JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 26, No. 6: 907~913, Jun. 2022

카테고리와 권한을 이용한 안드로이드 악성 앱 탐지

박종찬[1] · 백남균[2*]

The Detection of Android Malicious Apps Using Categories and Permissions

Jong-Chan Park[1] · Namkyun Baik[2*]

[1]Student, Information Security, Busan University of Foreign Studies, Busan, 46234 Republic of Korea
[2*]Professor, Information Security, Busan University of Foreign Studies, Busan, 46234 Republic of Korea

요 약

전 세계 스마트폰 이용자 중 약 70%가 안드로이드 운영체제 기반 스마트폰을 사용하고 있으며 이러한 안드로이드 플랫폼을 표적으로 한 악성 앱이 지속적으로 증가하고 있다. 구글은 증가하는 안드로이드 대상 악성코드에 대응하기 위해 'Google Play Protect'를 제공하여 악성 앱이 스마트폰에 설치되는 것을 방지하고 있으나, 아직도 많은 악성 앱들이 정상 앱처럼 위장하여 구글 플레이스토어에 등록되어 선량한 일반 사용자의 스마트폰을 위협하고 있다. 하지만 일반 사용자가 악성 앱을 점검하기에는 상당한 전문성이 필요하기에 대부분 사용자는 안티바이러스 프로그램에 의존하여 악성 앱을 탐지하고 있다. 이에 본 논문에서는 앱에서 쉽게 확인이 가능한 카테고리와 권한만을 활용하여 앱의 불필요한 악성 권한을 분류하고 분류한 권한을 통해 악성 앱을 쉽게 검출할 수 있는 방법을 제안한다. 제안된 방법은 '상용 악성 앱 검출 프로그램'과 미탐율·오탐율 측면에서 비교 분석하여 성능 수준을 제시하고 있다.

ABSTRACT

Approximately 70% of smartphone users around the world use Android operating system-based smartphones, and malicious apps targeting these Android platforms are constantly increasing. Google has provided "Google Play Protect" to respond to the increasing number of Android targeted malware, preventing malicious apps from being installed on smartphones, but many malicious apps are still normal. It threatens the smartphones of ordinary users registered in the Google Play store by disguising themselves as apps. However, most people rely on antivirus programs to detect malicious apps because the average user needs a great deal of expertise to check for malicious apps. Therefore, in this paper, we propose a method to classify unnecessary malicious permissions of apps by using only the categories and permissions that can be easily confirmed by the app, and to easily detect malicious apps through the classified permissions. The proposed method is compared and analyzed from the viewpoint of undiscovered rate and false positives with the "commercial malicious application detection program", and the performance level is presented.

키워드 : 안드로이드, 악성 어플리케이션, 권한, 스마트폰

Keywords : Android, malicous apps, permissions, smartphone, categorization

Received 25 April 2022, **Revised** 20 May 2022, **Accepted** 24 May 2022

*** Corresponding Author** Namkyun Baik(E-mail: namkyun@bufs.ac.kr, Tel: +82-51-509-6136)
Professor, Information Security, Busan University of Foreign Studies, Busan, 46234 Republic of Korea

Open Access http://doi.org/10.6109/jkiice.2022.26.6.907 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

글로벌 시장조사 업체 ‘statcounter’에서 예측한 통계에 따르면 2020년 12월부터 2021년 12월까지 스마트폰 이용자 중 약 70% 이상의 스마트폰 사용자가 안드로이드 OS를 통해 스마트폰을 사용하고 있는 것[1]을 확인할 수 있다. 많은 사용자를 보유한 안드로이드 OS는 악의적인 해커들의 공격 대상이 되어 이를 위협하는 악성 앱 또한 꾸준히 증가하고 있다. 그 결과 2020년부터 2021년까지 약 100만건 이상의 악성코드가 발견되는 등 안드로이드 대상 악성 앱은 꾸준히 증가[2]하고 있다.

증가하는 악성 앱에 대응하기 위해 구글은 ‘Google Play Protect’ 서비스를 제공하여 악성 앱이 구글 플레이 스토어에 등록 되는 것을 막아 사용자 스마트폰에 악성 앱이 설치 및 악성 활동 하는 것을 원칙적으로 방지하고 있다.[3] 하지만 많은 악성 앱이 ‘Google Play Protect’의 감시망을 우회하여 구글 플레이스토어에 등록 후 사용자 핸드폰으로 침투하고 있다.[4]

본 논문은 안드로이드 앱의 카테고리를 분류한 후 권한을 이용하여 기존의 방식보다 더 간단하고 명확한 악성 앱 검출 방법을 제안한다. 이를 증명하기 위해 구글 플레이 스토어에서 규정하고 있는 여러 카테고리 중 임의의 카테고리 ‘게임 카테고리’를 선택하여 본 논문에서 제안하는 방법으로 악성 앱을 판별할 수 있는지 검증한다. 실험 대상은 게임 카테고리에 등록 된 인기 앱 182개와 카테고리와 상관 없이 2개년(2019년, 2020년)의 대표 악성 앱 353개(160개,193개)[5][6]의 앱을 선정하여 악성 앱에서 자주 보이지만 게임 앱에는 보이지 않는 권한을 선별하여 악성 앱으로 판별할 수 있는지 확인한다.

Ⅰ장에서는 본 논문의 작성배경을 소개한다. Ⅱ장에서는 악성 앱 탐지를 위해 사용되는 파라미터 ‘카테고리’, ‘권한’에 관하여 정의하고 특히 악성 앱 탐지 방법에 대해 설명한다. Ⅲ장에서는 APK 파일의 카테고리 확인 방법과 APK파일 내에 존재하는 권한 추출 방법, 악성 권한을 산정하는 공식에 대한 방법을 설명한다. Ⅳ장에서는 논문에서 사용한 방법을 사용자 환경에서 적용할 수 있는 방법과 보완이 필요한 사항 에 대해 소개하고 논문의 결론을 정리한다.

Ⅱ. 관련연구

2.1. 카테고리

악성 앱 검사에 카테고리 분류를 사용하는 이유는 정상 앱이 작동하기 위해 필요한 권한은 카테고리마다 특색이 존재하기 때문이다. 만약 ‘사진 카테고리 앱’에서는 카메라 권한이 필수이지만 ‘날씨 카테고리 앱’에서는 카메라 권한이 필요하지 않은 권한일 수 있기에 앱의 특수성을 고려하기 위해 카테고리를 분류한다.

카테고리(Category)란 모바일 어플리케이션을 특징에 따라 분류하는 방식으로 분류의 기준은 ‘구글 플레이 스토어 웹사이트’(https://play.google.com/store/app)의 분류기준을 따른다. 특히, 분류 된 목록 중 중복되는 항목은 하나의 대표 항목으로 통일한다. 통일 되는 항목은 ‘표 1’에서 분류한 17개의 카테고리는 게임 카테고리로 하나로 통합하며, ‘표 2’에서 분류한 3개의 카테고리는 키즈 카테고리로 통합해 최종적으로 ‘표 3’과 같이 총 38개의 카테고리로 분류한다.

Table. 1 Game app category

Education	Sport	Adventure
Word	simulation	Music
Role-playing	Arcade	Car racing
Board	Action	Strategy
Card	Casual game	Puzzle
Casino	Quiz	

Table. 2 Kids app category

Under 5 years old	6-8 years old	9-12 years old

Table. 3 Final category

Daydream	Date	Cartoon
Health/Exercise	Tools	Personalization
Education	Books/References	Game
Finance	Kids	Beauty
Weather	Libraries/Demos	Business
News/Magazine	Life Style	Picture
Productivity	Food and beverage	Event
Social	Entertainment	Car
Childbirth/Parenting	Travel and local information	Augmented reality
Sport	Art/Design	Map/Navigation
Clock app	Music/Audio	Shopping
Watch face	Medical treatment	Communication
Video player/Editor	Real Estate/Home Terrier	

2.2. 권한

권한이란 안드로이드 OS 내 개인정보 보호를 위해 존재하는 기능이다. 모바일 앱은 샌드박스[7]로 구성되어 있으며 샌드박스 외부에 존재하는 제한된 데이터(갤러리 사진, 사용자의 연락처 정보 등 데이터 획득) 및 제한된 작업(오디오 녹음, 카메라 등 기능 사용)에 접근하기 위해 선행되어야할 작업이다.[8]

권한은 APK(Android aPplication pacKage) 파일 최상단 AndroidManifest.xml 파일에 저장 되어 있으며 AndroidManifest.xml 파일 내 <uses-permission> 요소를 포함하여 표현되고 있어야 한다. 예를 들어 카메라 기능 권한(android.permission.CAMERA)을 획득하기 위해서는 '그림 1'과 같이 AndroManifest.xml 파일내에 카메라 접근 권한을 선언해야한다.

```
<manifest ...>
    <uses-permission android:name="android.permission.CAMERA"/>
    <application ...>
        ...
    </application>
</manifest>
```

Fig. 1 'AndroidManifest.xml' permissions sector

본 제안에서 앱 접근 권한을 악성 앱 검사에 사용하는 이유는 두 가지이다. 첫 번째로는 APK파일 최상단 경로에 위치한 AndroidManifest.xml 파일을 통해 권한을 확인할 수 있어 선행지식 없이 누구나 추출하기 쉽다는 점이다. 두 번째로는 악성 앱을 이용하여 피해자로부터 가치 있는 데이터를 획득 및 전송하기 위해서 악성 앱이 다른 앱으로 접근할 수 있는 권한은 필수이므로 악성 앱 탐지 파라미터로 권한을 선택하였다.

2.3. 관련연구

권한을 사용하여 악성 앱과 정상 앱을 분류하는 방법은 이전에도 많은 연구가 진행되었으며 크게 세 가지 방법으로 분류할 수 있다.

첫 번째 방법으로는 카테고리를 분류 후 정상 앱의 평균 권한의 개수를 계산하여 평균보다 과도하게 권한의 개수가 많을 경우 악성 앱으로 의심하는 방법[9]이 있다. 권한의 개수로만 판별하는 방법은 간단한 방법으로 악성 앱을 탐지할 수 있는 장점이 있다. 다만 악성 앱의 판별 여부를 앱 자체가 갖는 권한의 개수로 판단하기에 정상 앱과 동일한 수준의 권한을 갖은 악성 앱을 탐지하지 못하거나 또는 많은 권한을 소유한 정상 앱을 악성 앱으로 판별(오탐)할 위험이 있다. 본 논문에서는 모든 권한의 개수를 확인하지 않고 악성 권한으로 분류한 권한만을 확인하여 악성 앱을 판별한다.

두번째 방법은 카테고리에 따라 앱을 분류한 후 동일 카테고리에 속한 정상 앱과 악성 앱의 권한을 비교하여 악성 앱에서만 대표적으로 나타나는 권한을 분류하여 악성 앱을 탐지하는 방법[10]이다. 본 논문과 가장 비슷한 방법이며 악성 앱의 판별을 쉽고 정확게 구별할 수 있다. 하지만 해당 방법은 APK 파일만으로 카테고리를 분류할 수 있는 방법이 존재하지 않아 실험자가 원하는 특정 카테고리의 악성 앱이 충분하지 않은 단점이 있다. 본 논문에서 악성 앱은 카테고리를 분류하지 않으며 구글 플레이스토어에 등록된 정상 앱으로 구글플레이스토어에서 분류한 방식으로 카테고리를 사용할 수 있다.

마지막 세번째는 악성 앱을 검사하기 위해 소스코드, API, 권한 등 여러 환경변수를 빅데이터에 적용해 악성 앱을 검사하는 방법[11],[12]으로 정확도가 높은 것이 장점이다. 하지만 하나의 검사 프로세스를 제작하기 위해서는 안드로이드, 빅데이터 등 여러 배경지식이 필요해 제작하기 어려운 단점이 있다. 따라서 본 논문에서는 추출하기 쉬운 권한만을 사용하여 악성 앱을 쉽게 판별할 수 있는 방법을 제시한다.

Ⅲ. 제안 방법

3.1. 카테고리 분리

카테고리의 분류는 APK 파일만 자체만으로 분류 할 방법이 없으며 구글플레이스토어 웹사이트의 html에서 확인할 수 있다. 만약 구글플레이스토어에 등록된 게임 앱 'sample.apk'에 대한 카테고리를 분류하기 위해서는 오픈소스 툴인 'curl'과 'xmlint'를 사용하여 카테고리를 분류할 수 있다. 'curl'의 역할은 HTTP Request 및 Response를 확인하는 툴로서 HTTP Response Body에 포함되어 있는 카테고리를 확인하기 위해서 사용한다. curl을 사용해 카테고리를 확인하는 방법은 '그림 2'와 같다. 구글 플레이스토어 내 sample.apk 상세 페이지(https://play.google.com/store/apps/details?id=sample_app_package)를 'curl'을 사용해 호출할 때 응답 값에 카테고리에 대한 값이 존재한다. 하지만 응답 값에는 카테

고리 외에도 많은 html 코드들을 포함하고 있고 카테고리 값만을 추출하기 위해 xmlint를 사용한다. xmlint는 xml 포맷의 파일 내부를 탐색하는 오픈소스 툴로 curl이 호출하는 응답 값을 xml 확장자로 저장 후 카테고리를 특징할 수 있는 부분(/store/app/category/'카레고리 값') 을 탐색한다.

두 오픈소스를 이용해 'sample.apk' 페이지에서 '그림 2' 법으로 카테고리를 추출한 결과 'GAME_ACTION' 이라는 결과를 얻을 수 있어 'sample.apk'파일은 게임 카테고리의 apk 파일인 것을 확인할 수 있다.

```
In: curl -L https://play.google.com/store/apps/details?id=sample
_app_package -o sample.html && xmllint −html −xpath /html/
body/div[1]/div[4]/c-wiz/div/div[2]/div/div/main/c-wiz[1]/c-wiz
[1]/div/div[2]/div/div[1]/div[2]/div[1]/div[1]/span[2]/a sample.ht
ml 2>/dev/null
Out: <a itemprop="genre" href="/store/apps/category/GAME_A
CTION" class="hrTbp R8zArc">Action</a>

※ In: 입력 값 Out: 출력 값
```

Fig. 2 Categorization using 'curl' and 'xmlint'

3.2. 권한 추출

APK파일에서 권한을 추출을 위해 본 연구에서는 Androguard를 사용한다. Androguard란 'Anthony Desnos' 외 3인이 제작한 파이썬 기반 오픈소스로 'APK파일'을 분해하여 쉽게 역분석 할 수 있게 해준다. Androguard를 사용해 '그림 3'과 같이 Sample.apk(경로:C:\Analysis\sample.apk)에서 권한 추출을 한다.

```
In [1]: from androguard.misc import AnalyzeAPK
In [2]: a, d, dx = AnalyzeAPK("C:\\Analysis\\sample.apk")
In [3]: a.get_permissions()
Out[3]:
['Permission 1', 'Permission 2',….]

※ In: 입력 값 Out: 출력 값
※ 윈도우의 경우 파이썬으로 구성된 androguard 특성상 경로에
   \\ 두 번을 입력해야 함
```

Fig. 3 Permission extraction using 'Androguard'

3.3. 악성 권한 상정

3.2에서 추출 한 권한을 모두 취합 후 수식(Malicious permissions extraction formula) (1)에 대입해서 악성권한을 판별 할 수 있는 수치를 계산한다. 이때 악성 앱의 수는 353(193+160)개이고 정상 앱의 수는 182개로 두 실험군의 비율은 1:1이 되지 않는다. 만약 '악성 앱'의 개수와 '정상 앱'의 개수가 1:1로 되지 않는다면 표본의 차이로 인해 결과 값은 편차를 보일 수 있어 두 표본의 비율을 1:1로 맞추기 위해 보정계수 'C'가 필요하다 보정 계수 'C'를 정하는 방법은 '악성 앱 개수'/'정상 앱 개수'로 구할 수 있으며 본 실험에서 악성 앱의 개수는 353개 정상 앱의 개수는 182개이므로 'C'의 값은 '353/182 '로 설정할 수 있어 본 연구에서 최종 Result 값은 (2)의 수식으로 구할 수 있다.

$$Result = \sum \text{악성앱 권한목록} - \sum(\text{정상 앱권한목록} * C(353/182)) \quad (1)$$

$$Result = \sum 2020\text{년}(193\text{개}),\ 2019\text{년}(160\text{개})\ \text{악성앱 권한목록} - \sum \text{정상 게임앱}(182\text{개})\text{권한목록} * C(353/182) \quad (2)$$

3.4. 악성 권한

3.3에서 추출한 Result 값을 사용하여 악성 권한을 선별하며 이 때 악성 권한과 정상권한을 판별하는 Result 값의 기준은 실험자가 정한다. 이 때 Result 값을 적정 수준 이상으로 높이면 비정상 권한도 정상 권한으로 볼 수 있으며 Result 값을 너무 낮추면 정상적으로 필요한 권한도 악성 권한으로 볼 수 있다. 본 실험에서는 Result 값의 기준을 60으로 잡았으며 그 이유는 'Write_Contacts' 권한의 Result 값은 60이며 그 다음 순위의 권한인 'Set_Wallpaper'권한은 Result 값이 52로 충분한 Result 값의 차이가 존재하기 때문에 Result 값이 60 이상인 권한을 악성 권한으로 선정하였다. Result 60 이상으로 계산되어 악성 권한으로 판별한 결과는 '표 4'와 같다.(소수점은 내림)

Table. 4 Final malicious permissions

Permission name	Result
Read_Phone_state	164
Read_Contacts	163
Read_SMS	161
Receive_SMS	140
Access_Fine_Location	125
Send_SMS	125
Call_Phone	118
System_alert_window	114
Record_Audio	99

Permission name	Result
Read_External_Storage	93
Camera	90
Change_WIFI_State	90
Get_Accounts	81
Read_Call_Log	76
Get_Tasks	75
Request_Ignore_Battery_Optimizations	74
Access_Coarse_Location	73
Write_SMS	69
Write_Settings	65
Process_Outgoing_call	62
WRITE_CONTACTS	60

3.5. 악성 앱 판별 기준

3.4에서 선정한 악성 권한을 이용해 악성 앱으로 판별할 수 있는 기준 수립이 필요하다. 기준을 잡기 위해서 (3)의 식을 사용하였으며 실험식에서 실험자는 상수 M의 값을 정해야한다. M의 값은 실험자가 미탐과 오탐 중 어느 것에 중요도를 가지고 가중치를 설정하는 부분이다. 만약 미탐에 중점을 둔다면 M의 값을 0에 가깝게 하고 오탐에 중점을 둔다면 M의 값을 1에 가깝게 설정한다. 다만 M의 값이 0에 가까이 수렴 시 정상 앱을 악성 앱으로 판별할 수 있으며 M의 값을 1에 가깝게 수렴하면 악성 앱을 정상 앱으로 판별할 수 있다. 본 실험에서는 오탐 보다 미탐에 가중치를 더 두어 0.4의 M 값을 설정하였다. 게임카테고리 앱의 경우 APK 당 평균 12개의 권한을 가지고 있으므로 'M×평균 값'을 계산한 결과 4.8의 값이 얻어진다. 즉 APK파일 내 Result 값이 60 이상인 권한이 최소 5개 이상 가지고 있다면 악성 앱으로 판별한다.

$$M \leq \frac{APK\text{파일 내 존재하는 악성 권한 수}}{APK\text{파일이 갖는 평균 권한 개수}} \qquad (3)$$

Ⅳ. 시 험

4.1. 실제 악성 게임 앱에 판별 기준 적용

구글 프로텍트를 우회하여 구글플레이 스토어에 등록 된 악성 게임 앱 4개에 대해 검증을 시행한 결과 4개 게임 앱 모두 악성 앱으로 검출한 결과를 '표 5'와 같이 확인 할 수 있었으며 적합한 권한의 수가 기준치보다 두 배 이상으로 검출 된 것을 통해 정상 앱에서는 갖고 있지 않지만 악성 앱은 가지고 있는 권한이 존재한다는 것을 확인할 수 있었다.

Table. 5 Verification of malicious apps

APK number	Hit count
1	12
2	12
3	8
4	9

4.2. 정상 게임 앱에 판별 기준 적용

22.01.19일 기준 임의의 신규 게임 앱 10개의 권한을 추출 후 상정한 악성 권한을 적용해본 결과 '표 6'과 같이 3번 어플리케이션 파일을 악성 앱으로 검출해 약 90%의 적중하는 것을 확인할 수 있었다. 다만 10%의 오탐이 발견된 이유는 악성 앱의 오탐보다 미탐이 발견되지 않도록 기준을 정하여 오탐이 발생하였다.

Table. 6 Verification of normal apps

APK number	Hit count
1	0
2	2
3	5
4	0
5	1
6	0
7	0
8	0
9	0
10	2

4.3. 상용 악성 앱 검출 솔루션과 비교

본 논문에서 제안하는 방법과 상용 악성 앱 검출 솔루션 VirusTotal과 비교한 결과 다음 '표 7'과 같은 결과를 도출할 수 있었다. '표 7'에서 정상 앱에 대해서만 10% 오탐이 있었고 악성 앱에 대해서는 100% 판별하였다. 즉 미탐율을 낮추는데 가중치를 둔 결과이며, 단순한 방식으로 도출된 결과이다. 즉, 상용 악성 앱 검출 솔루션의 경우 권한 외에도 여러 가지 항목을 점검하지만 본 논문에서 제시한 방법은 권한만을 활용하여 악성 앱을

검출하며 오탐보다 미탐이 없도록 기준을 잡아 성능을 비교 하였다.

Table. 7 Compared to commercial solutions 표

Target	Malicious app	Normal app
Paper	100%	90%
VirusTotal	100%	100%

4.4. 시험 결과

본 논문에서 제시한 방법으로 검증한 결과 악성 권한을 갖는 악성 앱에 대해서 100%의 검출율을 보여 상용 악성 앱 검출 솔루션인 VirusTotal에 비교해 뒤처지지 않는 검사결과를 획득할 수 있었다. 향후 구글프로텍트를 우회하여 구글플레이스토어에 등록 된 악성 게임 앱을 대량으로 확보할 수 있다면 좀 더 신뢰성 높은 테스트 결과가 제공되리라 확신한다.

시험결과 중 정상 앱을 악성 앱으로 판정하는지에 대한 오탐율을 검증한 결과 10%의 오탐 율을 보여 정상 앱으로 판정 된 앱의 경우 더 이상의 판정이 필요 없으나 악성 앱으로 오탐된 앱에 대해서만 다른 방법(백신)을 사용 해 악성 앱에 대해 검증을 수행하면 되기에 악성 앱을 점검하기 위해 소요되는 전체 시간과 프로세서의 리소스는 감소된다.

Ⅴ. 결 론

정적분석을 통한 악성 앱 검사에는 '권한', 'API', '소스코드' 등 많은 요소들을 사용할 수 있다. 이중 '권한'만을 악성 앱 탐지 요소로 선택한 이유는 안드로이드의 많은 지식 없이 쉽게 권한에 접근 및 추출이 가능한 장점으로 누구나 악성 앱을 판별할 수 있는 악성 권한을 제작하고 꾸준히 업데이트를 할 수 있는 장점이 있다. 하지만 권한 외에도 API, 소스코드 등 여러 환경 변수들을 사용하는 상용프로그램(ex. VirusTotal)와 비교했을 때 권한만을 사용하여 악성 앱을 탐지하는 방법은 정상 앱을 악성 앱으로 판별하는 오탐이 발견되는 것을 확인할 수 있었다. 하지만, 이러한 결과도 대량의 악성 앱과 정상 앱을 진단할 경우 악성 권한에 대한 적중의 수(M)의 값을 보정하면 미탐, 오탐을 최소화하면서 성능은 높일 수 있는 쉬운 방법이다. 따라서 본 논문에서 제시하는 방법은 악성 앱의 미탐이 발생 되지 않는 것이 더 중요하다 생각되어 미탐을 줄이는 방법으로 설계해 적용하였다.

또한 논문에서 제시한 방법에 대해 한가지 한계점이 존재하였다. 한계점은 악성 권한을 제작하기 위해 필요한 데이터 중 필수 환경 변수인 카테고리 분류에 대해 한계점이 존재한다. 핵심 환경 변수인 카테고리는 APK 파일 내부에 존재하지 않아 이를 분류하기 위해서는 구글 플레이스토어 웹 사이트에서 등록된 앱만 카테고리 분류가 가능해 그 결과 구글 플레이스토어에 등록 되지 않은 앱은 논문에서 제시한 방법으로 악성 앱 검사가 불가능하다는 한계점을 가지고 있다.

이러한 한계점으로 인해 검증에서 또한 한계점이 있다. 악성 앱에 대한 미탐 검증에서 실험 대상이 될 악성 앱도 카테고리 분류를 하기 위해서는 구글플레이스토어에 등록된 악성 앱이어야 했다. 하지만 구글 플레잇스토어에 악성 앱이 등록되기 위해서는 'Google protect'를 우회하여 등록되어야 했으며 위의 조건을 만족한 악성 게임 앱을 충분하게 획득하지 못해 검증이 부족한 아쉬움이 있었다. 이는 추후에 구글 플레이스토어에 등록된 악성 게임 앱을 대량으로 확보한다면 신뢰성 높은 결과를 제공하리라 확신한다.

본 논문에서 제시한 안드로이드 권한이란 누구나 접근하기 쉬우며 다른 환경변수 없이 권한만으로 악성 앱을 검출하기에 충분하다고 판단된다. 즉 누구나 안드로이드 악성 앱 검출 프로세스 제작이 가능하며 매년 새로운 악성 앱만 구할 수 있으면 쉽게 악성 권한의 업데이트가 가능해 모바일 앱 개발 시 필요 권한 설계에 대해 모바일 앱 기획자 및 개발자들에게 권한에 대한 가이드를 제공할 수 있을 것으로 기대된다.

ACKNOWLEDGEMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICAN (ICT Challenge and Advanced Network of HRD) Program (IITP-2022-2020-0-01825) supervised by the IITP (Institute of Information and Communications Technology Planning and Evaluation).

References

[1] Statcounter. Mobile Operating System Market Share Worldwide [Internet]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202012-202112.

[2] Securelist by Kaspersky. IT threat evolution Q1 2021. Mobile statistics. (2021, May). [Internet]. Available: https://securelist.com/it-threat-evolution-q1-2021-mobile-statistics/102547/.

[3] Google. Protect your device from malicious apps with Google Play Protect [Internet]. Available: https://support.google.com/googleplay/answer/2812853?hl=ko.

[4] Securityworld. 21 Malicious Game Apps That Break Through Google Play Store's Surveillance Network, (2020, October). [Internet]. Available: https://www.boannews.com/media/view.asp?idx=92163.

[5] sk3ptre. Popular Android threats in 2019 [Internet]. Available: https://github.com/sk3ptre/AndroidMalware_2019.

[6] sk3ptre. Popular Android malware seen in 2020 [Internet]. Available: https://github.com/sk3ptre/AndroidMalware_2020.

[7] Google. Application Sandbox [Internet]. Available: https://source.android.google.cn/security/app-sandbox?hl=ko.

[8] Google. Permission in Android [Internet]. Available: https://developer.android.com/guide/topics/permissions/overview?hl=ko.

[9] G. W. Park, Tamer Abuhmed, D. H. Min, D. H. Nyang, and K. H. Lee, “Improving Permission-based Android Malware Detection Using Control Flow Graph and Library Dependency Information,” *The Journal of KINGComputing*, vol. 15, no. 6, pp. 15-24, Dec. 2019.

[10] B. Kim, J. I. Im, and Y. H. Jo, “Privacy Situation and Countermeasures of Financial Apps based on the Android operating system,” *The journal of the institute of internet, broadcasting and communication*, vol. 14 no. 6, pp. 267-272, Jun. 2014.

[11] G. Y. Kim, S. R. Kim, Y. J. Jeon, and J. S. Kim, “A Trend of Machine Learning for Android Malware Detection and Permission Based Android Malware Detection using Deep Learning,” *Journal of Digital Forensics*, vol. 14, no. 3, pp. 316-326, Sep. 2020. DOI: 10.22798/kdfs.2020.14.3.316.

[12] S. W. Min, H. J. Jo, J. S. Shin, and J. C. Ryu, “Android Malware Detection Method Using Machine Learning,” *Korean Institute of Information Scientists and Engineers*, vol. 39, no. 1(C), pp. 280-282, Jun. 2012.

박종찬(Jong-Chan Park)

충남대학교 생물과학과 이학사
한국정보기술안전 선임컨설턴트
부산외국어대학교 정보보호학과 공학석사 재학
※관심분야 : 시스템 취약점 분석

백남균(Nam-Kyun Baik)

98.2 숭실대학교 전자공학과 공학사
01.2 숭실대학교 전자공학과 공학석사
11.2 숭실대학교 전자공학과 공학박사
00 ~ 17 한국인터넷진흥원 수석연구원
19.3.~현재. 부산외국어대학교 정보보호학과
※관심분야 : 스마트융합보안, 정보보안컨설팅