리케이션 상위 50개의 데이터의 분석하고, Protege를 이용하여 카테고리별 리소스 접근 권한 요청에 대한 관계를 설정하고, 카테고리별 온톨로지 모델링을 한다. 두 번째로 4.1절에서의 정의한 보안정책 요소 및 상황정보 요소를 기준으로 추론 규칙을 정의하고, Jena 엔진을 이용하여 결과를 검증한다.

[표 4-6] 실험 환경

구분	설명
테스트 환경	Intel(R) Core(TM) i7-3770 CPU@3.40GHz, RAM : 4GB
OS	Android 4.1.2(Jelly Bean)
개발 언어	Protege 4.3 / Java / Python 2.7

첫 번째 시나리오는 제안된 지능형 접근제어를 수행하기에 앞서 올바른 관계 설정이 되었는지 알아보기 위한 것이다. 이를 실험을 통하여 결과를 확인하기 위하여 “교통” 관련 애플리케이션을 대상으로 관련 애플리케이션에서 주요 사용하는 인터넷 상태 조회, 인터넷 엑세스, Wifi 엑세스, GPS 정보를 요청 받을 수 있는 권한과 앞서 요청하는 권한들과의 조합을 통하여 개인정보를 SMS로 유출될 수 있는 SEND_SMS 퍼미션 권한을 포함한 5개의 퍼미션 정보를 가지고 있는 애플리케이션을 통하여 실험을 진행하였다. 애플리케이션의 카테고리 중 하나인 “교통” 관련 애플리케이션이 실행 중일 때, 해당 온톨로지 모델링에서의 관계 설정을 통하여 “교통” 관련 애플리케이션이 요청하고 있는 리소스 권한 정보를 다음 [그림 4-1]의 결과를 통하여 확인한다. 해당 애플리케이션 결과는 추론 규칙을 적용하지 않았기 때문에 다음과 같은 애플리케이션의 권한 요청 정보를 출력 하는 것을 확인할 수 있다.



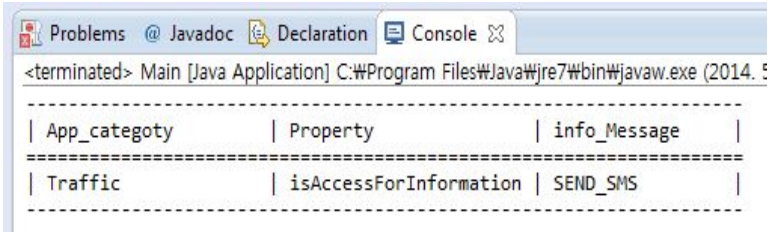
[그림 4-1] 추론 규칙 적용 전 결과

두 번째 시나리오는 정보유출 탐지를 위한 온톨로지 기반 접근제어 모델의 상황 정보 요소를 추가하여 상황 정보 온톨로지 추론을 통한 악성행위의 탐지와 동적인 접근 제어의 정확성을 평가하기 위한 것이다. 시나리오1에서 “교통” 관련 애플리케이션의 관계 설정을 통하여 얻은 리소스 접근 정보를 [표 4-6]의 추론 규칙을 통하여 실행 중인 애플리케이션에서 요청하고 있는 권한이 개인정보 관련 리소스 접근 요소와 관계를 추론한다.

```
[InformationleakageForTrafficeRule:
(?Application http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#
hasCategory ?App_Category )
(?App_Category http://www.w3.org/1999/02/22-rdf-syntax-ns#type http://www.semanticweb.org/smart/onto-
logies/2014/4/smart.owl#Per_Traffic)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#hasPermissionInfo ?Per_Traffic)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#AccessContext ?Info_Device)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#AccessContext ?Info_Message)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#AccessContext ?Info_Contact)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#AccessContext ?Info_Media)
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#AccessContext ?Info_Sdcard)
->
(?Traffic http://www.semanticweb.org/smart/ontologies/2014/4/smart.owl#isAccessForInformation
?ContentInformation)]
```

[표 4-6] 교통 관련 애플리케이션에서 추론 규칙

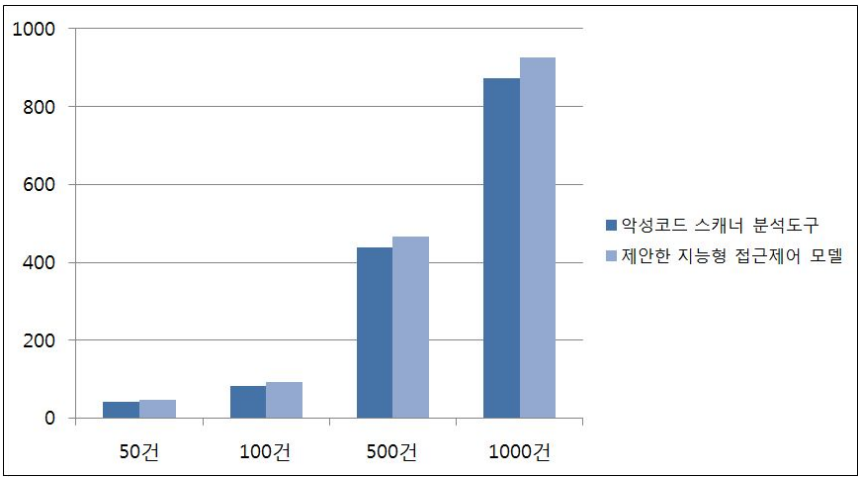
실험을 통하여 “교통” 관련 애플리케이션에서 발생할 수 있는 정보유출 상황 정보에 대한 Jena추론 엔진을 이용하여 [표 4-6]의 추론 규칙을 적용한 결과는 다음 [그림 4-2]와 같다.



[그림 4-2] 추론 규칙 적용 후 결과

해당 교통 관련 애플리케이션의 특정 권한 요소에는 인터넷 엑세스를 허가하는 INTERNET 권한과 GPS 정보를 얻을 수 있는 ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION의 권한은 포함하고 있지만 SEND_SMS 권한을 요청함으로써 GPS 정보를 획득하여 SMS를 통해 전송할 수 있는 위치정보 유출 행위가 발생 한다. 이를 추론을 통하여 카테고리별 주요 특성 권한 이상의 리소스 접근 정보를 탐지하고 있음을 보인다.

다음 [그림 4-3]은 악성 APK파일의 탐지 건수를 비교 분석한 결과이다. 그림을 살펴보면 실험에서 사용된 지능형 접근제어 기법의 탐지율이 더 높은 것을 확인할 수 있다. 이를 통하여 악성코드를 분석하고, 탐지하여 개인정보 유출 리소스 접근을 효과적으로 수행할 수 있음을 알 수 있다. 이는 더욱 세분화된 퍼미션 권한과 API 함수의 관계 설정을 구축한다면 신종 악성행위 발생 시에도 정확성이 높은 탐지를 할 수 있음을 나타낸다.



[그림 4-3] 악성코드 탐지 비교

[표 4-7]은 악성코드 탐지 결과에 대한 도표이다. 도표를 확인해보면 악성코드 샘플을 이용한 실험을 통해 기존의 사용되고 있는 기법들과 제안된 기법의 탐지율을 비교하여 악성코드 탐지율의 우수성이 높은 것을 확인 할 수 있다.[52,54]

[표 4-7] 악성코드 탐지 결과 비교표

구분	악성코드 스캐너 분석도구[54]	비율	제안한 지능형 접근제어 기법	비율
50	42	74%	46	92%
100	83	83%	93	93%
500	437	87.4%	467	93.4%
1000	873	87.3%	927	92.7%

(단위 : 건)

제안한 기법을 기존의 접근제어 모델과 비교하면 다음[표 4-7]과 같다. 제안한 상황정보 추론을 통하여 악성행위를 유발할 수 있는 악성코드의 탐지의 정확성을 높이고, 상황정보에 따른 능동적인 접근제어가 가능하다.

[표 4-7] 접근제어 모델 비교[7]

구분	임의접근제어 모델(DAC)	강제접근제어 모델(MAC)	제안한 상황 추론 기반 접근제어 모델
상황 인식	불가	불가	적용
보안 등급	미적용	적용	적용
접근 모드	불가	설정	설정
역할 상속	모든 권한	부분적인 권한	부분적인 권한
접근제어 능력	수동	수동,능동적	수동, 능동적
정책관리 편의성	낮음	낮음	높음
정책관리 효율성	낮음(정적)	낮음(정적)	높음(동적)
위임 주체	사용자	관리자, 사용자	사용자
위임 거부	가능	가능	가능
부분 권한 위임	불가	부분적 권한 영역설정	부분적인 권한 영역설정

제안한 지능형 접근제모 모델은 상황정보를 추가하여 정보유출 행위를 탐지를 수행한 결과 카테고리별 이상 권한의 사용을 상황정보 온톨로지 추론을 통해 탐지하였고, 악성코드 탐지율의 정확성을 높였다. 또한 카테고리별 애플리케이션 역할에 따라 역할에 부여된 접근 권한에 대한 평가를 올바르게 수행함을 확인하였다. 이를 통해 기존의 기법들과의 비교를 통하여 제안한 지능형 접근 제어모델의 우수성을 확인하였다.

V. 결론 및 제언

스마트폰 디바이스의 발전으로 인하여 데스크톱에서 이용하던 웹 서비스를 스마트폰을 통하여 사용할 수 있게 되었으며, 스마트폰의 사용량이 증가하게 됨에 따라 스마트폰의 내부 개인정보 보안의 중요성이 강조되고 있다. 특히 안드로이드 플랫폼의 개방성 특징으로 인해 애플리케이션을 손쉽게 개발하고 배포하기 때문에 악성코드의 유형과 수는 기하급수적으로 증가하고 있으며, 보안 사례도 급증하고 있다. 이러한 상황에서 스마트폰 내부의 개인정보를 효율적으로 관리하기 위하여 개인정보를 유출하는 행위를 탐지하고 차단하기 위한 새로운 방법이 요구되고 있다.

본 논문에서는 스마트폰의 개인정보를 유출하는 악성코드 특징과 행위에 대하여 알아보고, 개인 정보의 유출 행위를 탐지하고, 정보 자원에 대한 접근제어를 하기 위한 지능형 접근제어 모델을 제안하였다. 제안한 방법은 정보 자원에 대한 접근 권한을 역할에 할당하고, 애플리케이션의 종류에 따른 카테고리 분류를 통하여 카테고리별 요구하는 권한 특성을 바탕으로 해당 역할을 배정함으로써, 정당한 애플리케이션이 유효한 접근권한을 통해 정보 자원을 이용할 수 있는 서비스를 제공한다. 또한 정보 자원을 사용하기 위해 애플리케이션이 해당 역할에 배정되어 접근권한을 획득하고 있더라도 현재 상황 역할이 보안정책에 만족하는지 확인하여 획득한 접근 권한의 유효성을 판단하는 지능형 접근제어를 수행한다. 이를 통하여 기존의 악성코드와 알려지지 않는 신종 악성코드를 탐지하는데 있어 기존의 탐지 기법에 비해 정확성을 높이고, 개인정보의 접근에 대한 접근제어를 통해 개인정보를 보호할 수 있었다. 따라서 향후 연구로는 새로운 유형의 개인정보 유출 행위를 탐지하기 위하여 추가적으로 APK 파일 데이터를 이용하여 카테고리별 특징을 세부적으로 분석하고, 온톨로지 모델링하여 리소스 접근에 따른 여러 단계의 복합적인 검사 절차를 통해서 정확성을 높이고, 악성행위에 대한 탐지를 하여 능동적인 접근제어를 할 수 있도록 개선해야 할 것이다.

참고문헌

[1] Gruber T., "A Translation Approach to Portable Ontology Specification", in Knowledge Acquisition Journal, Vol. 5 No. 2, pp. 192-220, 1993.

[2] Pedo Carvalho de Oliveira, "Probabilistic Reasoning in the Semantic Web using Markov Logic", MSc Thesis, July 2009.

[3] M.J.Moyer and M. Ahamad, "Generalized Role-Based Access Control", IEEE International Conference on Distributed Computing Systems(ICDCS2001), pp. 391-398, 2001.

[4] D.F.Ferraiolo,J.A.Cuginiand D.R.Kuhn,"Role-Based Access Control(RBAC): Features and Motivations", 11th Annual Computer Security Application confernece, November 1995.

[5] R. S. Sand huand E.J.Coyne, "Role-Based Access Control Models", IEEE Computer, 20(2), pp. 38-47, February 1996.

[6] R.S.Sandhu,D.Ferraiolo and R.Kuhn,"The NIST Modelfor Role-Based Access Control : Towards a Unified ModelApproach", 5thACM Workshop on RBAC, August 2000.

[7] Chang Choi, Junho Choi, Pankoo Kim, "Ontology-based access control model for security policy reasoning in cloud computing", The Journal of Super computing, Vol. 67, No. 3, pp. 711-722, March 2014.

[8] Jae-Man You, In-Kyoo Park,"Android Storage Access Control for Personal Information Security", The Journal of The Institute of Internet, Broadcasting and Communication, VOl. 13, No. 6, December 2013.

[9] Soon-Seok Kwan and Young-Chan Kim, "A Studyon the Android Security Kernel Module based on Mandatory Access Control", The Korean Institute of Communications and Onformation Sciences, 2010.

[10] Clint Gibler, Honathan Crussell, Jeremy Erickson,HaoChen,"AndroidLeaks: Detecting Privacy Leaks In Android Applications", 2011.8.

[11] M. Nauman, S. Khan, and X. Zhang. "Apex: extending android

permissionmodel and enforcement with user-defined runtime constraints. in Proceedings of the 5th ACM Symposium on Information", Computer and Communications Security, 2010.

[12] Yan Wang, Hongli Liu, Jun Feng, "The Design of an Intelligent Security Access Control System Based on Fingerprint Sensor FPC1011C", Circuits and Systems, 2010.1.

[13] Eun-Sook Cho and Chul-jin Kim, "A Technique of Applying Ontology for Service Customization of Android", Journal of the korea Academia-Industrial cooperation Society, Vol. 13, No. 6, pp. 2707-2712, 2012.

[14] Tsung-Yi Chen, "Knowledge sharing in virual enterprises via an ontology-based access contorl approach", Computer in Industry 59, pp. 502-519, 2008.

[15] Haibo Shen, Yu Cheng, "A Semantic Context-Based Model for Mobile Web Services Access Control", Computer Network and Information Security, pp. 18-25, 2011.

[16] Mohamed Bourimi, Simon Scerri, "A two-level approach to ontology-based access control in pervasive personal servers", Scientific research paper (ger. Wissenschaftlicher Artikel), 2011.

[17] Enck, W., Octeau, D., McDaniel, P and Chaudhuri, S., "A Study of Android Application Security," USENIX Security, 2011.

[18] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., and Altmann, J., "Context-awareness on mobile devices - the hydrogen approach", In Proceddings of the 36th Annual Hawaii International Conference on System Sciences, pp. 292-302, 2002.

[19] Hong, C-S., Cho, J., Lee, K-W., Suh, Y-H., Kim, H., and Lee, H-C., "Developing Context-Aware System for Providing Intelligent Robot Services", In Proceeding of European Conference on Smart Sensing and Context, LNCS 4272, pp. 174-189, 2006.

[20] 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.

[21] 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.

[22] Y. Park, C. Lee, C. Lee, J. Lim, S. Han, M. Park and S. Cho, "RGBDroid: A Novel Response-Based Approach to Android Privilege Escalation Attacks," USENIX LEET, Apr 2012.

[23] Vineeth Kashyap and Ben Hardekopf, Security Signature Inference for JavaScript-based Browser Addons," in Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization (CGO '14), February 2014.

[24] David Barrera., H. Gunes Kayacik, "A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android",Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, pp. 274-277, 2010.

[25] P.P.F. Chan, L.C.K. Hui, and S.M. Yiu, Editors. "DroidChecker: Analyzing Android Applications for Capbility Leak." Proceedings of the 5th ACM conference on Security and Privacy in Wireless and Mobile Networks, (2012) Apr 16-18; Arizona, USA.

[26] D. F. Ferraiolo, J. A. Cugini and D. R .Kuhn, "Role-Based Access Control (RBAC) : Features and Motivations",11th Annual Computer Security Application Conference, November 1995.

[27] Y. Zhong, H. Yamaki, H. Takakura, “A Malware Classification method Based on Similarity of Function Structure,” 12th International Symposium of Applications and the Internet(SAINT), pp. 256-261, 2012.

[28] L.Tenenboim-Chekina, O. Barad, A. Shabtai, D., “Detecting Application Update Attack on Mobile Devices through Network Features”, 2013.

[29] 박수환,“유비쿼드터 환경에 적합한 상황인식 기반동적 접근제어 시스템“, 강원대학교대 석사학위논문, 2009.2.

[30] 권오현, 박준석, 염근혁, “모바일 상황 인식 서비스 추론을 위한 온톨로지 모델 선택 기법”, 정보과학회논문지, Vol. 39, No. 8, pp. 646-654, 2012.8.

[31] 윤진식, “정적 분석을 통한 안드로이드 기반 스마트폰의 악성코드 탐지 기법”, 한국해양대학교 석사학위논문, 2011.2.

[32] 최진혁, 박준석, “개인정보 유출 최소화를 위한 통합 접근통제 모델 연구”, 한국공안행정학회보, Vol. 38, No. 0, pp. 317-357, 2010.

[33] 전철, 장준혁, 김봉재, 정진만, 조유근, “효율적인 안드로이드 애플리케이션 검수를 위한 견고한 퍼미션 기반 악성 애플리케이션 여과 기법”, 보안공학연구논문지, Vol. 10, No. 2, pp. 184-189, 2013.4.

[34] 박익수, 오병균, “개인정보시스템에서 접근제어 모델 설계”, 한국컴퓨터종합학술대회 논문집, Vol. 34, No. 1, pp .76-79, 2007.6.

[35] 조병철, “역할기반 접근제어에서 ARBAC97의 역할관리 기법을 적용한 사용자 수준의 위임”, 서강대학교 대학원 석사학위논문, 2000.

[36] 이기철, 이지형, “Jess를 이용한 OWL과 SWRL 통합추론에 관한 연구″, 한국퍼지 및 지능시스템학회 2005년도 추계학술대회 학술발표논문집, Vol. 15, No. 2, pp. 213-216, 2005.11.

[37] 김도래, 박용수, “난독화된 안드로이드 악성 앱에서의 효과적인 악성 연관 API 호출 탐지 및 분석“, 한국정보과학회 추계학술발표회, pp. 790-792, 2013.

[38] 김상일, 김화성, “스마트폰 기반의 상황 추론을 위한 온톨로지 모델링”,한국정보과학회 2013 한국컴퓨터종합학술대회 논문집, pp. 432-433, 2013.6.

[39] 김영동, 김태현, “안드로이드 권한과 브로드캐스트 인텐트 매커니즘의 사용 현황 및 보안 취약성 분석”, 정보보호학회논문지, Vol. 22, No. 5, pp. 1145-1157, 2012.10.

[40] 함유정, 이형우, “안드로이드 모바일 정상 및 악성 앱 시스템 콜 이벤트 패턴 분석을 통한 유사도 추출기법”, 인터넷정보학회논문지, Vol. 14, No. 6, pp. 125-139, 2013.12.

[41] 정윤식, 작영웅, “안드로이드에서 개인정보 유출을 방지하기 위한 접근제어 및 디렉토리명 사상 기법”, 정보과학회논문지, Vol. 39, No. 6, pp.366-372, 2013.12.

[42] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/

[43] Anh-Duy Vu, “Simple Mandatory Access Control for Android(SMACA)“, 국민대학교 석사학위논문, 2011.

[44] StatCounter, “Global Top 8 Mobile Operating Systems″, 2014.

[45] 윤여욱, “코드 디핑 기법을 이용한 안드로이드 악성코드 분석 방법론”, 전남대학교 석사학위논문, 2013.2.
[46] 류윤지, 김바울, 김상욱, “사용자 상황인지 서비스를 위한 안드로이드 기반 모바일 플랫폼”, 한국멀티미디어학회 추계학술발표대회 논문집, Vol. 12, No. 2, pp. 156-158, 2009.11.
[47] 윤재성, 강휘강, “행위기반의 프로파일링 기법을 활용한 모바일 악성코드 분류기법”, 정보보호학회논문지, Vol. 24, No. 1, pp. 145-154, 2014.2.
[48] F-secure, “Q1 2014 Mobile Threat Report", April 29 2014
[49] AhnLab,, “2014년 1분기 스마트폰 악성코드 동향 발표"
[50] AhnLab, "2013년 모바일 악성코드 동향“
[51] http://developer.android.com/reference/pakages.html
[52] http://rogunix.com/docs/Android/Malware
[53] SandDroid, "http://sanddroid.xjtu.edu.cn/"
[54] https://www.virustotal.com/
[55] http://gs.statcounter.com/#mobile_os-ww-monthly-201204-201404
[56] http://jena.apache.org/documentation/inference/index.html
[57] 임형신, 황윤영, 엄동명, “OWL 기반의 온톨로지 도구 비교분석”, 한국한의학연구원논문집, Vol. 12, No. 1, pp. 4-12, 2006.4.
[58] 최창, “시공간 관계 온톨로지 구축을 통한 객체의 움직임 이해에 관한 연구”, 조선대학교 박사학위논문, 2012.4.