AndroidにおけるMultiple APKの脅威の調査と検証

関岡 好史[1]　今井 宏謙[1]　吉田 奏絵[1]　金岡 晃[1]

概要：Androidアプリケーションを提供しているGooglePlayでは、設定の異なる端末に対して別々のAPKを公開するMultiple APKの機能が備わっている。通常Androidアプリケーションはバージョンや画面サイズなど各端末の設定に対応するために代替リソースを含めているが、この代替リソースを含めることによりAPKがGooglePlayで公開するにあたり制限されているファイルサイズの100MBを超えてしまう場合がある。これを避けるためにGooglePlayではMultiple APKの機能を提供している。しかし、Multiple APKはその性質上異なるAPKを一つのアプリに含められる恐れがある。また、APKを検査、解析をするサービスや研究では、対象アプリケーションがMultiple APKの場合、それらサービスや研究では1つのAPKだけしか見ていない可能性があり、検査サービスを避けるような悪意のあるAPKを故意に作られる恐れがある。本研究ではMultiple APKを取り巻く環境の網羅的な調査や、Multiple APKにおける脅威がどれほど存在するか検証を行った。

Survey and verification of Multiple APK threat on Android

YOSHIFUMI SEKIOKA[1]　HIRONORI IMAI[1]　KANAE YOSHIDA[1]　AKIRA KANAOKA[1]

1. はじめに

Androidアプリケーション(以後、アプリと記す)を提供しているGooglePlayでは、設定の異なる端末に対して別々のAPKを公開するMultiple APKの機能が備わっている。通常Androidアプリはバージョンや画面サイズなど各端末の設定に対応するために代替リソースを含めているが、この代替リソースを含めることによりAPKがGooglePlayで公開するにあたり制限されているファイルサイズの100MBを超えてしまう場合がある。これを避けるためにGooglePlayでは各端末に対応した別々のAPKを1つのアプリとして公開するMultiple APKを提供している。

実際にMultiple APKとして公開するにはいくつか規則がある。しかし、それは同じパッケージ名、同じ署名鍵と証明書であることが必要である。複数のAPKがそれぞれどの端末設定に対応しているかはAndroidManifest.xml内にある要素「API レベルの宣言」「画面サイズの宣言」「サポートするOpenGL-ESの宣言」から判断されるが、その他の要素については触れていない。このことからそれぞれのAPKで違う機能の実装や、異なるパーミッションが設定されていたとしても1つのアプリとして公開できる可能性があり、場合によっては悪質なAPKをMultiple APK内に含められる恐れが考えられる。検査サービスや研究では何らかの目的によりAPKを収集することがあるが、そこで利用されるダウンローダやクローラがMultiple APKを利用しているアプリをダウンロードする場合、Multiple APKに含まれる全てのAPKがダウンロードされるのかは定かではない。もし検査サービスがそのアプリの1面しか見ていない場合、Multiple APKに含めた悪意のあるAPKを検知できない恐れがある。著者の知る限り、これまでそういった研究においてMultiple APKへの言及や対応を謳った研究は存在していなかった。

そこで本研究ではMultiple APKの構造を調べ、実際にさまざまなケースのアプリを作成、GooglePlayへアップロード、またそれらを端末にインストールすることでGooglePlayでのMultipleAPKへの取り扱いや端末上でのMultipleAPKの挙動、現在のGooglePlay上におけるMultiple APKの利用率などMultipleAPKを取り巻く環境を網羅的に調査し、MultipleAPKにおける脅威がどれほ

[1] 東邦大学
Toho University

ど存在するかを検証する。

なお、本論文の構成は第 2 章に関連する論文の紹介、第 3 章で Multiple APK の調査、第 4 章で調査の際に発見した脅威の検証、第 5 章に検証結果、第 6 章に今後の課題、第 7 章にまとめといった構成になっている。

2. 参考文献

先述したとおり、我々の知る限りこれまでの研究や実装において Multiple APK の存在を考慮して Google Play の分析や APK 収集が行われた研究やソフトウェアはない。ここでは、いくつか関連する研究を紹介する。

Viennot らが提案した PLAYDRONE[1] では、大規模なクローリングを Google Play に対して行い 100 万を超える APK を収集したが、その収集ツールにおいてはデバイスを T-mobile Galaxy Nexus デバイスから得られた情報を基にクロールしたと言及されており、Multiple APK に対応したアプリではその一部分の APK しか取得できていない可能性が考えられる。

Allix らが提案した AndroZoo[2] では複数のマーケットに対するクローリングを行い、2017 年 5 月時点で 500 万を超える APK を収集し公開をしている。Google Play 用のクローラにおいて Multiple APK に関した記述はなく、ここでも一部分の APK しか取得できていない可能性が考えられる。

Android アプリケーションの静的解析を行う研究は数多く行われており、そこでは Android アプリケーション自身が入力とされて分析が行われるが、その入力されるアプリケーションについて Multiple APK を考慮した言及があるものは我々の調査では見つからなかった [3], [4], [5], [6], [7]。ここに挙げた論文はいずれも GoogleScholar 上では引用数が 100 を超えるものであるが、それらの論文自体での Multiple APK への言及がされていないだけでなく、それらを参照する論文群においても Multiple APK への言及をしているものは見つからなかった。

3. 脅威の調査

3.1 Multiple APK の仕組みについて

- Multiple APK のフィルタ

Android 端末にはユーザが GooglePlay を利用できるようあらかじめ GooglePlay ストアアプリがインストールされている。ユーザがこのアプリを利用した場合、端末の設定に対応していないアプリは GooglePlay ストア上には表示されない。これはいくつかの条件に基づきデータを選別する機能を持つフィルタによるものであり、GooglePlay 上のフィルタではユーザの端末設定とアプリ内にある AndroidManifest.xml から表示するアプリを選別する。AndroidManifest.xml とは Android アプリのルートディレクトリに存在するファイルであり、パッケージ名の指定や

表 1 Multiple APK のフィルタ
Table 1 Multiple APK filter

端末の設定	AndroidManifest.xml での指定例
OpenGL-ES のバージョン	<uses-feature android: glEsVersion=”0x00020000” android:required=”true”/>
ディスプレイ構成	<supports-screens android:smallScreens=”true” android:normalScreens=”true” android:largeScreens=”true” android:xlargeScreens=”true”/>
API レベル	<uses-sdk minSdkVersion=”8”/>
CPU アーキテクチャ	

API レベルの宣言、パーミッションの宣言、アプリの構成などアプリに関する情報を所持している。Multiple APK に含まれている APK の内どの APK がどの端末設定に対応しているかもまた AndroidManifest.xml から判断される。各 APK がどの端末設定に対応しているか判断する際、GooglePlay が確認する要素は以下の表 1 の通りである。

- Multiple APK の構造

GooglePlay は AndroidManifest.xml を読み取り、そのアプリがユーザの端末に対応しているかを判断する。そのため複数の APK を扱う MultipleAPK の場合、どのアプリがどの端末設定に対応しているかを理解し、どのアプリを表示させるか判断する必要がある。このとき、APK 内における AndroidManifest.xml と実行コード、リソース等の関係性は 2 通りの構成が考えられる。

(1) 図 1 のようにそれぞれの APK は共通の AndroidManifest.xml と異なる実行コードやリソースで構成されている

(2) 図 2 のようにそれぞれの APK は AndroidManifest.xml も異なる構成である
実際にはどちらの構成であるか知るため、GooglePlay にアカウント登録し、実際にアプリを制作することで調査を行った。調査では Multiple APK の要件を満たす簡単な APK を 2 つ作成、これを Multiple APK として公開した。その結果、Multiple APK は後者に挙げた図 2 のような構成であることが分かった。

3.2 Multiple APK の利用状況調査

Multiple APK がどれほど使われているのかを知るために調査を行った。調査のために GooglePlay にアクセスするクローラを作成、これを利用して GooglePlay 上で分けられているいくつかのカテゴリーにおける人気上位 100 位までのアプリを取得、これらの内、何%が Multiple APK を利用しているか調査した。ランキングの種類はいくつか

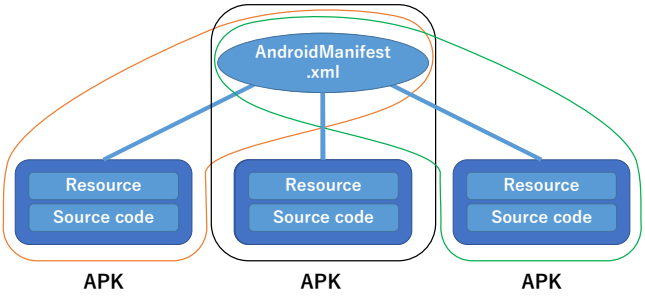

図 1 Multiple APK 構造の仮定
Fig. 1 Multiple APK structure assumption

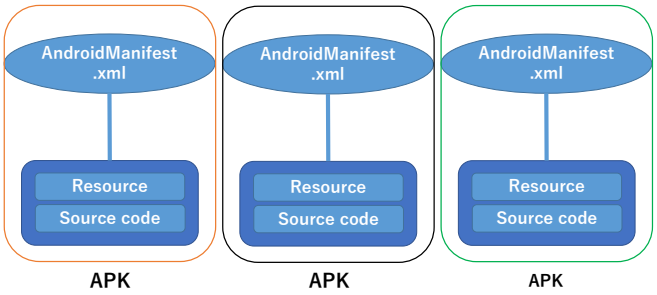

図 2 Multiple APK 実際の構造
Fig. 2 Multiple APK actual structure

表 2 無料トップにおける Multiple APK の割合
Table 2 Percentage of Multiple APK at free top

ジャンル名	アプリ個数	Multiple APK 個数	Multiple APK 割合 (%)
ANDROID_WEAR	98	45	45.9
ENTERTAINMENT	91	9	9.9
PERSONALIZATION	99	11	11.1
GAME	100	0	0.0
COMICS	100	3	3.0
SHOPPING	96	9	9.4
SPORTS	100	5	5.0
SOCIAL	100	12	12.0
TOOLS	93	27	29.0
NEWS_AND_MAGAZINES	100	14	14.0
BUSINESS	94	14	14.9
FINANCE	90	12	13.3
LIFESTYLE	88	10	11.4
LIBRARIES_AND_DEMO	100	7	7.0
MEDICAL	100	1	1.0
MUSIC_AND_AUDIO	100	27	27.0
EDUCATION	100	4	4.0
HEALTH_AND_FITNESS	100	14	14.0
PRODUCTIVITY	100	35	35.0
PHOTOGRAPHY	72	19	26.4
BOOKS_AND_REFERENCE	100	17	17.0
COMMUNICATION	97	28	28.9
WEATHER	100	17	17.0
TRAVEL_AND_LOCAL	100	20	20.0

存在し、本研究では 2017 年 1 月 19 日に「無料トップ」「売上トップ」「新着無料トップ」「急上昇」で順位付けされたアプリの上位 100 位を調査し、以下の表 2〜5 にまとめた。また、カテゴリーや順位付けの方法によっては上位 100 位までないものも存在したため、その場合は取得できた個数の内、何%が Multiple APK を利用しているか算出した。

3.3 Multiple APK の悪用可能性とリスク

調査の結果から Multiple APK に含まれる複数の APK はそれぞれの AndroidManifest.xml を所持していることが分かった。このことから Multiple APK は同じパッケージ名、同じ署名鍵と証明書でありながらそれぞれの APK は独立していることになる。3.1 章に記述したように AndroidManifest.xml ではパッケージ名の指定や API レベルの宣言の他にパーミッションの宣言、アプリの構成などの記述がされている。各 APK がどの端末設定に対応しているか確認する API レベルや画面サイズの宣言が各 APK の AndroidManifest.xml 毎に異なることは認められているが、その他の要素が異なっている場合に公開が可能であるかは定かではない。もし公開することが可能である場合、各 APK 毎にパーミッションの宣言やアプリの構成が異なる APK を 1 つのアプリとして公開できることになる。その場合、特定の端末を対象に許される権限や機能が異なるような悪質な APK を Multiple APK に含めて配信するといった悪用の可能性が考えられる。この場合、Multiple APK に含まれる全ての APK は別々のものであると考えるべきであり、検査サービスによる分析は全ての APK に対して行わなければ安全であるとはいえない。そのためダウンローダやクローラが Multiple APK の一部のみをダウンロードしている場合、それらを用いた検知システムを回避する恐れが存在する上、収集した APK によるマーケット分析等も未完全なものであるといえる。また、GooglePlay 上で掲載されている情報ではアプリを利用したユーザのレビューも載っているが、Multiple APK のレビューにおいては APK 毎に分けられているといったことはなく、各 APK に対するレビューは 1 つのアプリに対するレビューとしてまとめられる。このことから GooglePlay 上に掲載されている情報のみから一部の APK に限り異常があることを認識するのは難しいといえる。

4. 検証用アプリの作成

3 章で挙げた Multiple APK における脅威がどれほど存在するかを検証するため、検証用アプリを作成した。本研究ではそれぞれの APK がどの程度の差異まで Multiple

表 3 売上トップにおける Multiple APK の割合
Table 3 Percentage of Multiple APK at sales top

ジャンル名	アプリ個数	Multiple APK 個数	Multiple APK 割合 (%)
ANDROID_WEAR	73	29	39.7
ENTERTAINMENT	100	7	7.0
PERSONALIZATION	62	10	16.1
GAME	100	1	1.0
COMICS	73	5	6.8
SHOPPING	10	2	20.0
SPORTS	100	3	3.0
SOCIAL	100	5	5.0
TOOLS	100	25	25.0
NEWS_AND_MAGAZINES	86	5	5.8
BUSINESS	100	7	7.0
FINANCE	80	19	23.8
LIFESTYLE	100	7	7.0
LIBRARIES_AND_DEMO	9	1	11.1
MEDICAL	65	1	1.5
MUSIC_AND_AUDIO	100	26	26.0
EDUCATION	100	12	12.0
HEALTH_AND_FITNESS	100	25	25.0
PRODUCTIVITY	100	26	26.0
PHOTOGRAPHY	83	20	24.1
BOOKS_AND_REFERENCE	100	12	12.0
COMMUNICATION	100	19	19.0
WEATHER	61	13	21.3
TRAVEL_AND_LOCAL	100	20	20.0

表 4 新着無料トップにおける Multiple APK の割合
Table 4 Percentage of Multiple APK at Newly Free Top

ジャンル名	アプリ個数	Multiple APK 個数	Multiple APK 割合 (%)
ANDROID_WEAR	2	0	0.0
ENTERTAINMENT	100	0	0.0
PERSONALIZATION	100	0	0.0
GAME	100	1	1.0
COMICS	27	0	0.0
SHOPPING	30	0	0.0
SPORTS	64	0	0.0
SOCIAL	99	0	0.0
TOOLS	76	1	1.3
NEWS_AND_MAGAZINES	28	0	0.0
BUSINESS	50	0	0.0
FINANCE	44	1	2.3
LIFESTYLE	88	0	0.0
LIBRARIES_AND_DEMO	24	0	0.0
MEDICAL	20	0	0.0
MUSIC_AND_AUDIO	100	0	0.0
EDUCATION	100	0	0.0
HEALTH_AND_FITNESS	50	2	4.0
PRODUCTIVITY	87	0	0.0
PHOTOGRAPHY	46	0	0.0
BOOKS_AND_REFERENCE	100	0	0.0
COMMUNICATION	47	0	0.0
WEATHER	42	1	2.4
TRAVEL_AND_LOCAL	28	0	0.0

APK として扱うことが許されるのか、また、どういった点を基準に設定が異なる端末に対し APK を振り分けているのかを把握するために 6 種類の検証用アプリを作成した。

(1) MultipleTest_about_Contact

・このアプリに含まれる APK : APK(APILevel 19 以上), APK(API Level18 以下)
・APK(APILevel 19 以上) パーミッションの宣言 : READ_CONTACTS, WRITE_CONTACTS
・APK(APILevel 18 以下) パーミッションの宣言 : READ_CONTACTS
・目的 : パーミッションが異なっていても GooglePlay 上にアプリの公開が可能であるかテストを行う (APK 間で共通のパーミッション有、APK 間で異なるパーミッション有)
・内容 : 画面の中心にボタンが 2 つ設置されており、電話帳に Test データを登録する機能と電話帳を開いて中身を確認する機能が備わったアプリである。基本的には Multiple APK の要件に従って作られた API レベルの宣言が異なる 2 つの APK だが、片方の APK のみ連絡先の書き込みを行うパーミッション「WRITE_CONTACTS」を宣言していない。そのため、意図的に電話帳に Test データの登録を行うことができないといった不具合を起こしている。

(2) Found_the_MultipleThreat

・このアプリに含まれる APK : APK(APILevel 19 以上), APK(APILevel 18 以下)
・APK(APILevel 19 以上) パーミッションの宣言 :
・APK(APILevel 18 以下) パーミッションの宣言 : READ_CONTACTS
・目的 : 1 同様パーミッションが異なっていても公開が可能であるかテストを行う (共通のパーミッションなし、APK 間で異なるパーミッションあり)
・内容 : 悪質なアプリの一例として連絡先データの抜き取りを行うアプリが挙げられる。このアプリではそのような悪質なアプリを例えて作成した。API レベル 18 以下の端末に限り画面中心のボタンを押すと端末の連絡先データを読み取り画面上に表示させる。

(3) MultipleTest_about_App

・このアプリに含まれる APK : APK(APILevel 19 以

表 5　急上昇における Multiple APK の割合
Table 5　Percentage of Multiple APK in the rapid increase

ジャンル名	アプリ	Multiple APK	
	個数	個数	割合 (%)
ANDROID_WEAR	26	17	65.4
ENTERTAINMENT	96	4	4.2
PERSONALIZATION	50	2	4.0
GAME	100	0	0.0
COMICS	23	0	0.0
SHOPPING	48	4	8.3
SPORTS	17	1	5.9
SOCIAL	71	6	8.5
TOOLS	89	12	13.5
NEWS_AND_MAGAZINES	41	7	17.1
BUSINESS	81	12	14.8
FINANCE	73	9	12.3
LIFESTYLE	80	3	3.8
MEDICAL	35	1	2.9
MUSIC_AND_AUDIO	86	11	12.8
EDUCATION	100	4	4.0
HEALTH_AND_FITNESS	100	11	11.0
PRODUCTIVITY	85	22	25.9
PHOTOGRAPHY	34	1	2.9
BOOKS_AND_REFERENCE	68	5	7.4
COMMUNICATION	25	6	24.0
WEATHER	56	7	12.5
TRAVEL_AND_LOCAL	52	9	17.3

上), APK(APILevel 18 以下)
・APK(APILevel 19 以上) パーミッションの宣言 : READ_CALENDAR, READ_CONTACTS, READ_HISTORY_BOOKMARKS, READ_USER_DICTIONARY, WRITE_CALENDAR, WRITE_CONTACTS, WRITE_HISTORY_BOOKMARKS, WRITE_USER_DICTIONARY
・APK(APILevel 18 以下) パーミッションの宣言 : READ_CONTACTS
・目的 : 1, 2 同様パーミッションが異なっていても公開が可能であるかテストを行う（共通のパーミッションあり（1 つ)、APK 間で異なるパーミッションあり (複数)）
・内容 : 複数のパーミッションの宣言がされているが、権限の必要な機能は一切搭載されておらず、実行中特に異なる部分がないアプリ。片方の APK では連絡先を読み取る「READ_CONTACTS」のみパーミッション宣言されているのに対し、もう片方では連絡先やカレンダーに関する 8 つのパーミッションが宣言されている。

(4) MultipleTest_about_layout

・このアプリに含まれる APK : APK(APILevel 19 以上), APK(APILevel 18 以下)
・目的 : 実行コードやリソースが異なっていても公開が可能であるかテストを行う
・内容 : 3 までのアプリでは AndroidManifest.xml に注目して異なる APK を作成したが、このアプリでは実行コードやリソースに注目し、機能やレイアウトなど実行時に明らかに別物であると分かる APK を 1 つのアプリとして公開することが可能であるかを目的とした。APK(APILevel 19 以上) ではテキストの入力や送信ボタン、ページの遷移等の機能が搭載されているが、APK(APILevel 18 以下) では特に機能を備えてはおらず、画面上には円が描画される全く異なるアプリである。

(5) MultipleTest_about_OpenGL

・このアプリに含まれる APK : APK(OpenGL ES バージョン 1.0), APK(OpenGL ES バージョン 2.0), APK(OpenGL ES バージョン 3.0)
・目的 : 端末のサポートしている OpenGL ES のバージョンによってダウンロードされる APK が変わるかテストを行う
・内容 : 4 までのアプリは API レベルの宣言を変えることで特定の端末を対象にした APK を作成しているが、Multiple APK の用意しているその他の要件でもアプリの公開、及び目当ての端末に APK のインストールが行えるか確認するために作成したアプリ。このアプリでは AndroidManifest.xml の<uses-feature>要素を追加することで端末のサポートしている OpenGL ES のバージョンによってインストールされる APK が異なるよう記述している。図**??**は OpenGL ES バージョンが 3.0 の端末にインストールされる APK のメイン画面である。

(6) MultipleTest_about_Screen

・このアプリに含まれる APK : APK(xlarge), APK(large), APK(normal), APK(small)
・目的 : 端末の画面サイズによってダウンロードされる APK が変わるかテストを行う
・内容 : AndroidManifest.xml において<supports-screens>要素の設定をすることで端末の画面サイズによってインストールされる APK が異なることを目的にしたアプリ。図**??**は画面サイズが large の端末にインストールされる APK のメイン画面である。

図 3 アプリに含まれる 1 つ目の APK の詳細

Fig. 3 Details of the first APK included in the app

図 4 アプリに含まれる 2 つ目の APK の詳細

Fig. 4 Details of the second APK included in the app

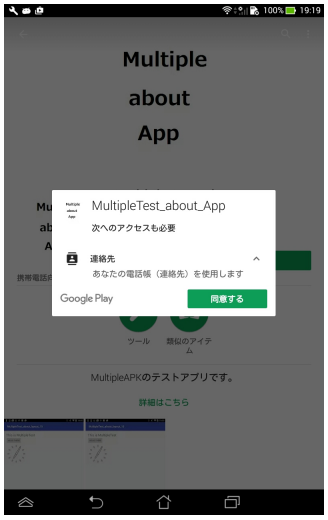

図 5 Fonepad7 に表示された権限の一覧

Fig. 5 List of permissions displayed on Fonepad 7

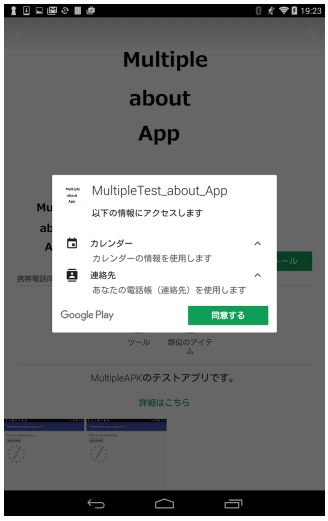

図 6 Nexus7(2013) に表示された権限の一覧

Fig. 6 List of permissions displayed on Nexus7(2013)

5. 検証結果

5.1 検証方法

(1) 4 章で作成した APK が GooglePlay 上にアップロード、公開できるか試みる

(2) 設定が異なる 2 つの端末で GooglePlay 上に公開できたアプリのインストールを行う。本研究では API レベル 19 の Nexus 7(2013) と API レベル 18 の ASUS Fonepad7 を使用する

(3) Web 版、アプリ版の APK ダウンローダー、およびクローラを利用しアプリのダウンロードを試みる。アプリ版の APK ダウンローダは API レベル 19 の Nexus 7(2013) から使用し、クローラの端末設定は Nexus(2012) で行う

5.2 検証結果

表 6 は検証方法 1、及び検証方法 2 の検証結果である。表 6 の通り検証用に作成したアプリは全て GooglePlay 上にアップロード、公開することができた。また、GooglePlay へアップロードした APK はそれぞれ詳細を確認することができ、「MultipleTest_about_Contact」の各 APK の詳細を確認した際、必要な権限と書かれた項目において、権限数が異なっていることが確認できた（図 3、4）。

検証方法 2 においてもそれぞれの端末でアプリのインストールを行うことができ、各端末設定に対応した APK がインストールされていることが確認できた。また、Google-

なお、本研究中に GooglePlay から「MultipleTest_about_Contact」に対して以下のような警告メールが送られてきた。

> 警告の詳細: Google Play では、ユーザーや端末に関する機密情報を要求する、または取り扱うアプリの場合、デベロッパーは有効なプライバシーポリシーを提供する必要があります。当該のアプリは、個人情報または機密情報に関わる権限（カメラ、マイク、アカウント、連絡先、スマートフォンなど） またはユーザー データを要求していますが、有効なプライバシーポリシーが確認できませんでした。

Play ストアからアプリのインストールを試みる際、そのアプリが必要とする権限の一覧が表示されたが、インストールされる APK によって必要な権限が異なっていることが確認された（図 5、6）。

検証方法 3 ではそれぞれ表 7 に記載されている APK がダウンロードされた。Web 版ダウンローダではパソコン上から利用したため端末の設定なしにダウンロードしたが、ダウンロードされた APK は一部のみであった。クローラでは「MultipleTest_about_Contact」のみダウンロードを行うことができなかった。

5.3 検証結果の考察

- Multiple APK に含まれる APK について

検証結果から AndroidManifest.xml においてはパーミッションの宣言やアプリの構成などの要素や実行コード、リソース等が異なっていても 1 つのアプリとして公開できることが分かった。このことから GooglePlay 上に公開されている Multiple APK を利用したアプリは、様々な端末に対し、異なる APK を提供していることになる。悪意のあるユーザが Multiple APK を利用することで特定の端末にのみ他の APK とは全く異なる機能をもつ悪意のある APK

表 6　検証方法 1, 2 の検証結果
Table 6　Verification result of verification method 1, 2

ID	アプリ名	GooglePlay へのアップロード	Nexu7(2013) へのインストール	Fonepad7 へのインストール
1	MultipleTest_about_Contact	○	○	○
2	Found_the_MultipleThreat	○	○	○
3	MultipleTest_about_App	○	○	○
4	MultipleTest_about_layout	○	○	○
5	MultipleTest_about_OpenGL	○	○	○
6	MultipleTest_about_Screen	○	○	○

表 7　検証方法 3 の検証結果
Table 7　Verification result of verification method 3

ID	アプリ名	Web 版ダウンローダ	アプリ版ダウンローダ (Nexus7(2013))	クローラ (Nexus7(2012))
1	MultipleTest_about_Contact	APK(API19 以上)	APK(API19 以上)	×
2	Found_the_MultipleThreat	APK(API19 以上)	APK(API19 以上)	APK(API18 以下)
3	MultipleTest_about_App	APK(API19 以上)	APK(API19 以上)	APK(API18 以下)
4	MultipleTest_about_layout	APK(API19 以上)	APK(API19 以上)	APK(API18 以下)
5	MultipleTest_about_OpenGL	APK(OpenGL ES 3.0)	APK(OpenGL ES 3.0)	APK(OpenGL ES 2.0)
6	MultipleTest_about_Screen	APK(normal)	APK(large)	APK(large)

をインストールさせることもまた可能である。

- ダウンローダ、クローラが見る Multiple APK について

検証結果からダウンローダ、クローラのどちらも Multiple APK の一部しか見ていない可能性を示した。このことからダウンローダ、クローラを利用した検査サービスでは Multiple APK を利用したアプリが安全であると判断することはできない。検査サービスに安全と判断されたアプリは、ダウンローダやクローラのダウンロードした APK と同様に Multiple APK の 1 部だけで判断された可能性がある。また、3.2 節の調査結果から Multiple APK は幅広く利用されており、Multiple APK の分析が正しく行えていない以上ダウンローダやクローラを利用したマーケット分析は未完全なものであるといえる。

- MultipleTest_about_Contact について

表 7 にあるように今回利用したクローラでは「MultipleTest_about_Contact」のみインストールすることができなかった。理由としては 3 つ程考えられ、1 つ目はこのアプリに限り、現在はマーケット上に存在していない。2 つ目はクローラの設定に対応していないと判断したため表示されなかった。3 つ目は「MultipleTest_about_Contact」が警告を受けていることが関係している等が考えられる。

- GooglePlay からの警告について

本研究中に GooglePlay から「MultipleTest_about_Contact」に限り警告が送られてきた理由として 3 つ程のパターンが考えられる。1 つ目は GooglePlay がダウンロードし、検査を行ったアプリが「MultipleTest_about_Contact」のみであること。2 つ目は「MultipleTest_about_Contact」のみが警告を受けるに値する問題を抱えていた。3 つ目は GooglePlay が「MultipleTest_about_Contact」からダウンロードした APK のみが警告を受けるに値しており、その他のアプリからダウンロードした APK は安全なものと判断された等が考えられる。また、このような警告が送られてくることから、GooglePlay 側はどのように検査を行っているのか、どのような端末設定でアプリを収集し、検査を行っているのかが把握できる可能性がある。より意図的に検査サービスを避けるよう悪意のある APK を Multiple APK に含めることが可能である。反面、警告の内容から MultipleAPK に対応した検知の 1 つとして使うことも可能であると考えられる。たとえばユーザがアプリのプライバシポリシを閲覧した際に、そこで説明されているパーミッションがユーザのダウンロードした APK では宣言されていないことが発覚し、他の APK とは違う挙動を起こしている可能性が示唆される。これを利用して、そのアプリに対してはより深い調査をするなどの手段が考えられる。

6. 今後の課題

6.1 Multiple APK 対応クローラのアイデア

本研究では Multiple APK が多く利用されている GooglePlay において、APK の分析を行うには通常のクローラでは

不十分であることが分かった。今後の課題として Multiple APK である場合は含まれている全ての APK も収集するようなクローラが必要である。3.2 章では Multiple APK が利用されている割合を算出するため、そのアプリが Multiple APK を利用しているか判断するクローラを用いて調査を行った。このようにクローラ自身がそのアプリが Multiple APK を利用しているか判断することができるため、Multiple APK を利用しているアプリに限り特定のプログラムを組むことができる。また、MultipleAPK は設定する項目が限定されているため、それらの設定を全て抑えたクローラであれば良い。MultipleAPK において設定可能な項目は、API レベルが「ＡＰＩ 1〜25」の 25 種類、OpenGL が「1.0」,「1.1」,「2.0」,「3.0」,「3.1」,「3.2」の 6 種類、画面サイズが「small」「normal」「large」「xlarge」の 4 種類なので、全部で 600 種類のパターンに分けられる。Multiple APK を利用しているアプリに限り、それぞれのパターンに応じた機種であることを AndroidID を変えるなどしてクローラが名乗ることで MultipleAPK に含まれる APK も全て収集できれば Multiple APK に対応したクローラが実現できるのではと考えられる。

Multiple APK 対応のクローラが実現できた場合、これまでに GooglePlay から収集したデータセットを用いた調査研究は改めて見直すことが可能になり、これまでの研究の正しさもまた検証することができる。

6.2 Android バージョン毎における脆弱性の調査

検証結果から特定の端末のみを対象にした悪質な APK が検出されることなく公開することが可能であると分かった。このことから攻撃者は Android 全体で見た脆弱性をついた攻撃ではなく、脆弱性のあった古い Android バージョンのみを狙った攻撃を行う可能性がある。この脅威を可視化するため、各 Android バージョン毎にどういった脆弱性が存在しているか、どういった端末が攻撃の対象となりうるか調査する必要があると考えられる。

7. まとめ

本研究では Multiple APK の構造や Google Play での取り扱い、端末上での挙動、現在の GooglePlay 上における Multiple APK の利用率など Multiple APK にかかわる全体像を把握するため、GooglePlay にアカウント登録し、実際に Multiple APK の作成、アップロードを行うなどして調査を行った。その際にいくつかの脅威となるうる問題点が挙げられ、これらを検証するために検証用の Multiple APK を 6 種類作成した。検証の主な目的は 3 つあり、1 つ目は様々な点においてどの程度異なる APK を Multiple APK として公開することが可能であるか。2 つ目はどの程度設定の異なる端末に対して別々の APK を提供することができるか把握すること。3 つ目はダウンローダやクローラが Multiple APK を利用したアプリをダウンロードした際にどうなるかを理解することである。検証の結果、本研究で作成した 6 種類のアプリは全て公開することができ、この結果からそれぞれ宣言するパーミッションが異なっていたり、宣言する数自体が異なるアプリでも公開がすることが可能であり、アプリの実行時に明らかに別物であると分かる APK 同士でも 1 つのアプリとして公開することが可能であると分かった。また、これらのアプリを 2 つの端末を用いて GooglePlay からインストールしたところ、各端末の設定に対応した APK がそれぞれ正常にインストールされたことが確認できた。次に Web 版、アプリ版のダウンローダとクローラを利用してこれらのアプリをダウンロードしたところ、どのダウンローダやクローラもダウンロードしたのは Multiple APK に含まれる APK の 1 つであり、APK の取り残しがあることが分かった。以上のことから本研究では現在のダウンローダやクローラでは Multiple APK に含まれる全ての APK を収集することはできず、これらを利用した検査サービスやマーケット分析は未完全であるといえることが分かった。また、Multiple APK には意図的に悪意のある APK を含められる上、検査サービスに引っかからない恐れがあり、脅威であるといえる。今後の課題としてはこれらの問題を解決するため、Multiple APK に対応したクローラを作成する等が考えられる。

参考文献

[1] Nicolas Viennot, Edward Garcia, and Jason Nieh, “A measurement study of google play”, In The 2014 ACM international conference on Measurement and modeling of computer systems (SIGMETRICS ’14), 2014

[2] Kevin Allix, Tegawend F. Bissyand, Jacques Klein, and Yves Le Traon, “AndroZoo: collecting millions of Android apps for the research community”, In Proceedings of the 13th International Conference on Mining Software Repositories (MSR ’16), 2016

[3] Yu Feng, Saswat Anand, Isil Dillig, and Alex Aiken, “Apposcopy: semantics-based detection of Android malware through static analysis”, In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014), 2014

[4] Alexandre Bartel, Jacques Klein, Yves Le Traon, and Martin Monperrus, “Dexpler: converting Android Dalvik bytecode to Jimple for static analysis with Soot”, In Proceedings of the ACM SIGPLAN International Workshop on State of the Art in Java Program analysis (SOAP ’12), 2012

[5] Leonid Batyuk, Markus Herpich, Seyit Ahmet Camtepe, Karsten Raddatz, Aubrey-Derrick Schmidt, and Sahin Albayrak, “Using static analysis for automatic assessment and mitigation of unwanted and malicious activities within Android applications”, In Proceedings of the 2011 6th International Conference on Malicious and Unwanted Software (MALWARE ’11), 2011

[6] Aubrey-Derrick Schmidt, Rainer Bye, Hans-Gunther Schmidt, Jan Clausen, Osman Kiraz, Kamer A. Yksel,

Seyit A. Camtepe, and Sahin Albayrak, "Static analysis of executables for collaborative malware detection on android", In Proceedings of the 2009 IEEE international conference on Communications (ICC'09), 2009

[7] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel, "FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps", SIGPLAN Not. 49, 6, 259-269, 2014