안드로이드 환경에서의 광고 인젝션 앱 분석

구성민*, 김덕한*, 오세라*, 김영갑*†
*세종대학교 정보보호학과 보안공학연구실
e-mail : koosmin99@gmail.com, lpsi4862@naver.com, terious551@sju.ac.kr,
alwaysgabi@sejong.ac.kr

Analyzing Ad Injection Apps in Android

Seong-Min Koo*, Deok-Han Kim*, Se-Ra Oh*, Young-Gab Kim*
*Security Engineering Lab., Dept. of Computer and information Security, Sejong University

요 약

모바일 환경이 발전함에 따라 기존 PC 환경에서의 보안 위협이 모바일 환경으로 옮겨 짐으로써, 기존 PC 환경에서 발생하던 악성 광고 인젝션 또한 모바일 환경으로 옮겨져 가고 있다. 악성 광고 인젝션은 컨텐츠 제공자에게 정당한 광고의 노출을 방해함으로써 수익 창출을 방해하고, 사용자에게는 원치 않는 광고로 인해 불편함을 야기한다. 이러한 모바일 환경에서의 악성 광고 인젝션을 막기 위해 몇 가지 연구가 진행되었지만 아직 악성 광고 인젝션 앱의 특징에 대한 연구가 미비하다. 따라서, 본 논문에서는 GPC(Google Play Crawler)를 통해 선별한 앱들 중 실제로 악성 광고 인젝션을 수행하는 앱들을 분석하여 악성 광고 앱들의 특징을 도출해 내고, 도출된 특징의 활용 방안에 대해 서술한다.

1. 서론

모바일 환경이 발전하면서 본래 PC 환경을 목표로 하던 공격들이 모바일 환경에서 늘어나는 추세이다. 기존 PC 의 인터넷 환경에서 나타나던 광고 인젝션 또한 모바일 환경에서 나타나고 있다. 광고 인젝션이란 정상적으로 출력 되어서 사용자와 상호작용해야 할 광고들이 출력되지 않고, 엉뚱한 광고가 노출되는 것을 말한다. 이러한 광고 인젝션은 컨텐츠 제공자에게 정당한 광고 노출로 발생하는 수익을 잃게 하고, 사용자에게는 원치 않는 광고로 사용에 불편함을 발생시킨다. 하지만 현재 모바일 환경에서의 광고 인젝션 관련 연구는 국내외로 미진한 상황이다[1].

현재 국내 스마트폰 OS 의 많은 부분을 안드로이드가 차지하고 있기 때문에[2] 안드로이드 환경에서의 악성 인젝션 앱에 대한 연구가 필요하다. 본 논문에서는 향후 악성 광고 인젝션 앱의 연구를 위해 안드로이드 환경을 대상으로 하는 광고 인젝션 앱을 분석하고, 분석된 앱들의 특징 도출한다. 안드로이드 환경에서는 사용자가 원하는 앱을 설치하기 위해 Google Play 를 사용하는 경우가 많기 때문에, 악성 광고 인젝션 앱을 선별하기 위해 GPC(Google Play Crawler)를 사용한다. GPC 는 구글 플레이 스토어에서 사용자의 리뷰를 기반으로 악성 광고 인젝션 의심 앱을 일차적으로 선별해주는 도구이다.

본 논문의 구성은 다음과 같다. 2 장은 모바일 앱 광고 인젝션에 대한 기존 연구를 기술한다. 3 장에서는 악성 광고 인젝션 앱으로 판별된 앱을 분석하고, 특징을 도출해 낸다. 마지막으로 4 장에서 결론 및 본 연구의 시사점에 대해 기술한다.

2. 관련 연구

오세라 외[3]는 광고 인젝션을 수행하는 앱을 탐지하기 위하여 구글 플레이의 앱 페이지에서 리뷰와 별점을 크롤링하여 앱의 광고 인젝션 가능성을 판단할 수 있는 탐지 기법에 관해 연구하였고, 이를 기반으로 한 GPC 와 분석기를 개발하였다. 조상현 외[1, 4]는 PC 환경에서의 인터넷 광고 인젝션 유형을 분류하고 유형별로 분석을 했으며, 모바일 환경에서의 광고 인젝션 유형과 모바일 광고 인젝션 사례에 대해 연구하였다. 모바일 환경에서의 광고 인젝션은 배너(banner)형, 노티 바(notification bar)형, 팝업(pop-up)형으로 분류하였다. 또한 앱이 설치되었을 때와 설치되지 않았을 때의 네트워크 트래픽 변화를 동적으로 분석하고, 소프트웨어 역공학 기법을 통해 광고 인젝션 코드에 대한 정적 분석을 수행하였다.

3. 광고 인젝션 앱 분석

본 논문에서는 크롤러(GPC)와 분석기를 활용하여 광고 인젝션 의심 앱을 일차적으로 선별한 뒤, 해당 앱들을 대상으로 정적, 동적 분석을 수행하여 광고 인젝션 앱을 수집한다. 또한, 수집된 광고 인젝션 앱

† 교신 저자

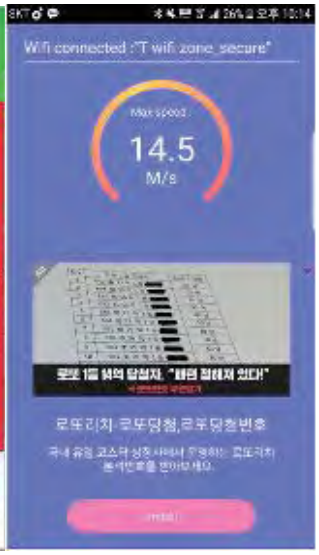
이 연구는 NAVER 주식회사로부터 지원되었습니다.

들을 대상으로 정적, 동적 분석을 수행하여 광고 인젝션 앱을 수집한다. 또한, 수집된 광고 인젝션 앱을 상세 분석하여 모바일 환경에서의 광고 인젝션의 특징을 도출한다. 앱의 악성 광고 인젝션을 직접 관찰하기 위해 Genymotion[5] 등의 가상 환경과 실제 스마트 폰에서 악성 광고 인젝션 의심 앱을 설치하여 악성 광고 인젝션 행위를 확인한 후, 실행중인 서비스 분석, dex 디컴파일러[6]를 통한 소스 분석, AndroidManifests.xml 분석을 통해 악성 광고 인젝션 앱의 특징을 간추려 낸다. 3.1 절과 3.2 절에서는 GPC를 통해 간추려낸 앱 중 실제로 악성 광고 인젝션을 수행하는 앱인 'C' 와 'P' 앱에 대해서 각각 상세한 분석을 수행하였다 [1] . 분석에 사용한 환경은 LG Q6(API 25)와 Galaxy S7 edge(API 24)이다.

3.1. 'C' 앱 분석

'C' 앱을 설치하였을 때 해당 앱은 (그림 1)와 같이 통화를 종료하면 광고를 노출하거나 (그림 2)처럼 전면 광고를 스마트 폰을 사용하는 중에 수시로 노출하는 형태로 악성 광고 인젝션을 수행하는 것을 확인하였다.

(그림 1) 광고 사진



(그림 2) 팝업 광고 창

<표 1>은 광고 인젝션을 위해 'C' 앱이 사용하는 주요 권한이다.

<표 1> 광고 인젝션을 위해 ' C' 앱이 사용하는 권한

- android.permission.REORDER_TASK
- android.permission.RECEIVE_BOOT_COMPLETED

'C' 앱은 다른 앱이 실행 중이더라도 'android.permission.REORDER_TASK' 권한을 이용해 광고 화면을 우선적으로 출력하였다. 또한 악성 광고 인젝션 앱을 설치하고 광고 인젝션이 발생하기 위해서는 광고 인젝션을 수행하는 서비스가 백그라운드에서 계속 유지되어야 하므로, 서비스를 계속 유지하기 위해서 'android.permission.RECEIVE_BOOT_COMPLETED' 권한을 사용한다. 백그라운드에서 항상 동작하는 서비스를 분석한 결과, 악성 광고 인젝션을 수행하는 서비스는 'AdvHightCommonService' 이었으며 API 버전에 따라 'AdvLowCommonService'가 실행되지만 기능상의 차이는 없는 것으로 확인하였다. (그림 3)과 같이 'AdvHightCommonService' 서비스에서 브로드 캐스트 리시버를 통해 전화를 걸거나 전화의 수신을 감지하고 출력될 광고창을 구성해 주는 코드를 확인했다.

```
public class AdvHightCommonService
  extends JobService
{
  private static int b = 1;
  private static AdvHightCommonService c;
  private List<a> a;
  private String d = "XHTCaller";

  private void a()
  {
    if (this.a == null) {
      this.a = new ArrayList();
    }
    Object localObject = new CallerControl(getBaseContext());
    this.a.add(localObject);
    localObject = new ScreenOnService(getBaseContext());
    this.a.add(localObject);
    localObject = new KeyService(getBaseContext());
    this.a.add(localObject);
  }
```

(그림 3) 광고 인젝션을 수행하는 서비스

3.2. 'P' 앱 분석

'P' 앱은 휴대폰이 충전 중일 때 (그림 4)와 같이 광고 인젝션을 수행한다.

(그림 4) 'P' 앱 광고 인젝션

<표 2>는 'P'앱이 광고 인젝션을 수행하기 위해 사용하는 권한이다.

<표 2> 광고 인젝션을 위해 ' P' 앱이 사용하는 권한

- android.permission.SYSTEM_ALERT_WINDOW
- android.permission.RECEIVE_BOOT_COMPLTED

'P' 앱은 다른 앱 위에 우선적으로 광고를 출력하기 위해 'android.permission.SYSTEM_ALERT_WINDOW' 권한을 사용하며, 광고 서비스를 유지하기 위해서

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

‘android.permission.RECEIVE_BOOT_COMPLTED’을 사용한다.

직접적으로 광고 인젝션을 수행하는 것은 ‘P’ 앱의 ‘ScreenLockService’ 서비스로 확인되었다. ‘ScreenLockService’ 서비스는 (그림 5)에서 볼 수 있듯이 백그라운드 서비스로 실행되고, (그림 6)에 있는 ‘BetteryCheckReceiver’를 이용하여 휴대폰의 배터리 상태를 확인하고 충전 중이라면 ‘ScreenLockService’를 실행한다. (그림 7)은 ‘ScreenLockService’ 코드의 일부이다. ‘BetteryCheckReceiver’에서 실행된 ‘ScreenLockService’가 광고를 출력하는 것을 확인할 수 있다. ‘ScreenLockService’는 광고 라이브러리를 이용하여 광고를 노출시켰다.

(그림 5) ‘P 앱의 백그라운드 서비스

```
if (l - ((Long)Pref.load(paramContext, "pref_key_long_last_power_connected_receiver")).longValue() <= 60000L)
{
  Pref.save(paramContext, "pref_key_long_last_power_connected_receiver", Long.valueOf(l));
}
else
{
  Pref.save(paramContext, "pref_key_long_last_power_connected_receiver", Long.valueOf(l));
  paramIntent = new android/content/Intent;
  paramIntent.<init>(paramContext.getApplicationContext(), ScreenLockService.class);
  paramIntent.setAction("com.iconnect.app.START_FROM_BATTERY_CHARGE");
  paramContext.startService(paramIntent);
}
```

(그림 6) 배터리의 충전을 감지하는 서비스

```
long l = ((Long)Pref.load(getApplicationContext(), "pr
if (Calendar.getInstance().getTimeInMillis() - l <= 60
{
  Cons.log("tag", "아직 갱신 시간은 안됐음");
}
else
{
  this.mLockView.checkLockScreenAd();
  this.mLockView.updateClockTimeInfo();
  this.mLockView.refreshWeatherInfo();
  this.mLockView.setVisibility(0);
  this.mLockView.startAdLoad();
  if (this.mStatusBarView != null) {
    this.mStatusBarView.setVisibility(0);
  }
}
```

(그림 7) 광고 인젝션을 수행하는 서비스

3.3. 모바일 광고 인젝션 앱의 특징 분석

악성 광고 인젝션 앱에는 주로 다른 앱 위에 광고를 띄우기 위해 ‘android.permission.SYSTEM_ALERT_WINDOW’ 권한과 ‘android.permission.REORDER_TASK‘ 권한을 사용했다. 또한, 광고 서비스를 계속 유지하기 위해서 ‘android.permission.RECEIVE_BOOT_COMPLETED’ 권한을 필요로 했다. “모바일 광고 인젝션 사례 연구”[1] 에서는 스마트 폰 사용자의 인터넷 URL 히스토리를 파싱하여 앱 개발사에게 넘겨주는 행위도 확인하였다. 사용자가 방문한 URL 을 획득하기 위해서 ‘com.android.browser.permission.READ_HISTORY_BOOKMARKS’ 가 사용되었다. 그리고 앱 설치 시 광고와 관계되는 파일이 생성되는 경우도 있었다.

<표 3> 광고 인젝션 앱 권한 표

권한명	‘C’ 앱	‘P’ 앱	‘V’ 앱	‘K’ 앱
SYSTEM_ALERT_WINDOW	X	O	O	O
REORDER_TASK	O	X	X	X
RECEIVE_BOOT_COMPLETED	O	O	O	X

4. 결론

모바일 환경에서 악성 광고 인젝션은 앱 제공자의 정상적인 광고 수익 창출에 악영향을 미칠뿐더러, 앱 사용자에게는 무분별한 광고의 출력으로 불편함을 주고 광고주에겐 의뢰한 광고가 정상적으로 출력되지 않아 광고 플랫폼에 대한 신뢰를 잃게 한다. 하지만 현재 모바일 환경에서의 광고 인젝션 앱에 대한 연구가 부족하여, 광고 인젝션 앱을 탐지할 수 있는 구체적인 방법론이나 시스템은 존재하지 않는다. 따라서 본 논문에서는 향후 악성 광고 인젝션의 탐지와 모니터링 등에 활용할 수 있는 모바일 광고 인젝션 앱의 특징들을 분석하였다.

가장 큰 모바일 광고 인젝션 앱의 특징 두 가지는 광고 인젝션을 수행하기 위해서 다른 앱에 영향을 줄 수 있는 권한들을 사용한다는 것과 광고 인젝션을 수행하는 서비스를 계속 유지하기 위한 권한을 사용한다는 점이었다. 광고 인젝션 앱들은 다른 앱에 광고를 띄우기 위해 ‘android.permission.SYSTEM_ALERT’와 ‘android.permission.REORDER_TASK’를 사용했으며, 서비스를 유지하기 위해 ‘android.permission.RECEIVE_BOOT_COMPLETED’를 사용했다.

참고문헌

[1] 조상현, 허규, 최현상, 김영갑. "모바일 광고 인젝션 사례 연구." 정보보호학회논문지 27(5), pp. 1049-1058, 2017.

[2] 한국인터넷진흥원, “국내 인터넷 이용환경 현황조사 결과”, 2015.

[3] 오세라, 조상현, 김영갑. "GPC: 모바일 광고 인젝션 앱 탐지를 위한 도구 개발." 예술인문사회융합멀티미디어논문지 7, pp. 881-889, 2017.

[4] 조상현, 최현상, 김영갑. "인터넷 광고 인젝션 유형에 대한 연구." 정보보호학회논문지 27(2), pp. 213-222, 2017.

[5] Genymobile. SAS https://www.genymotion.com/

[6] dex2jar. https://github.com/pxb1988/dex2jar