안드로이드 광고 인젝션 앱 분석 방법에 대한 연구

나윤종*, 오세라*, 김영갑*†
*세종대학교 정보보호학과 보안공학연구실
e-mail : nooryyaa2@gmail.com, terious551@sju.ac.kr, alwaysgabi@sejong.ac.kr

A Study on the Analysis of Malicious Advertisement Injection on Android Application

Yoon-Jong Na*, Se-Ra Oh*, Young-Gab Kim*
*Security Engineering Lab., Dept. of Computer and Information Security, Sejong University

요 약

모바일 앱 환경에서의 광고는 앱 내부에 배너형, 팝업 형 등 다양한 형태의 광고를 띄우고, 사용자의 터치를 유도하는 등의 방법으로 광고 수익이 발생한다는 특징이 있다. 하지만, 악의적 광고주 또는 공격자는 이러한 특징을 악용해 앱 내에서의 정당한 광고를 다른 광고로 바꾸거나, 사용자에게 의도되지 않는 광고를 노출하는 등의 광고 인젝션을 발생시킨다. 광고 인젝션은 광고주에겐 수익 저하, 서비스 제공자에겐 서비스의 품질 저하, 서비스 이용자에겐 불편함을 야기하는 등 문제가 된다. 모바일 앱 환경에서의 광고 인젝션에 대한 연구 및 탐지 방법에 대한 연구는 몇몇 진행되었으나, 미진한 상황이다. 본 연구에서는 모바일 환경에서의 광고 인젝션 탐지를 위한 광고 인젝션 앱만의 특징 수집을 위해 광고 인젝션 앱, 특히 사용자가 많은 안드로이드 환경에서의 앱을 분석하는 방법을 제시한다.

1. 서론

모바일 앱 환경에서의 광고는 오프라인 환경의 광고와 다른 방법을 이용할 수 있다. 앱 개발자가 앱 내에 배너형 광고를 추가하거나 팝업 광고 등 다양한 형태의 광고를 넣고 사용자의 터치를 유도하거나 사용자의 정보를 수집해 사용자가 관심있을 만한 내용을 제공하는 등 다양한 방법으로 앱 내에 광고를 배치한다. 사용자가 광고를 클릭하게 되면 광고주의 사이트로 이동하게 되며 광고주는 거기에 대한 대가를 지불하면 된다.

하지만 원래 서비스 제공자가 의도한 광고와 전혀 다른 광고를 띄우거나, 사용자가 이용하기 불편하게 사용자의 의지와 상관없이 아무 때나 아무 장소에 팝업 광고를 띄우는 등의 광고 인젝션(advertisement injection; ad injection)으로 인해 서비스 제공자, 서비스 이용자, 광고주 모두에게 피해를 주기도 한다[1].

KISA (Korea Internet & Security Agency)에서 발표한 “2017 년 인터넷 이용실태조사 최종보고서”[2]를 참조하면 <표 1>과 같이 국내에서 대부분의 가구가 모바일을 이용해 인터넷 사용하고 있음을 알 수 있다. 따라서 모바일 이용자를 대상으로 한 광고 인젝션 앱 또한 무시할 수 없다.

<표 1> 인터넷 접속 기기 비율[2]

기기	스마트폰	데스크탑	노트북	디지털 TV	스마트 TV	게임기
비율	94.1%	61.2%	31.6%	35.6%	13.2%	3.0%

광고 인젝션과 관련된 주요 연구를 살펴보면, 조상현 외[1, 3]는 인터넷 광고 인젝션의 유형에 대해 연구하였으며, 모바일 환경에서의 광고 인젝션을 사례를 통해 모바일 광고 인젝션을 분석하고 PC 환경과 어떻게 다른 지 소개하였다. 오세라 외[4]는 구글 플레이에 존재하는 앱 중 광고 인젝션을 포함하고 있을 가능성이 높은 앱을 찾아내기 위한 도구를 개발하였다. 하지만 광고 인젝션 앱에 대한 연구나 분석 결과가 많지 않아 의심되는 앱을 수집하는데 그치고 있다. 이러한 한계점을 극복하고 광고 인젝션 앱 자동 탐지 연구를 심화하기 위해서는 광고 인젝션에 대한 특징들의 분석을 통해 광고 인젝션이 의심스러운 앱을 수집하고 분석할 필요가 있다.

따라서 본 연구에서는 모바일, 특히 사용자가 많은 안드로이드 환경에서 어떻게 광고 인젝션 앱을 분석해야 효과적으로 분석하고 탐지를 위한 정보를 수집할 수 있는지 알아본다. 이를 위해, 세가지 광고 인젝션 앱을 분석하여, 모바일 환경에서의 광고 인젝션 앱의 특징들을 살펴본 뒤 이들을 분석하는 효과적인 방법에 대해 연구한다.

† 교신 저자

이 연구는 NAVER 주식회사로부터 지원되었습니다.

본 논문의 구성은 다음과 같다. 2 장에서는 세가지 광고 인젝션 앱[1]의 차별적인 특징들을 살펴본다. 이후 3 장에서는 앞서 살펴본 특징들을 통해 어떻게 광고 인젝션 앱을 효과적으로 분석할 수 있는지 연구한 뒤 결론짓는다.

2. 광고 인젝션 앱 수집 및 분석

광고 인젝션 앱을 효과적으로 분석할 방법에 대해 연구하기 위해 광고 인젝션 앱의 차별적인 특징을 먼저 알아볼 필요가 있다. 따라서 광고 인젝션 앱 분석 방법 연구에 앞서서 광고 인젝션 앱을 우선 수집하고 분석하여 보았다.

광고 인젝션을 포함하고 있는 앱을 수집하기 위해서[3]의 GPC(Google Play Crawler)를 사용하였다. 수집한 앱이 실제로 광고 인젝션을 포함하는지 여부를 판별하기 위해선 자세한 분석이 필요하다. 따라서 먼저 광고 인젝션 앱을 동적 분석 및 정적 분석을 통해 분석해 보고 특징들을 살펴보았다.

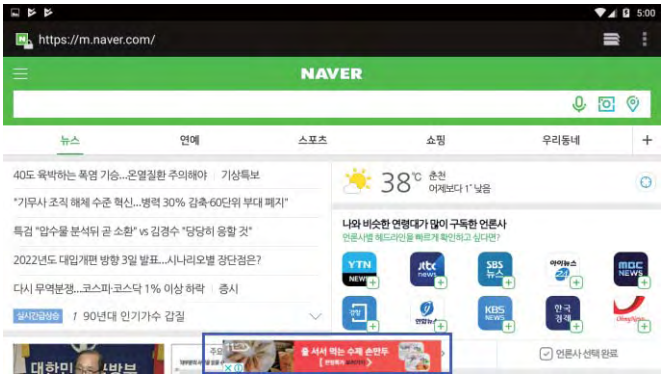
동적 분석은 안드로이드를 구동 시킬 수 있는 가상환경 및 안드로이드 태블릿으로 진행하였다. 안드로이드 버전은 6.0.0 및 7.0.0 으로 2018 년 9 월기준으로 각각 22.7% 및 20.3%의 점유율로 가장 많은 사용자들이 사용중인 버전이다[4]. 사용한 가상환경은 녹스 앱플레이어(NoxPlayer)[5] 및 지니모션(Genymotion)[6]을 사용하였다. 각각의 버전 및 안드로이드 구동 환경은 <표 2>와 같다.

<표 2> 동적 분석에 사용된 가상환경 정보

대상	도구의 버전	안드로이드 버전	API 레벨
녹스 앱플레이어	6.0.0.0	6.0.0	23
지니모션	Personal 2.12.1	6.0.0	23
안드로이드 태블릿	-	7.0.0	24

동적 분석을 통해 발견된 특징의 경우, 우선 ‘S’앱의 경우 (그림 1)처럼 다른 앱 사용 중에 정상적인 앱의 경우 발생하지 말아야 할 배너형 광고가 나타난다. ‘P’ 앱의 경우 사용자의 의지와 관계없이 아무 때나 주기적으로 홈 화면에 팝업 동영상 광고가 나타나는 방식으로 앱 밖에 광고를 띄운다. ‘S’앱과 같은 개발사의 ‘M’앱의 경우 특정 앱이 실행 중일 때만 실행중인 앱 위에 배너형 광고를 띄운다. 또한 (그림 2)와 같이 ‘S’앱, ‘P’앱 및 ‘M’앱은 모두 앱은 앱이 구동 중이 아닐 때도 광고를 띄우기 위한 프로세스가 백그라운드에서 동작한다. 게다가 기기를 재 기동 후, 혹은 프로세스 종료 후에도 앱에서 실행한 프로세스가 한다. 하지만 해당 특징은 일반적인 앱에서도 발생하는 특징으로 더 자세히 알아보기 위하여 정적분석을 수행하였다.

[1] 본 연구에서의 분석 대상 앱에 대하여 관련 업체의 개인 정보 보호를 우려하여 모바일 앱의 이름을 간략히 표기하였음.



(그림 1) 배너형 광고가 다른 앱 위에 발생

(그림 2) 앱이 수행 중이 아닐 때 작동중인 프로세스

AndroidMenifest.xml 파일에서는 해당 앱이 가진 권한을 볼 수 있다. 특히 광고 인젝션을 포함한 앱이 앱이 아닌 곳에 광고를 발생시키는 경우 자주 발견되는 권한이 몇 가지 있다. API 레벨 23 이상인 안드로이드 환경에서는 앱 바깥에서 무언가를 화면에 띄우기 위해서는 SYSTEM_ALERT_WINDOW 권한이 필요하다. 해당 권한은 광고 인젝션을 포함한 앱에서 자주 발견되는 권한이다. API 레벨이 23 미만인 안드로이드 환경에서만 광고 인젝션이 발생할 경우 앱에 해당 권한이 없어서 API 레벨이 23 이상인 안드로이드 환경에 광고 인젝션이 발생하지 않을 수 있으므로 해당 권한의 유무를 확인할 필요가 있다. RECEIVE_BOOT_COMPLETED 또한 자주 발견되는 권한으로, 기기가 부팅 시 광고 인젝션을 포함한 서비스를 구동 시키는데 사용된다. 주로 동적 분석에서 발견된 앱을 종료하였거나 앱을 구동 시키지 않았음에도 불구하고 실행된 서비스는 RECEIVE_BOOT_COMPLETED 권한을 이용하였다. 발견된 권한은 앞서 발견한 백그라운드에서 동작하며 광고를 띄우는 서비스가 이용하는 경우가 많다.

(그림 3)은 SYSTEM_ALERT_WINDOW 권한과 RECEIVE_BOOT_COMPLETED 권한이 AndroidMenifest.xml

에 명시되어 있는 'S' 앱의 예시이다. (그림 4)는 같은 앱의 광고 인젝션 수행 서비스가 다른 앱 위에 그리기 권한이 있는지 체크한 뒤, 광고 SDK(software development kit)를 시작하는 실제 코드의 예시이다. 해당 앱은 앞에서 살펴본 (그림 1)과 같이 사용자가 사용중인 앱과 관계없이 홈 화면을 포함해 배너형 광고를 띄운다. 'M' 앱의 경우 'S' 앱과 다르게 특정 앱이 수행 중일 때만 배너형 광고가 발생하였으며, 'S' 앱과 동일한 권한을 가졌다. 또한 광고 인젝션 관련 서비스 시작 전 권한을 확인하는 부분 또한 동일하게 존재했다.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<permission android:name="
love.tarot2014.saju.relationship.permission.C2D_MESSAGE" android:
protectionLevel="signature"/>
<uses-permission android:name="
love.tarot2014.saju.relationship.permission.C2D_MESSAGE"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="
com.google.android.gms.permission.ACTIVITY_RECOGNITION"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.REORDER_TASKS"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.READ_USER_DICTIONARY"/>
<uses-permission android:name="android.permission.WRITE_USER_DICTIONARY"/>
```

(그림 3) AndroidMenifest.xml 에서 발견된 광고 인젝션을 위해 사용된 권한

```
protected void onActivityResult(int paramInt1, int paramInt2, Intent paramIntent)
{
  if ((Build.VERSION.SDK_INT >= 23) && (paramInt1 == OVERLAY_PERMISSION_REQ_CODE))
  {
    if (!Settings.canDrawOverlays(this))
    {
      Log.i("TAG", "필요한 퍼미션을 사용 할 수 없어 TNAd 사용이 불가능 합니다.");
      return;
    }
    startTNAdOBSdk();
  }
```

(그림 4) 다른 앱 위에 그리기 권한 확인 부분

(그림 5) 사용자의 의지와 관계없이 발생한 팝업 동영상 광고

'P' 앱의 경우에는 (그림 5)와 같이 주기적으로 홈화면에 팝업 동영상 광고를 띄우는 방식으로 광고 인젝션을 수행한다. 'P' 앱의 경우 일부 안드로이드 모바일 기기에 초기상태에 설치되어 있는 앱임에도 불구하고 광고 인젝션을 수행한다. 'P' 앱 또한 SYTEM_ALERT_WINDOW 권한 및 RECEIVE_BOOT_COMPLETED 권한을 가지고 있으며 이를 광고 인젝션에 이용한다. 'S' 및 'M' 앱과 다른점은 월정액을 결제할 수 있으며, 결제자의 경우 광고를 띄우지 않는다는 점이다. 'P' 앱의 경우 코드상에서는 광고를 시작하는 부분이 난독화 되어 있어 분석을 어렵게 하고 있다.

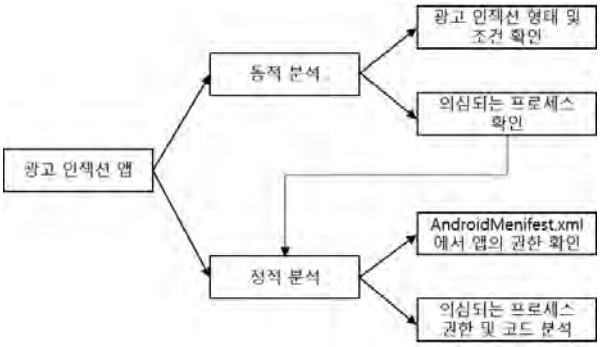
앞서 분석한 광고 인젝션 앱들의 특징을 정리하면 <표 3>과 같다. 광고 인젝션을 포함한 앱은 다른 일반 앱들과 몇 가지 다른 특징을 가진다. SYSTEM_ALERT_WINDOW 권한을 광고에 이용하는 등 여러 특징들을 결합시키면 광고 인젝션에 이용할 수 있게 되는 것이 있으므로, 이들을 중점으로 분석하면 탐지 및 모니터링에 활용할 수 있는 특징들을 쉽게 추려낼 수 있을 것이다. 이를 바탕으로 어떻게 하면 효과적으로 광고 인젝션 앱을 분석할 수 있을지 3 장에서 알아본다.

<표 3> 분석한 일부 광고 인젝션 앱들의 특징

앱	증상	특징
'S'앱	상시 배너형 광고 발생	SYSTEM_ALERT_WINDOW 권한 및 RECEIVE_BOOT_COMPLETED 권한 이용. 광고 인젝션 서비스 시작 시 권한 확인.
'M'앱	특정 앱 실행 중 배너형 광고 발생	SYSTEM_ALERT_WINDOW 권한 및 RECEIVE_BOOT_COMPLETED 권한 이용. 광고 인젝션 서비스 시작 시 권한 확인.
'P'앱	홈 화면에 주기적으로 팝업 동영상 광고 실행	SYSTEM_ALERT_WINDOW 권한 및 RECEIVE_BOOT_COMPLETED 권한 이용. 월정액 결제자는 광고를 띄우지 않음. 코드 난독화.

3. 광고 인젝션 앱 분석 방법 연구

앞선 분석을 통해 광고 인젝션이 포함된 앱들의 경우 일반 앱들과 다른 특징을 몇 가지 발견할 수 있다. 이들을 바탕으로 어떻게 하면 효과적으로 광고 인젝션 앱을 분석할 수 있는지 알아본다. (그림 6)은 본 연구에서 제안한 광고 인젝션 앱 분석 방법을 요약한 것이다.



(그림 6) 광고 인젝션 앱 분석 방법

동적 분석에는 안드로이드를 구동 시킬 수 있는 가상환경 및 안드로이드 모바일 기기를 이용한다. 구축한 안드로이드 환경에서는 어떤 앱에 의해 광고 인젝션이 발생하는지 정확히 식별하기 위하여 다른 앱을 설치하지 않은 초기 상태에서 분석을 진행하도록 한다. 이후 광고 인젝션이 다른 앱에도 영향을 주는지 확인해야할 경우 추가로 분석한다. 기기나 환경, 안드로이드 버전에 따라 광고 인젝션이 발생할 수도 발생하지 않을 수도 있으므로 다양한 환경을 준비한다.

앞서 동적 분석을 통해서 발견한 몇 가지 특징으로는 우선 앱 외의 곳에서 광고가 주로 발생하였다. 또한 앱을 사용 중이지 않은 경우에도 광고 인젝션을 포함한 앱에서 실행된 프로세스를 확인할 수 있었다. 따라서 분석 시 의심되는 앱을 안드로이드 환경에서 실행한 뒤 광고가 어떤 조건하에 발생하는지 분석한 뒤, 앱에 의해 실행된 프로세스를 확인하고 의심되는 프로세스를 추려낸다. 앱을 사용 중이지 않은 경우에 발생하는 프로세스의 경우 일반 앱에서도 볼 수 있는 특징으로 더 자세한 분석이 필요하다. 이 경우 정적 분석을 수행한다.

정적 분석은 광고 인젝션 의심 앱을 안드로이드에서 apk 파일을 추출한 뒤, 디컴파일 툴을 이용하여 디컴파일하여 AndroidMenifest.xml 파일을 통해 앱이 가진 권한 및 서비스 등을 분석한다. 이후 앞서 수행한 동적 분석을 통해, 광고 인젝션을 수행하는 것으로 의심된 서비스를 AndroidMenifest.xml 파일에서 찾아 디컴파일한 코드에서 분석한다. 또한, 광고 인젝션 기능을 SDK 을 이용하여 수행할 수 있으므로[3] 사용하는 SDK 에는 어떤 코드를 포함하고 있는지 살펴본다. 이후 실제 코드에서 자세한 분석을 하여 서비스가 광고 인젝션 기능을 하는지 살펴본다. 디컴파일 및 분석에 사용한 툴들은 <표 4>와 같다. jd-gui 를 정적 분석에 이용할 경우 apktool 을 이용해 안드로이드의 달빅(dalvik) 가상머신에서 이용되는 바이트 코드(byte code)인 dex 파일을 추출할 필요가 있다. 해당 dex 파일을 자바 아카이브 파일인 jar 파일로 변환해주면 jd-gui 를 이용해 분석할 수 있다. 또한 apktool 을 이용해 dex 파일을 저급언어로 디컴파일한 smali 코드를 이용해 직접 분석할 수도 있다. jadx 를 사용할 경우 apk 파일을 바로 자바 언어로 디컴파일 해 준다.

<표 4> 정적 분석에 이용된 도구 정보

도구	도구의 버전	도구의 용도
apktool	2.3.3	apk 파일을 디컴파일 하여 smali 코드를 생성
dex2jar	2.0.0	apk 파일에서 추출한 classes.dex 파일을 jar 파일로 변환하기 위한 도구
jd-gui	windows-1.4.0	추출한 jar 파일을 이용해 디컴파일하기 위한 자바 디컴파일 도구
jadx	0.6.1	apk 파일을 자바로 디컴파일 하기 위한 자바 디컴파일 도구

앞서 정적 분석을 통해 살펴본 몇 가지 특징은 다음과 같다. 우선 AndroidMenifest.xml 파일에서 SYSTEM_ALERT_WINDOW 권한을 포함한 몇 가지 권한이 광고 인젝션을 수행하는 서비스에 이용되었다. 또한 RECEIVE_BOOT_COMPLETED 를 이용하여 기기가 부팅될 시 동적 분석 과정에서 발견된 광고 인젝션을 수행하는 서비스를 실행시켜 앱을 사용 중이 아닐 때도 광고 인젝션을 수행하도록 하기도 하였다. 따라서 정적 분석 시에는 앱이 사용중인 권한을 확인하고, 앞서 동적 분석을 통해 발견한 의심되는 서비스가 어떤 권한을 사용하는지 살펴본다. 그리고 실제 코드에서 해당 서비스가 광고 인젝션을 수행하는지 확인한다.

4. 결론

기존 연구에서는 광고 인젝션에 대해 연구하거나 모바일 환경에서 광고 인젝션을 포함한 앱을 탐지하기 위한 연구가 진행되었다. 그러나 의심되는 앱을 수집하는데 그쳤으며 연구 또한 많이 부족하여 확실하게 탐지하는데 무리가 있다. 이러한 문제를 해결하기 위하여 모바일 광고 인젝션에 대한 특징들을 연구하는 것이 선행되어야 하며, 본 연구에서는 효과적인 광고 인젝션 분석 방안에 대해 연구하고 구체적인 과정을 제안하였다. 일반적인 광고를 포함한 앱인지, 광고 인젝션을 포함한 앱인지 구별하기 위한 특징 분석은 광고 인젝션 앱 판별에 있어 중요하다. 제시한 방법은 향후 서비스 제공자, 서비스 이용자, 광고주에게 피해를 주는 모바일 광고 인젝션 앱 탐지 및 모니터링을 위한 특징 분석 및 연구에 사용될 것으로 기대된다.

참고문헌

[1] 조상현, 최현상, 김영갑. "인터넷 광고 인젝션 유형에 대한 연구." 정보보호학회논문지 27.2 (2017): 213-222.
[2] KISA https://www.kisa.or.kr
[3] 조상현, 허규, 최현상, 김영갑. "모바일 광고 인젝션 사례 연구." 정보보호학회논문지 27.5 (2017): 1049-1058.
[4] 오세라, 조상현, 김영갑. "GPC: 모바일 광고 인젝션 앱 탐지를 위한 도구 개발." 예술인문사회융합멀티미디어논문지 7 (2017): 881-889.
[5] Google https://developer.android.com
[6] NOX LIMITED https://bignox.com
[7] Genymobile SAS https://www.genymotion.com/