

WebView搭載Androidアプリケーションにおける ユーザへのCookieコントロール機能提供状況の調査

坂本 一仁^{1,a)} 藤本 裕之¹ 松永 昌浩¹

概要: モバイル端末のアプリケーション（以降、アプリ.）では、WebView と呼ばれる組み込みブラウザ機能が利用可能である。WebView の実装によっては、アプリにおいて Web ブラウザと同様の機能を実現可能である。すなわち WebView を搭載したアプリの数に応じて、独自ブラウザが存在することとなるが、そのようなアプリがユーザへ十分な Cookie コントロール機能を提供しているかは不透明な状況である。本稿では、WebView が搭載された Android アプリを大規模に調査し、ユーザへの Cookie コントロール機能の提供状況を明らかにした。17,757 の Android アプリを調査した結果から、14,627 (82.4%) のアプリが WebView を利用し、414 (2.3%) のアプリがメイン機能側の WebView 内で Cookie およびサードパーティ Cookie を許可していた。さらにその内の 7 アプリのみが、Cookie 削除に関するユーザインタフェースを備えていた。

Why Johnny Can't Remove Cookies in Apps: User Privacy Control on Android Applications Using WebView

TAKAHITO SAKAMOTO^{1,a)} HIROYUKI FUJIMOTO¹ MASAHIRO MATSUNAGA¹

Abstract: In the applications (apps) of mobile devices, developers can use the function of the embedded browser, called WebView. Employing WebView makes an app a browser, thus there are many browser-like apps in accordance with the number of apps embedded WebView. However, it is unclear whether these apps have provided users enough with a function of controlling cookies. In this paper, we investigated Android apps embedded WebView on a large scale and found the situation of having provided cookie control functions to users. From our results of analyzing 17,757 of apps, 14,627 (82.4%) of apps using WebView and 414 (2.3%) of apps enable cookie and third-party cookie of WebView in main functions of the apps. Moreover, 7 out of 414 apps had the user interface related to delete cookies.

1. はじめに

ユーザがインターネットの様々なサービスを利用する際、多くのプライバシー問題に直面する。その 1 つにトラッキングの問題があり、今日まで様々な研究が報告されている [1], [2], [3], [4], [5]。例えばサービス事業者は、ユーザの嗜好に沿った広告配信やアクセス解析のために Cookie を利用し、ユーザをトラッキングして履歴情報を常に収集している。一方で、現在の Cookie 仕様ではそのリスクやユーザコントロールに言及しており [6]、ほとんどの主要ブ

ラウザにおいてプライバシー保護機能として、Cookie 削除・拒否の機能が実装されている。

これまでのトラッキング問題に対する研究の多くは、PC のブラウザを対象にしている。しかしながら、今日のユーザのインターネット利用は、PC からスマートフォン（以降、スマホ.）に代表されるモバイル端末へ移っている [7]。スマホにおけるインターネット利用が PC と異なる点は、多くのユーザがアプリケーション（以降、アプリ.）を起点としてサービスを利用しているところである [8]。スマホアプリでは WebView と呼ばれる組み込みブラウザ機能を実装でき、Cookie を利用可能であり、ブラウザと同様の機能を実現できる [9], [10]。すなわち、WebView を搭載した

¹ セコム株式会社 IS 研究所
Intelligent Systems Laboratory, SECOM CO.,LTD
^{a)} takah-sakamoto@secom.co.jp

アプリはブラウザのような User Agent (UA) [11] と言え、Cookie 仕様 [6] で提供すべき (SHOULD) と言及するように、Cookie 削除・拒否の機能が提供されるべきである。

本稿では、WebView を搭載した Android アプリに焦点を当て、Cookie が利用されている場合にユーザへ Cookie コントロール機能が提供されているか否かを大規模に調査した。我々は、Google Play のアプリランキング 33 カテゴリからそれぞれ最大 600 アプリを選定し、約 18,000 アプリをダウンロードし、調査を行った (3 節)。次に WebView を利用し、かつ Cookie およびサードパーティ Cookie (以降、TPCookie。)の利用を許可しているアプリを静的解析によって発見した (4 節)。さらに、WebView で Cookie を利用しているアプリに対し、アプリの WebView を操作した際の Cookie を観測および分析し、ユーザに対する Cookie 削除・拒否機能の提供状況を明らかにした (5 節)。

本稿の貢献は下記の通りである。

- Google Play におけるアプリランキングを利用し、近年のユーザが広く利用するアプリを調査対象とし、さらに正規のアプリ配信プラットフォームからアプリを取得することで、ユーザの利用実態に即した調査を実施した。
- 17,757 アプリの静的解析を実施し、そのうち 14,627 (82.4%) のアプリが WebView を利用しているという WebView に関する最新の調査結果を示した。
- アプリのメイン機能において WebView を使用し、かつ Cookie および TPCookie を利用しているアプリは 414 (2.3%) であった。
- 414 アプリに対し動的解析を行い、7 アプリのみがユーザへ明示的に Cookie 削除機能のユーザインタフェース (UI) を提供していることを発見した。
- アプリのカテゴリによって、WebView 内 Cookie 許可に違いが見られた。本稿の調査では DATING (出会い) カテゴリが最も WebView 内で Cookie および TPCookie を許可しているものが多く存在した。
- Cookie 削除・拒否の UI を含め、アプリにおけるデータ管理の UI は様々な種類が存在することを発見した。

2. 背景と調査目的

2.1 スマートフォンでのインターネット利用

2018 年、個人のインターネット利用はスマホがパソコン (パーソナルコンピュータ/PC) を上回っており [7]、スマホのユーザは、アプリを起点としてインターネットの様々なサービスを利用することが多い [8]。例えば、ユーザはアプリ配信プラットフォーム (Google Play[12], App Store[13] 等) からスマホアプリをダウンロード・インストールし、アプリを起動してインターネットの様々なサービス (facebook, Twitter 等) を利用することができる。



図 1 WebView 搭載 Android アプリの例

Fig. 1 Sample of Android app using WebView.

スマホにおいても PC と同様に、ブラウザを利用してインターネット上の Web サイトへアクセスすることができるが、アプリでは WebView と呼ばれる組み込みブラウザ機能をアプリ内に実装し、アプリ内で Web サイトへアクセスし、HTML を描画させることが可能である [9], [10]。例えば、図 1 に示すようにあるニュースアプリでは、ニュース記事を WebView を利用して表示していることが、UI Automator Viewer[14] によって確認できる。このようなアプリは、複数の異なるニュースサイトを表示できる、ブラウザ機能を有する UA と考えられる。すなわち、WebView を搭載したアプリの数に応じて独自のブラウザが存在することとなり、今日のユーザはそのような無数の独自ブラウザを介して、インターネットの様々な Web サイトを利用している。

2.2 トラッキング問題

ユーザがインターネットの様々なサービスを利用する際のプライバシー問題の 1 つに、トラッキングの問題がある。Cookie を利用したトラッキングは、行動ターゲティング広告やアクセス解析などで広く利用されているが、ユーザの履歴情報が常に事業者へ提供されるというリスクが存在する。トラッキングの状況調査やトラッキング防止ツールの効果測定など、これまで様々な研究がされている [1], [2], [3], [4], [5]。Acer らの調査 [3] では、TPCookie が許可された状態のとき、161 のドメインがユーザの履歴情報の 40% 以上を把握できると報告している。アプリの WebView で Cookie および TPCookie が許可されている場合、WebView における行動が多くの事業者にトラッキングされることとなる。

2.3 Cookie コントロール機能

Cookie は RFC6265[6] において仕様が策定されている。その中で、トラッキングの問題やユーザコントロールに言及しており、ブラウザのような UA はユーザに Cookie

管理（削除等）の機能を提供すべき (SHOULD) , Cookie 拒否の機能を提供すべき (SHOULD) と記載されている。そのため、PC およびスマホのほとんどのブラウザは、これらの機能を提供している。我々は事前調査として、Luo ら [15] がセキュリティ調査を行った 20 のモバイルブラウザにおいて、Cookie コントロール機能が存在するかを調査した。2019 年 7 月時点では、1 ブラウザを除く 19 のブラウザが利用可能であり、全てのブラウザが Cookie 削除機能を提供していた。モバイルブラウザであっても、Cookie の仕様を尊重し、Cookie コントロール機能を提供していることが観測できた。

2.4 本稿の目的と概要

2.1 節および 2.2 節では、WebView を搭載したアプリはブラウザのような UA であり、Cookie および TP Cookie が許可されていると、ユーザはトラッキングの問題に晒されることを述べた。また 2.3 節で述べたように、WebView 搭載アプリがブラウザのように Cookie を取り扱うならば、Cookie の仕様を尊重し、WebView 搭載アプリは Cookie コントロール機能をユーザに提供すべきである。しかしながら、このようなアプリにおける Cookie コントロール機能の提供状況は不透明な状況にある。

そこで本稿では、WebView を搭載した Android アプリに焦点を当て、Cookie および TP Cookie が許可されている場合のユーザへの Cookie コントロール機能提供状況を大規模に調査し、現状を明らかにすることを目的とする。本稿では便宜上、WebView が利用され、かつ Cookie および TP Cookie が許可されたアプリを **UA アプリ** と呼ぶ。

我々は、Google Play のアプリランキング 33 カテゴリを利用し、それぞれのカテゴリから最大 600 アプリを選定することで、現状ユーザに広く利用されている約 18,000 アプリを調査対象とした。アプリの取得に関しては 3 節で述べる。さらに取得したアプリに対し、UA アプリか否かを静的解析によって調査し、ブラウザのような UA アプリを抽出した。静的解析手法と結果は 4 節で述べる。そして、UA アプリにおいて Cookie コントロール機能の提供有無を動的解析によって調査した。動的解析手法と結果は 5 節で述べる。

3. Android アプリの取得

本稿では、ユーザに広く利用されているアプリを調査し、最新の結果を報告することを目指す。そのため、過去の Android アプリのデータセット等は利用せず、正規のアプリ配信プラットフォーム (Google Play) に存在するアプリを取得し、分析を行った。

3.1 調査対象アプリの選定

Google Play[12] の日本向け無料トップ (topselling-free)

表 1 静的解析の文字列パターン

Table 1 String patterns of static analysis.

	パターン	説明
$p1$	(WebView)	WebView の作成
$p2$	CookieManager.getInstance	Cookie の利用宣言
$p3$	setAcceptThirdPartyCookies	TP Cookie の許可

のアプリランキングにおける、33 カテゴリを対象に、各カテゴリ最大 600 アプリを選定し、調査対象とした。ここで、ゲーム (GAME) カテゴリはアプリ容量上の問題のため、家族向け (FAMILY) カテゴリは他のカテゴリとの重複が多いため対象外とした。一例として、下記のような URL によって 1 カテゴリ 120 アプリずつ最大 600 アプリのランキングが取得可能である。

https://play.google.com/store/apps/category/SPORTS/collection/topselling_free?hl=ja&num=120

本稿では、Google Play の各カテゴリの Web ページをスクレイピングし、33 カテゴリ合計 18,384 アプリ（重複除く）を調査対象とした。

3.2 調査対象アプリのダウンロード

本稿では、Node.js ライブラリ [16] を利用し、Android 端末の通信をエミュレートして Google Play からアプリのダウンロードを行った。実験に利用した Android 端末は、Google Pixel 3 (Android バージョン 9) である。アプリのダウンロードは 2019 年 4 月から 6 月の期間で実施した。アプリのダウンロードを実施した結果、587 アプリは端末との互換性の問題で取得できなかったため、最終的には 17,797 アプリの取得に成功した。

4. UA アプリの調査: 静的解析

本稿では、取得に成功した 17,797 アプリのうち、Google Play のアプリ説明において「リバース・エンジニアリング禁止」等の文言がある 29 アプリを除く、17,768 アプリに対して UA アプリの静的解析調査を実施した。

4.1 静的解析手法

Android アプリは *.apk という zip ファイル形式であり、unzip コマンド等で展開が可能である。本稿では JADX[17] を利用し、apk 内の class*.dex ファイルを *.java ファイルにデコンパイルして調査を実施した。UA アプリとして、WebView の利用、Cookie の利用、TP Cookie の許可の 3 つを検出する必要がある。表 1 に上記それぞれを特定するためのクラスおよびメソッド利用の文字列パターン $\{p1, p2, p3\}$ を示す。静的解析では、各アプリの *.java において表 1 の $\{p1, p2, p3\}$ をそれぞれ抽出した。また、Android アプリのメイン機能に関するソースコードは一般的にパッケージ名に沿ったディレクトリ構造下に配置されている。例え

表 2 静的解析の結果 ($N = 17,757$)
Table 2 Results of static analysis ($N = 17,757$).

	main $\cup$ other	main	other	main $\cap$ other
WebView	14,627 (82.4%)	6,513 (36.7%)	12,361 (69.6%)	4,247 (23.9%)
WebView + Cookie	11,056 (62.3%)	1,827 (10.3%)	10,282 (57.9%)	1,053 (5.9%)
UA アプリ (WebView + Cookie + TPCookie)	4,034 (22.7%)	414 (2.3%)	3,749 (21.1%)	129 (0.7%)

アルゴリズム 1 クラスおよびメソッド呼び出しの抽出

```

1:  $appName \leftarrow \text{"com.example.app"}$ 
2:  $mainDir \leftarrow \text{"com/example/app"}$ 
3:  $patterns \leftarrow \{p1, p2, p3\}$  ▷ 表 1 参照
4:  $fileList \leftarrow [\text{all java files in this app}]$ 
5:  $output \leftarrow []$ 
6: for all  $file \in fileList$  do
7:   for all  $line \in file$  do
8:     for all  $p \in patterns$  do
9:       if  $p \in line$  then
10:        if  $file.startswith(mainDir)$  then
11:           $output.append([file, line, p, main])$ 
12:        else
13:           $output.append([file, line, p, other])$ 

```

ば、アプリのパッケージ名が“com.example.app”であれば、メイン機能は“com/example/app/MainActivity.java”に格納されている。アプリにおけるクラスおよびメソッド呼び出しの抽出処理を、アルゴリズム 1 に示す。

アルゴリズム 1 において、main と分類されれば、アプリのメイン機能側で WebView 等が利用されていること、または other と分類されれば、広告ライブラリのようなサードパーティライブラリ (TPL) で WebView 等が利用されていることを推定できる。

4.2 静的解析結果

静的解析を実施した 17,768 アプリのうち、11 アプリは zip ファイルの破損等でデコンパイルに失敗した。本稿では、最終的に 17,757 アプリの静的解析に成功した。

4.2.1 UA アプリの結果

解析結果を表 2 に示す。main および other はアルゴリズム 1 に従い、アプリのメイン機能側で発見したか否かを示している。main $\cup$ other は、アプリのソースコードにおいて少なくとも 1 つ以上発見されたということを示している。結果から、我々は 82.4% のアプリが WebView を利用していることを発見した。そして、22.7% のアプリが Cookie および TPCookie が許可されている UA アプリであり、2.3% にあたる 414 アプリがメイン機能側で WebView を利用し、Cookie および TPCookie を許可していた。

4.2.2 カテゴリ別 UA アプリの結果

表 2 の main $\cup$ other 列に関して、カテゴリ毎に分割した結果を図 2 に示す。図 2 では、各カテゴリの調査アプリ総数 (最大 600 アプリ) に対する WebView、WebView + Cookie、および UA アプリ (WebView + Cookie

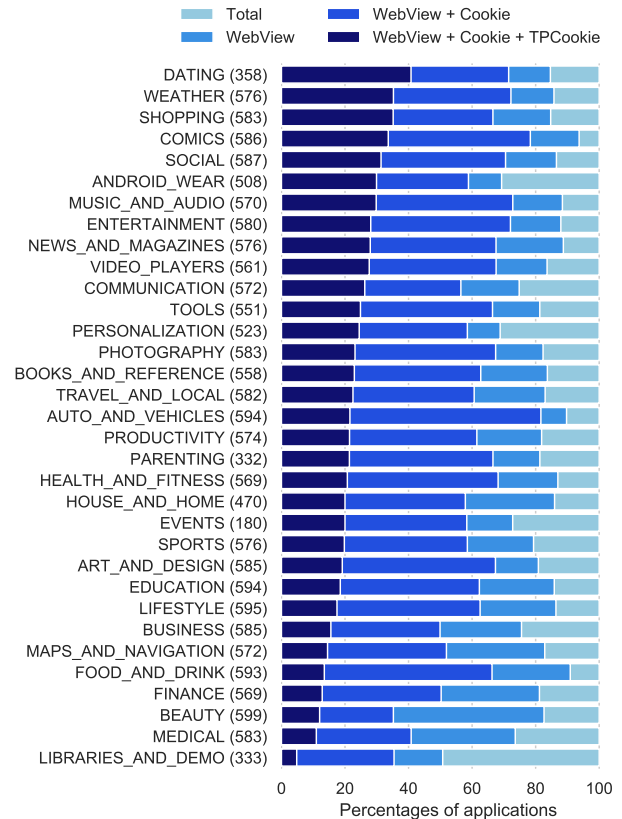


図 2 カテゴリ毎の UA アプリ (WebView + Cookie + TPCookie) の割合。() 内の数値は各カテゴリのアプリ総数である。

Fig. 2 Percentages of UA apps each category.

+ TPCookie) の割合を示し、UA アプリの割合が大きいカテゴリ順に並べている。最も UA アプリの割合が大きいカテゴリは DATING (出会い) の 40.8% であり、続いて WEATHER (天気) 35.2%, SHOPPING (ショッピング) 35.2%, COMICS (マンガ) 33.6%, SOCIAL (ソーシャルネットワーク) 31.3% となった。最も UA アプリの割合が小さいカテゴリは LIBRARIES_AND_DEMO (ライブラリ & デモ) の 4.8% であった。

また、最も WebView が利用されている割合が多いカテゴリは、COMICS (マンガ) の 93.7% であり、最も Cookie が利用されているカテゴリは AUTO_AND_VEHICLES (自動車) の 81.6% であった。

5. Cookie コントロール機能の調査: 動的解析

静的解析の結果から、414 アプリがメイン機能側で Web-

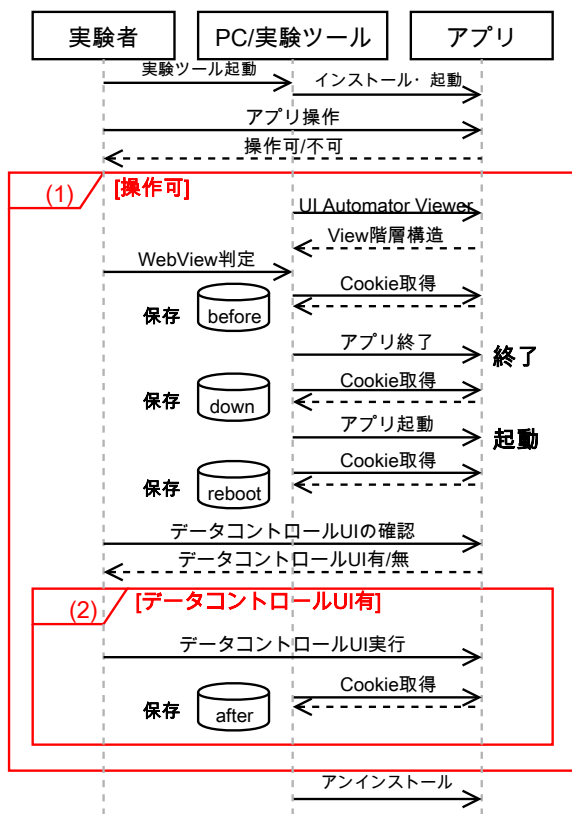


図 3 動的解析の手順

Fig. 3 Procedure of dynamic analysis.

View を利用し、Cookie および TPCCookie を許可している UA アプリであった。動的解析では、これら 414 のアプリに対し Cookie コントロール機能の有無を調査した。Cookie コントロール機能としては Cookie 削除機能と Cookie 拒否機能に関して調査を行う。

5.1 動的解析手法

各アプリにおける調査手順を図 3 に示す。我々は、Android Debug Bridge (ADB) [18] を元に実験ツールを作成し、調査を半自動化した。実験ツールでは主にアプリのインストール、起動、終了、アンインストール、Cookie データベース (DB) の取得を行う。実験者は 1 名であり、PC は Ubuntu 18.04、スマホ端末は Google Pixel 3 を使用した。

実験者は、まず PC の実験ツールを用いてアプリをインストールし、起動する。アプリの操作にサインアップあるいはログインが必要な場合、かつサインアップが可能な場合には、実験者はサインアップあるいはログイン操作を行う。実験者がアプリを操作できない場合は、解析を終了し、実験ツールを用いてアプリをアンインストールする。

アプリの操作が可能な場合、図 3 (1) の調査を実施する。図 3 (1) の調査では主に WebView の判定、データコントロール UI の確認、Cookie の取得を行う。以降、それぞれの調査内容を説明する。

表 3 Cookie の観測ポイント

Table 3 Observation points of cookies.

ポイント	説明
before	WebView を操作した後
down	アプリを終了した後
reboot	アプリを再起動した後
after	データコントロール UI を実行した後

5.1.1 WebView の判定と操作

静的解析によって、アプリのメイン機能に WebView が利用されているものを、動的解析のアプリとして選定しているが、WebView が実在するかどうかを、UI Automator Viewer[14] を用いて確認した。UI Automator Viewer を利用すると、図 1 のようにアプリの View 構造を階層的に可視化することができる。さらに、後述する Cookie DB の取得のため、WebView 内において Web サイトの表示を数回行った。

5.1.2 データコントロール UI の確認

次に、我々はアプリ内に Cookie の削除や拒否に関連するような、データコントロールに関する UI が存在するか確認した。我々は、Cache や Data のような UI 上の表現に関しても、Cookie の操作が行われる可能性を考慮し、広くデータコントロールに類する UI 機能の確認を行った。データコントロール UI の存在が確認された場合、図 3 (2) の調査を実施する。

5.1.3 Cookie の観測

WebView の Cookie は `/data/data/<package_name>/app_webview/Cookies` としてアプリ毎に保存される。Cookie DB は、図 3 および表 3 に示す通り、4 つのポイントで観測する。そして、4 つのポイントにおける Cookie DB の存在および Cookie DB 内のレコード数を測定し、Cookie 利用の有無を判断する。具体的には、before または down 時に DB が存在し、レコード数が 0 以上のときは、Cookie が使用されていると判断する。さらにデータコントロール UI がある場合、after 時のレコード数が 0 になっていれば、UI によって Cookie が削除されると判断する。

5.2 動的解析結果

動的解析の結果を表 4 に示す。414 アプリのうち、179 (43.2%) が操作にログインが必要なアプリであり、12 (2.9%) がブラウザアプリであった。また、90 アプリは起動ができない、サインアップができない等によって、我々の環境では操作ができないアプリであった。WebView が確認された 304 アプリのうち、before, down の Cookie DB を分析することで、264 (86.8%) のアプリにおいて Cookie の使用が確認された。さらに、データコントロール UI が確認されたアプリは 45 であり、明示的に Cookie 拒否の UI が確認されたアプリは 4 であった。

表 4 動的解析の結果 ($N = 414$)Table 4 Results of dynamic analysis ($N = 414$).

	総数	操作可能	WebView 利用	Cookie 利用	データコントロール UI	Cookie 拒否 UI
all	414	324	304	264	45	4
login	179	89	80	67	9	1
browser	12	12	12	11	7	1

表 5 データコントロール UI のラベル付け

Table 5 Labeling user interface related to data control.

ラベル	説明
<i>cookie</i>	明示的に Cookie 削除に類する表現がある
<i>cache</i>	Cookie はないが Cache 削除に類する表現がある
<i>data</i>	データ削除に類する表現がある
<i>history</i>	履歴削除に類する表現がある
<i>auto-clear</i>	起動/終了時削除に類する表現がある

表 6 動的解析結果の詳細

Table 6 Detailed results of dynamic analysis.

	ラベル	UI 確認	Cookie 削除確認
all (414)	<i>cookie</i>	7	6
	<i>cache</i>	22	3
	<i>data</i>	7	3
	<i>history</i>	7	0
	<i>auto-clear</i>	2	0
login (179)	<i>cookie</i>	0	0
	<i>cache</i>	6	0
	<i>data</i>	1	0
	<i>history</i>	3	0
	<i>auto-clear</i>	0	0
browser (12)	<i>cookie</i>	4	4
	<i>cache</i>	0	0
	<i>data</i>	2	1
	<i>history</i>	1	0
	<i>auto-clear</i>	0	0

データコントロール UI については、動的解析を実施する中で様々な表現が観測された。表 5 に示す通り、我々は UI の表現を 5 つのラベルとして分類した。

表 6 はデータコントロール UI の各ラベルにおける結果を示している。414 アプリ全体のうち、7 アプリのみが明示的に Cookie 削除に関する UI を搭載していた。しかしながら、after における Cookie DB を確認した結果、6 アプリが実際に Cookie が削除され、1 アプリは Cookie が削除されなかった。また、*cache* や *data* に関する UI 上の表記であり、Cookie と明示していない場合であっても、いくつかのアプリは Cookie が削除されることが確認された。我々が選定したアプリでは、ブラウザアプリが 12 であったが、4 アプリが明示的に Cookie 削除に関する UI を搭載していた。

6. 議論

6.1 ユーザへの Cookie コントロール機能提供状況

本稿では、17,757 アプリの静的解析を行い、414 アプリがアプリのメイン機能側で WebView を利用し、かつ Cookie および TP Cookie を許可している UA アプリということを発見した。414 アプリの動的解析行った結果、アプリの操作が可能で Cookie の保存を確認できた 264 アプリのうち、Cookie 削除と明示的に表現してる UI を備えたものは 7 アプリ (2.7%)、実際に Cookie が削除されたものは 6 アプリ (2.3%) であった。また、Cookie 拒否機能を備えているアプリは 4 アプリ (1.5%) であった。Cookie の仕様 [6] では、Cookie 削除機能を提供すべき (SHOULD) とされているが、ユーザへの Cookie コントロール機能を提供しているアプリは非常に少ない状況である。

さらに、UI が存在しても正しく Cookie が削除されないアプリや、Cookie という表記ではなく、キャッシュ削除、データ削除等に関する UI であるが、Cookie が削除されているアプリも確認された。ユーザにとってはデータコントロール UI がどのような動作をするものか、不透明な状況であることが確認できる。

なお、静的解析の結果 (表 2) から、*main* と分類されたアプリのメイン機能での WebView 利用よりも、*other* と分類された広告ライブラリ等での WebView 利用の方が多いことが読み取れる。本稿の調査では、*other* のアプリにおける Cookie コントロール機能の有無は調査していないが、広告ライブラリの多くが Cookie を利用していると予想される。

6.2 トラッキングについて

本稿では、WebView における Cookie 利用に焦点を当て、調査を行なった。WebView の Cookie はアプリ単位で Cookie DB に保存され、Cookie を用いて WebView 内での Web サイト閲覧等をトラッキング可能である。しかしながら、アプリは Android 端末固有の識別子 (IMSI) や広告用識別子 (AdvertisingID)、さらに Device Fingerprinting 等 [19] と Cookie による識別を組み合わせることで、WebView やアプリを超えたユーザの行動に対するトラッキングが可能となる。本稿の調査では、Cookie 以外の識別子およびトラッキング技術については対象としていないが、今

後として他のトラッキング技術との組み合わせを調査することが考えられる。

6.3 制限事項

本稿では、Google Play における日本向けのアプリランキングを利用して、調査アプリを選定した。US 向けアプリランキングの利用も想定されるが、US 向けのアプリの多くが、日本では利用不可といったエラーのため、取得できなかった。また iOS アプリの調査は実施していない。日本向け以外の Android アプリ、および iOS アプリの調査は今後の課題である。

AppBrain の統計情報によると、Google Play には 270 万アプリ以上が存在している [20]。本稿では、約 18,000 アプリを調査した。しかしながら、調査アプリの選定にランキングを用いることで、多くのユーザに利用されているアプリを重点的に調査した。

本稿の調査では WebView と Cookie 利用に焦点を当てた。WebView 内での JavaScript の許可や、他の端末、アプリおよび広告識別子 (IMSI, Android_ID, AdversingID 等) については対象としていない。

WebView 以外の組み込みブラウザ機能として Chrome Custom Tabs[21] がある。これは Chrome をアプリ内で利用する機能であり、Cookie は Chrome の保存領域に蓄積される。ユーザは Chrome の Cookie コントロール機能を利用することができる。

6.4 研究倫理について

本稿では 2.4 節で述べた目的を達成するため、クローリングによるアプリの取得、アプリの静的解析、アプリの動的解析を行った。研究倫理への対応として本稿では、コンピュータセキュリティシンポジウム (CSS) 2019 の「サイバーセキュリティ研究における倫理的配慮のためのチェックリスト [22]」を活用した。また、本稿は提出前に社内審査によって、記載内容に法的・倫理的問題がないかの審査を実施している。本稿の各調査における細かな倫理的配慮事項は、下記の通りである。

6.4.1 アプリ取得時の倫理的配慮

本稿の Android アプリ取得は、Google 社が運営する Google Play に対して行った。Google Play に過度な負荷をかけないように、アプリのダウンロードリクエストは、最低でも 1 分の間隔を開け実行した。本調査中に Google Play への接続が遮断されるようなことは起こっていない。

6.4.2 静的解析時の倫理的配慮

静的解析においてアプリのデコンパイル (リバース・エンジニアリング) を行った。2019 年 1 月 1 日に改正された著作権法の一部を改正する法律 [23] では、「プログラムの調査解析を目的としてプログラムの著作物を利用する行為 (いわゆる「リバース・エンジニアリング」)」は認められて

いるが、アプリの説明欄に明示的に「リバース・エンジニアリング禁止」等の文言がある場合、当該アプリの解析は行っていない。

6.4.3 動的解析時の倫理的配慮

動的解析においては、アプリの改変は行わず、Cookie の挙動を観測している。

7. 関連研究

初期の WebView 搭載 Android アプリのセキュリティ調査として、Luo らは WebView とネイティブアプリの処理が連携でき、脆弱性を孕むことを指摘している [15]。その過程で、86%のアプリに WebView が搭載され、5 つの広告モジュールが WebView を利用していることを発見している。また、Chin らの WebView に関する調査では、70.4%のアプリが WebView を利用しており、そのうちの 11%が WebView 内の JavaScript から端末データへアクセスが可能な脆弱なものであると報告している [24]。また、AdSplit[25] では、アプリのメイン機能と広告部分を分離してセキュリティを向上させる提案をしている。AdDroid[26] では、Android OS を改良し、広告専用の API とパーミッションを追加した上で権限の明確化を行っている。Son らは、WebView の JavaScript から Android 端末の外部ストレージへアクセスできることを指摘し、3 種類のアプリと 4 つの広告ライブラリにおいて攻撃が可能なことを実証している [27]。

Android アプリにおける広告や取得情報の調査として、Grace らは 10 万アプリを調査し、約 3 分の 1 のアプリに 1 つ以上の広告ライブラリが導入されていることを報告している [28]。Stevens らは、アプリのトラフィックデータを解析し、広告ライブラリが実際に端末識別子、アプリのパッケージ名や位置情報などを送信していることを明らかにしている [29]。また、ReCon でも同様に、通信内容から端末識別子や個人情報などの漏洩を調査し、検知するシステムを提案している [30]。Book らは、Google 社の広告ライブラリである AdMob に対してターゲティング広告の調査を行い、ほぼ全ての広告がアプリベース、時間ベースのターゲティングを実施し、39%がユーザベースのターゲティングを行っていることを報告している [31]。

8. まとめ

本稿では、Google Play において多くのユーザに利用されている Android アプリを対象に、WebView が搭載されたアプリにおけるユーザへの Cookie コントロール機能の提供状況を調査した。17,757 アプリの静的解析を行った結果、82.4%のアプリが WebView を利用し、アプリのメイン機能において WebView, Cookie, および TP Cookie を利用している UA アプリは 414 であった。414 アプリを動的解析し、UI としての Cookie コントロール機能の有無、

Cookie DB の挙動分析を行った結果、7 アプリのみが明示的に Cookie 削除の UI を提供していた。しかしながら、1 アプリは Cookie が実際には削除されなかった。さらに、4 アプリのみが Cookie 拒否の機能を提供していた。

WebView を利用し、かつ Cookie および TP Cookie を利用しているアプリはブラウザと同様の機能を備え、様々な事業者が Cookie によるトラッキングを実現可能である。Cookie の仕様で言及されているように、プライバシーリスクに配慮し、そのようなアプリは Cookie 削除・拒否の機能をユーザへ提供すべきであるが、現状では多くのアプリが提供していない実態が明らかとなった。今後はユーザのプライバシーを尊重し、多くのアプリがユーザヘデータコントロール機能を提供することが期待される。

参考文献

- [1] Mayer, J. R. and Mitchell, J. C.: Third-Party Web Tracking: Policy and Technology, *2012 IEEE Symposium on Security and Privacy*, pp. 413–427 (2012).
- [2] Balebako, R., Leon, P., Shay, R., Ur, B., Wang, Y. and Cranor, L.: Measuring the effectiveness of privacy tools for limiting behavioral advertising, *Web 2.0 Security and Privacy (W2PS)*, IEEE (2012).
- [3] Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A. and Diaz, C.: The web never forgets: Persistent tracking mechanisms in the wild, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 674–689 (2014).
- [4] Englehardt, S. and Narayanan, A.: Online Tracking: A 1-million-site Measurement and Analysis, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1388–1401 (2016).
- [5] Libert, T., Graves, L. and Nielsen, R. K.: Changes in third-party content on European news websites after GDPR, <https://reutersinstitute.politics.ox.ac.uk/our-research/changes-third-party-content-european-news-websites-after-gdpr/> (2018).
- [6] Internet Engineering Task Force (IETF): 7.2 User Controls - HTTP State Management Mechanism (RFC6265), <https://tools.ietf.org/html/rfc6265#section-7.2> (2011). (参照 2019-08-20).
- [7] 総務省: インターネットの利用状況 - 情報通信白書平成 30 年版, <http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd252120.html> (2018). (参照 2019-08-20).
- [8] nielsen: Digital Trends 2016, https://www.netratings.co.jp/news_release/2017/03/Newsrelease20170309.html (2017).
- [9] Google: WebView - Android Developers, <https://developer.android.com/reference/android/webkit/WebView/>. (参照 2019-08-20).
- [10] Apple: WKWebView - Apple Developer Documentation, <https://developer.apple.com/documentation/webkit/wkwebview/>. (参照 2019-08-20).
- [11] Wikipedia: User agent, https://en.wikipedia.org/wiki/User_agent. (参照 2019-08-20).
- [12] Google: Google Play, <https://play.google.com/store/apps/>. (参照 2019-08-20).
- [13] Apple: App Store, <https://www.apple.com/ios/app-store/>. (参照 2019-08-20).
- [14] Google: UI Automator - Android Developers, <https://developer.android.com/training/testing/ui-automator/>. (参照 2019-08-20).
- [15] Luo, M., Laperdrix, P., Honarmand, N. and Nikiforakis, N.: Time Does Not Heal All Wounds: A Longitudinal Analysis of Security-Mechanism Support in Mobile Browsers, *Network and Distributed System Security Symposium (NDSS)* (2019).
- [16] : gpapi, <https://www.npmjs.com/package/gpapi>. (参照 2019-08-20).
- [17] : JADX, <https://github.com/skylot/jadx>. (参照 2019-08-20).
- [18] Google: Android Debug Bridge - Android Developers, <https://developer.android.com/studio/command-line/adb/>. (参照 2019-08-20).
- [19] Olejnik, L., Englehardt, S. and Narayanan, A.: Battery Status Not Included: Assessing Privacy in Web Standards, *Privacy Enhancing Technologies Symposium* (2017).
- [20] AppBrain: Android and Google Play statistics, <https://www.appbrain.com/stats/>. (参照 2019-08-20).
- [21] Google: Chrome Custom Tabs, <https://developer.chrome.com/multidevice/android/customtabs/>. (参照 2019-08-20).
- [22] コンピュータセキュリティシンポジウム (CSS)2019 研究倫理委員会: サイバーセキュリティ研究における倫理的配慮のためのチェックリスト, https://www.iwsec.org/css/2019/ethics_list.html (2019). (参照 2019-08-20).
- [23] 文化庁: 著作権法の一部を改正する法律 (平成 30 年法律第 30 号) について, http://www.bunka.go.jp/seisaku/chosakuken/hokaisei/h30_hokaisei/. (参照 2019-08-20).
- [24] Chin, E. and Wagner, D.: Bifocals: Analyzing webview vulnerabilities in android applications, *International Workshop on Information Security Applications*, pp. 138–159 (2013).
- [25] Shekhar, S., Dietz, M. and Wallach, D. S.: AdSplit: Separating Smartphone Advertising from Applications, *USENIX Security Symposium* (2012).
- [26] Pearce, P., Felt, A. P., Nunez, G. and Wagner, D.: Android: Privilege separation for applications and advertisers in android, *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 71–72 (2012).
- [27] Son, S., Kim, D. and Shmatikov, V.: What Mobile Ads Know About Mobile Users, *Network and Distributed System Security Symposium (NDSS)* (2016).
- [28] Grace, M. C., Zhou, W., Jiang, X. and Sadeghi, A.-R.: Unsafe exposure analysis of mobile in-app advertisements, *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pp. 101–112 (2012).
- [29] Stevens, R., Gibler, C., Crussell, J., Erickson, J. and Chen, H.: Investigating user privacy in android ad libraries, *Workshop on Mobile Security Technologies (MoST)* (2012).
- [30] Ren, J., Rao, A., Lindorfer, M., Legout, A. and Choffnes, D.: Recon: Revealing and controlling pii leaks in mobile network traffic, *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, pp. 361–374 (2016).
- [31] Book, T. and Wallach, D. S.: An empirical study of mobile ad targeting, *arXiv preprint arXiv:1502.06577* (2015).