캡스톤 1 차 보고서

교수님저희에이쁠 2 조

팀명: 교수님저희에이쁠 2 조

팀원:

소프트웨어학과 32207508 안석현

소프트웨어학과 32190393 김다은

소프트웨어학과 32184210 정지헌

발표일: 2023.03.30

머신러닝을 이용한 악성 앱 탐지에 사용할 수 있는 특징정보 식별을 위한 APK 분석

1. APK

: 안드로이드 앱은 기본적으로 .apk라는 파일 확장자 명을 가지며, 이는 Android Package의 줄임말이다. APK는 자바 JAR를 확장한 것이며, JAR 파일은 zip 포맷이다. APK파일은 애플리케이션의 실행파일과 권한을 정의하는 파일, 그 밖에 애플리케이션에 필요한 여러 리소스 파일과 Native Library가 존재한다. 안드로이드 앱 설치 파일 구성을 가지는 zip 알고리즘의 압축 파일이며, 단순 압축 해제 시에는 바이너리 파일 형태를 가지기 때문에 디컴파일을 통해 내부 폴더를 읽을 수 있게 된다. 아래 그림들은 APK 파일을 각각 압축 해제한 경우와 디컴파일한 경우이다.

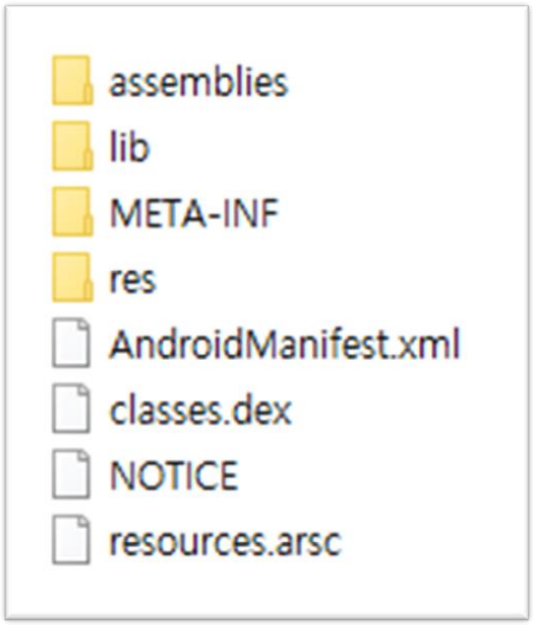

Figure 1 APK 단순 압축 해제 시의 파일 형태

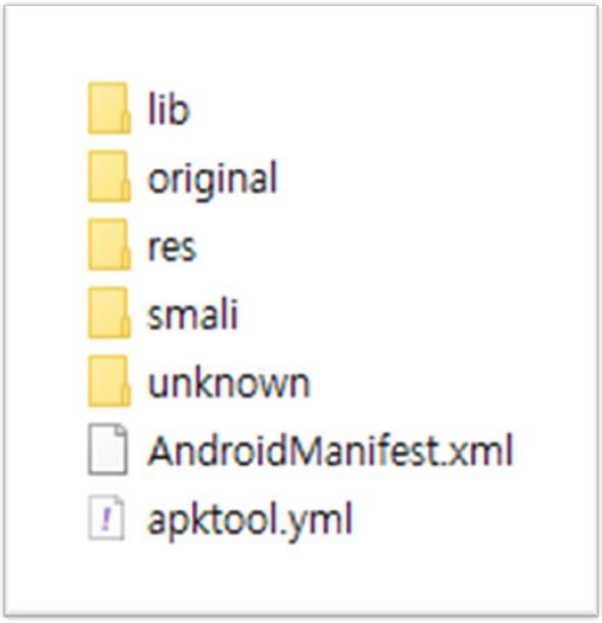

Figure 2 APK 디컴파일(Apktool 이용) 시의 파일 형태

1.1. APK 파일의 내부 구조

- lib: JNI를 통해 Native Library(C/C++ 코드를 사용하게 해주는 도구)를 사용하는

애플리케이션은 lib 폴더에 .so 확장명의 라이브러리가 존재한다.

- META-INF: 자바 설정 관련 파일들이 저장되어 있다. 파일 안쪽에는 MANIFEST.MF라는 사양서 파일과 CERT.SF, CERT.RSA 파일이 들어있으며, java jdk bin 폴더에 있는 서명 도구를 통해 apk 파일을 서명할 수 있다.
- original: 이 안쪽에 있는 파일들은 원본 그대로 존재함
- Res: 애니메이션, 이미지 파일, 추가 정보 등이 저장되어 있는 폴더.
- smali: 악성코드도 난독화가 되어있어 식별이 어렵고, 특징정보로써의 기능을 하기 어렵다는 내용으로
- AndroidManifest.xml: 애플리케이션의 패키지 이름, 최소 실행 버전, 컴포넌트, 필요한 권한 등이 명시되어 있는 파일이다. 안드로이드의 4대 컴포넌트와, 컴포넌트 간의 메시지를 전달하는 intent-filter가 존재한다.
- classes.dex: dalvik byte code로 구성되어 있는 파일로, 안드로이드 애플리케이션의 실행 코드이다. 자바로 개발되는 안드로이드 애플리케이션이 안드로이드 가상 머신인 ART(Android RunTime)에서 작동할 수 있게 우선 javac를 통해 .class 파일을 생성하고, dx(Android 3.1부터 D8로 대체, 현재는 선택적으로 R8 이용가능)를 이용해 .dex 파일을 만든다. (해당 dex 파일과 리소스 파일들은 aapt라는 안드로이드 도구를 이용해 APK 파일 형태로 구성할 수도 있다)
- resources.arcs: 안드로이드 환경에 애플리케이션을 설치하기 위해서는 서명(sign) 과정을 거쳐야 하고 jarsigner가 서명을 담당한다.(Android 11부터는 apksigner로 대체) APK에서 사용되는 문자 등이 들어있는 파일이다. 디컴파일을 거치면 해당 파일은 해석 가능한 Res 파일로 변환된다

2. 특징정보 후보 선정

: 악성앱의 목적이 사용자의 개인정보 탈취이기 때문에, APK를 디컴파일 했을 때 사용자의 개인정보를 얻어올 수 있는 행위와 관련된 정보들을 특징정보로 활용해보고자 한다.

앱의 개발환경과 언어가 모두 다름에도 공통적으로 사용되어야 하고, 앱이 악성임을 구분해낼 수 있는 특징을 가져야 하며, 어느 정도 표준을 따라야 한다.

2.1. API

: classes.dex 파일 안에 들어있는 application Programming Interface의 줄임말인 API는 프로그램 개발 시에 기능을 쉽게 구현하기 위해 미리 구현된 기능의 집합을 의미하는 것으로, 응용 프로그램과 운영체제의 통신을 쉽게 하기 위한 연결 인터페이스이다. API 호출은 프로그램의 행위정보를 반영하기 때문에 특징정보로 사용할 수 있다. "API calls may demonstrate how a specific task is performed." [1]

안드로이드 애플리케이션에 사용되는 공식 API의 개수는 10,000개를 넘으며, 애플리케이션 개발자가 임의로 만들어 사용하는 Third-party API 또한 존재한다. Third-party API의 경우 각각의 애플리케이션에서 정의하기 때문에 개발자에 따라 같은 기능을 수행하는 API여도 다른 이름으로 호출할 수 있으며, 일부 애플리케이션에서만 사용되기 때문에 이 API를 사용하는 것은 sparse data 문제까지 야기할 수 있으므로 공식 API 가 아닌 경우 특징 정보로 활용하기에 무리가 있다. 따라서, API의 선정 기준은 몇 가지 논문을 통하여 참고해서 진행할 예정이다.[2], [3]

2.2. Permission

: permission은 AndroidManifest.xml에 선언되어 있고, 애플리케이션이 기능을 수행할 수 있는 권한을 의미한다. 악성앱이 악성 행위를 하기 위해서는 권한 사용이 필수적이고, AndroidManifest.xml 파일은 APK에 필수적으로 포함된 파일이므로 특징정보로 선정하기에 적절하다고 판단했다. permission 또한, API와 같이 공식 permission과 custom permission으로 나뉘는데, 서드파티 API를 사용하지 않은 이유와 동일한 이유로 공식 permission만 사용하기로 결정했다.

공식 permission 목록은 Android Developer, Aosp, Androguard 등 여러 공신력 있는 사이트에서 제공하고 있지만, 공식 permission 선정 기준과 업데이트 시기가 달라 각기 다른 목록이 제공되고 있으므로 각 사이트를 참고하여 선정해보고자 한다.

2.3. Intent

: intent는 수행할 작업에 대한 추상적인 설명으로, 안드로이드의 4대 컴포넌트인 액티비티, 서비스, 브로드캐스트 리시버, 컨텐트 프로바이더 간의 메시징, 즉 프로세스 간의 통신이 intent를 통해 이루어지며 해당 컴포넌트들을 사용할 수 있게 하기 때문에 안드로이드 프레임워크에서 중요 서비스에 액세스하기 위해 필수적이다. 대상이 정해진 명시적 intent와, 대상이 정해지지 않은 암시적 intent가 존재하는데, 명시적 인텐트는 특정한 형식 없이 java 코드 안에 들어있어 식별과 추출이 어렵다.

해당 그림과 같이 명시적 intent의 경우 구성요소를 정확하게 "com.android.chrome" 으로 정해주지만 암시적 인텐트의 경우 구성요소를 정해주지 않기 때문에 intent 를 가로채거나 위조하여 공격할 수 있다.

Table 2 – Sample code snippet of explicit and implicit Intents.

Explicit Intent	Implicit Intent

String url = "www.yahoo.com";
Intent explicit = new Intent(Intent.ACTION_VIEW);
explicit.setData(Uri.parse(url));
explicit.setPackage("com.android.chrome");
startActivity(explicit)
	String url = "www.yahoo.com";
	Intent implicit = new Intent(Intent.ACTION_VIEW);
	implicit.setData(Uri.parse(url));
	startActivity(implicit);

Figure 3출처: "Androdialysis: Analysis of android intent effectiveness in malware detection."

반면에 암시적 intent는 APK에 필수적으로 포함된 AndroidManifest.xml 파일에 intent-filter의 형태로 들어있고, 형식이 갖추어져 있어 추출하기 쉬운 부분에서 특징정보로써 의미가 있다.

위의 intent와 permission이 특징 정보로써 적합하다는 근거는 Khariwal 등[4]과 Feizollah 등[5]의 연구를 참고했다.

3. 추가 논의 사항

: 악성앱 탐지 앱에 효과적인 기능을 탑재하고자 논의 중에 있다.

(1) 악성으로 탐지된 앱을 경고하고 삭제해주는 기능

(2) 사용자가 검사를 신청한 앱이 악성일 확률이 몇 퍼센트인지 알려주고, 악성 행동을 할 수 있음을 경고하는 기능

(3) 악성으로 탐지된 앱을 대신해 사용할 수 있는 앱을 추천해주는 기능

4. 참고문헌

[1] Salehi, Zahra, Ashkan Sami, and Mahboobe Ghiasi. "Using feature generation from API calls for malware detection." Computer Fraud & Security 2014.9 (2014): 9-18.

[2] D.. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. E. R. T. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket," NDSS. Vol. 14, pp. 23-36, Feb. 2014

[3] Y. Aafer, W. Du, and H. Yin, "Droidapiminer: Mining api-level features for robust malware detection in android," International conference on security and privacy in communication systems, Springer, Cham, pp. 86-103, Sep. 2013.

[4] Khariwal, Kartik, Jatin Singh, and Anshul Arora. "IPDroid: Android malware detection using intents and permissions." 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2020.

[5] Feizollah, Ali, et al. "Androdialysis: Analysis of android intent effectiveness in malware detection." computers & security 65 (2017): 121-134.