

スマートフォン上のアプリケーションにおける利用者情報の  
取扱いに係る技術的検証等の諸問題に係る実証調査研究

## 静的解析エンジンインターフェース仕様書

平成29年2月1日



## 目次

<b>1</b>	<b>はじめに.....</b>	<b>1</b>
<b>2</b>	<b>バイトコード解析機能.....</b>	<b>2</b>
2.1	概要.....	2
2.2	入力インターフェース.....	2
2.3	出力.....	3
2.4	ファイル仕様.....	3
2.4.1	解析結果レポート.....	3

## 用語の定義

用語	内容
apk ファイル	Android のアプリケーションパッケージファイル
XML	Extensible Markup Language の略。インターネット上で様々なデータを扱う場合に有効で、多様な情報を「情報の意味」と「情報の内容」に分けてテキストで記述する方法（言語）のこと
静的解析	Android アプリケーションの解析手法で、アプリケーションを実行せずに得られる情報（API: 利用するアプリケーションインターフェース・バイトコード等）を基に利用者情報の送信有無・内容の解析を行うもの
パーミッション	Android プラットフォームのセキュリティ機能であり、アプリケーションが必要とする権限を開発者があらかじめ明示し、インストール時にユーザーがそれを確認することで、インストールの可否を判断できる仕組み

# 1 はじめに

本書は、平成 28 年度の総務省施策である「スマートフォン上のアプリケーションにおける利用者情報の取扱いに係る実証調査研究の請負」の静的解析システムで利用しているパッケージソフト「静的解析エンジン」についてのインターフェース仕様書である。

## 2 バイトコード解析機能

### 2.1 概要

アプリケーションを実行せずに得られる情報を基に解析を行うバイトコード解析ライブラリーを利用し、静的解析を行う。

apk ファイルを入力とし

- パーミッション情報
- 動的な外部クラスの呼び出し有無
- 外部スクリプト・コマンドなどの呼び出し有無
- 利用者情報および送信可否

について取得する。

### 2.2 入カインターフェース

apk ファイルを入力とする。

「図 2.2-1 コマンドライン」「表 2.2-1 コマンドラインパラメータ」に示すコマンドラインで実行する。

```
> java -jar ByteCodeMod.jar apkFilePath PropFilePath ResultReportPath
```

図 2.2-1 コマンドライン

表 2.2-1 コマンドラインパラメータ

項番	パラメータ名	説明	指定例
1	apkFilePath	分析対象となる apk ファイルのパス	./request/SampleApp.apk
2	PropFilePath	バイトコード解析設定ファイルのパス	./conf/BytecodeAnalyze.properties
3	ResultReportPath	実行結果の記載された解析結果レポートのパスおよびファイル名	./report/report.xml

## 2.3 出力

出力は「表 2.3-1 出力値」のファイルである。ファイルは XML 形式でファイル仕様は、「2.4 ファイル仕様」に示す。

表 2.3-1 出力値

項番	出力値	説明	備考
1	解析結果レポート	実行結果成否、情報漏えいの有無とその詳細情報が記載されたファイルを出力する。	

## 2.4 ファイル仕様

### 2.4.1 解析結果レポート

出力ファイルの XML フォーマットを、「表 2.4-1 XML ファイルフォーマット」に示す。

ファイル名：任意

ファイル形式：XML

エンコード：UTF-8

表 2.4-1 XML ファイルフォーマット

項番	要素名	要素数	データ型	説明
1	Report	1	-	
2	execinfo	1	-	
3	result	1	文字列	・ Success ・ Error ・ Timeout 解析処理が成功したかどうか。 Timeout は制限時間を超過した場合の結果
4	riskCount	1	数値	バイトコード解析した

					結果、リスク有りとは判断されたフロー数
5	apkInfo		0or1	-	
6		apkFileName	1	文字列	apk ファイル名
7		apkHash	1	文字列	apk ファイルを一意に示す ID。SHA256 ハッシュ
8		apkVersion	1	文字列	バージョン
9		apkDate	1	文字列	apk ファイルの生成日付
10		activity	0 以上	文字列	AndroidManifest.xml で定義された activity 要素の android:name 属性
11		permission	0 以上	文字列	AndroidManifest.xml で定義された uses-permission 要素の android:name 属性
12		classLoaderExecute	0 以上	文字列	動的クラスローダーを実行しているメソッド名
13		scriptExecute	0 以上	文字列	内部で Scri



					pt を実行しているメソッド名
14		parentId	1	数値	バージョン違い apk をグループ化する ID
15		libraryInfo	0 以上		情報収集/外部モジュール情報
16		libName	1	文字列	ライブラリー名
17		devName	1	文字列	開発社名
18		moduleName	1	文字列	モジュール名
19	riskInfo		0or1	-	
20		risk	0 以上	-	
21		level	1	数値	1:リスク中 2:リスク大
22		startClass	1	文字列	始点となるメソッド
23		personalDataAccess	0or1	文字列	端末情報取得メソッド名
24		personalDataAccessExecute	0or1	文字列	端末情報取得を行った apk 内のメソッド
25		leakMethod	0or1	文字列	NW 通信メソッド名
26		leakMethodExecute	0or1	文字列	NW 通信を行った apk 内のメソッド

「図 2.4-1 XML サンプル」に XML のサンプルを示す。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Report xmlns="urn:ntts-bytecode-1.0">
  <execInfo>
    <result>00</result>
    <riskCount>1</riskCount>
  </execInfo>
  <apkInfo>
    <apkFileName>/var/data/request/56/*****.1.7.4.apk</apkFileName>
    <apkHash>B86C1AB7664EA216A38B7438DD7D44B1D92550DDBC3A93DC8CFEA4F9D16C951F</apkHash>
    <apkVersion>1.7.4</apkVersion>
    <apkDate>1979-11-30T00:00:00+0900</apkDate>
    <activity>tw.mobileapp.qrcode.banner.CaptureActivity</activity>
    <activity>tw.mobileapp.qrcode.banner.CaptureDecodeActivity</activity>
    <activity>tw.mobileapp.qrcode.banner.QRIntroduction</activity>
    <activity>tw.mobileapp.qrcode.banner.HistoryView</activity>
    <activity>com.google.android.gms.ads.AdActivity</activity>
    <activity>com.google.android.gms.ads.purchase.InAppPurchaseActivity</activity>
    <activity>com.google.android.gms.common.api.GoogleApiActivity</activity>
    <permission>android.permission.CAMERA</permission>
    <permission>android.permission.CHANGE_WIFI_STATE</permission>
    <permission>android.permission.READ_EXTERNAL_STORAGE</permission>
    <permission>android.permission.ACCESS_NETWORK_STATE</permission>
    <permission>android.permission.INTERNET</permission>
    <parentId>1591</parentId>
  </apkInfo>
  <riskInfo>
    <risk>
      <level>1</level>
      <startClass>void com.google.android.gms.ads.internal.overlay.zzv.onSensorChanged(android.hardware.SensorEvent)</startClass>
      <personalDataAccess>android.hardware.SensorEvent.values</personalDataAccess>
    </risk>
  </riskInfo>
</Report>
```

```
<personalDataAccessExecute>void com.google.android.gms.ads.internal.  
overlay.zzv.onSensorChanged(android.hardware.SensorEvent)</personalDataAccessExe  
cute>  
  </risk>  
</riskInfo>  
<evalInfo>  
  <result>  
    <CASE001>0</CASE001>  
    <CASE002>0</CASE002>  
    <CASE003>0</CASE003>  
    <CASE004>0</CASE004>  
    <CASE005>0</CASE005>  
    <CASE006>0</CASE006>  
    <CASE007>0</CASE007>  
    <CASE008>0</CASE008>  
    <CASE009>0</CASE009>  
    <CASE010>0</CASE010>  
    <CASE011>0</CASE011>  
    <CASE012>0</CASE012>  
    <CASE013>0</CASE013>  
  </result>  
</evalInfo>  
</Report>
```

図 2.4-1 XML サンプル