TaintDroidを用いた利用者情報送信の動的制御手法の実現と評価

小倉 禎幸　　山内 利宏

岡山大学大学院自然科学研究科
700-8530 岡山県岡山市北区津島中 3-1-1
ogura@swlab.cs.okayama-u.ac.jp　yamauchi@cs.okayama-u.ac.jp

あらまし　近年，Android 端末の普及に伴い，Android を標的とするマルウェアが増加し，マルウェアへの対策が重要視されている．特に，マルウェアによる端末外部への個人情報の漏洩が問題となっている．そこで，文献 [9] において，TaintDroid を利用し，端末外部への個人情報の漏洩を防止する手法の設計について述べた．本稿では，文献 [9] の手法の実現方式と評価について述べる．本手法は，端末内部の個人情報の伝搬を追跡し，個人情報が端末外部に漏洩する際に利用者の判断に従って AP の動作を動的に制御することで，端末外部への個人情報の漏洩を防止する．本手法の Android での実装について述べ，処理のオーバヘッドと追跡精度を評価した結果を報告する．

Implementation and Evaluation of Dynamic Control Method for Sending User Information using TaintDroid

Yoshiyuki Ogura　　Toshihiro Yamauchi

Graduate School of Natural Science and Technology, Okayama University
3-1-1 Tsushima-naka, Kita-ku, Okayama, 700-8530 JAPAN

Abstract　In recent years, Android malware has been increasing, and countermeasures against them has become an important issue. In particular, leakage of user information by malware has become an issue. We previously proposed the design of dynamic control method for sending user information using TaintDroid [9]. This method uses TaintDroid for prevention of leakage of the user information from the Android device. This paper describes the implementation and evaluation results of the method. This method tracks the propagation of user information in a device and dynamically controls the action of an AP when a leak of the user information is detected. Additionally, the method prevents the leakage of the user information from the device. This paper describes the implementation of the method and reports the evaluation results of the detection accuracy and overheads.

1　はじめに

近年，Android[1] 端末が普及し，Google Play[2] などのマーケットにおいて多種多様なアプリケーションプログラム（Application Program：以降，AP と略す）が配布され，利用されている．

しかし，配布されている AP の中には，システムの脆弱性を攻撃して管理者権限を不正に取得するマルウェア，ネットワークや SMS（short message service）経由で外部サーバと通信して料金を発生させるマルウェア，および利用者情報を外部に漏洩するマルウェアなどが発見され [3]，被害を及ぼしている．特に，利用者情報を外部に漏洩するマルウェアによる被害が問題となっている [4]，[5]．このため，端末外部に利用者情報を漏洩させるマルウェアへの対策が重要視されており，この問題に対処するために，利用者情報の漏洩防止に関する様々な手法が研究されている [6]-[12]．しかし，これらの研究には，追跡粒度が粗く誤検知が多い，利用者情報の漏洩に関わった AP の情報を利用者が正確に把握できない，

利用者情報の取得制限による AP の動作の妨害，および利用者の判断による AP の動作制御ができないという問題のいずれかがある．

そこで，文献 [9] では，これらの問題をすべて解決するため，TaintDroid [10] を用いた細粒度の情報追跡による利用者情報送信の動的制御手法の設計について述べた．本稿では，文献 [9] の手法の実現方式と評価について述べる．

なお，本稿において，利用者情報は Android のパーミッションを利用して取得できる情報のうち，端末や利用者に関連する情報のことを指す．

2 既存研究の問題点

文献 [9] で述べたように，利用者情報の漏洩防止に関する様々な手法が研究されている [6]-[12]．しかし，これらの研究には，以下に示すいずれかの問題が存在する．

（問題 1） 追跡粒度が粗く誤検知が多い

（問題 2） 利用者情報の漏洩に関わった AP を利用者は正確に把握不可

（問題 3） 利用者情報の取得制限による AP の動作の妨害

（問題 4） 利用者の判断による AP の動作制御不可

本稿では，これらの問題点を解決するために設計した文献 [9] の手法の実現方式と評価について述べる．

3 設計

3.1 システムへの要求

本章では，文献 [9] の利用者情報の漏洩防止手法の設計について説明する．本手法は，AP による利用者の意図しない利用者情報の漏洩を防止することを目的としている．2 章で示した各問題に対処するために必要な本手法への要求は以下のようになる．

（要件 1） 細粒度で利用者情報の伝搬を追跡できること

（要件 2） 利用者が利用者情報の漏洩に関わった AP を正確に把握できること

（要件 3） AP が利用者情報を端末内部で使用する限り，AP の動作を妨害しないこと

（要件 4） 利用者が AP が端末外部に利用者情報を送信する処理を動的に制御し，必要に応じて利用者情報の漏洩を防止できること

（要望 1） 利用者の負担を少なくすること

（要望 2） 利用者の判断を支援すること

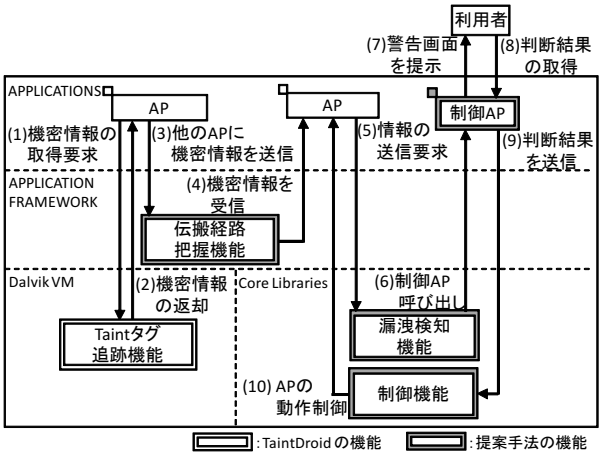

図 1 TaintDroid を用いた利用者情報送信の動的制御手法の全体図

（要望 1）と（要望 2）は，利用者情報の漏洩防止の観点から要件とはならないが，重要な課題である．

3.2 本手法の全体像

本手法の全体像を図 1 に示す．本手法は，一部に TaintDroid の機能を利用した以下の 5 つの機能からなる．

（1）Taint タグ追跡機能（TaintDroid）

AP が利用者情報を取得した際，その取得した利用者情報の伝搬を追跡する．

（2）伝搬経路把握機能

AP 間で利用者情報がやり取りされた場合，その伝搬経路を把握する．

（3）漏洩検知機能

AP がデータを端末外部に送信する動作を検知し，送信されるデータに利用者情報が含まれているか否かを判断する．利用者情報が含まれていた場合は，制御 AP を呼び出す．

（4）制御 AP

利用者情報の端末外部への送信を許可するか否か端末の利用者に尋ね，利用者の判断結果を取得する．

（5）制御機能

制御 AP において取得した利用者の判断結果に従って，AP が利用者情報を端末外部に送信する動作を制御する．

表 1 Taint タグの定義

TAINT_CLEAR	0x00000000
TAINT_LOCATION	0x00000001
TAINT_CONTACTS	0x00000002
TAINT_MIC	0x00000004
TAINT_PHONE_NUMBER	0x00000008
TAINT_LOCATION_GPS	0x00000010
TAINT_LOCATION_NET	0x00000020
TAINT_LOCATION_LAST	0x00000040
TAINT_CAMERA	0x00000080
TAINT_ACCELEROMETER	0x00000100
TAINT_SMS	0x00000200
TAINT_IMEI	0x00000400
TAINT_IMSI	0x00000800
TAINT_ICCID	0x00001000
TAINT_DEVICE_SN	0x00002000
TAINT_ACCOUNT	0x00004000
TAINT_HISTORY	0x00008000

4 実現方式

4.1 Taint タグ追跡機能（TaintDroid）

Taint タグ追跡機能は，TaintDroid により利用者情報の伝搬を細粒度で追跡する機能である．TaintDroid は，以下に示す変数や配列に Taint タグ（表 4.1）を付与している．なお，Taint タグは，Taint.h と Taint.java で定義されている．

(1) メソッドのローカル変数

(2) メソッド引数

(3) クラスの静的フィールド

(4) クラスのインスタンスフィールド

(5) 配列

Taint タグの付与は，具体的には，32bit の変数に隣接して 32bit の TaintTag を付与し，この 2 つの隣接した値を 1 つの 64bit の値として解釈する．例えば，SIMRecords.java 内の SIMRecord クラスの setMsisdnNumber メソッドが呼ばれた際，msisdn に TAINT_PHONE_NUMBER という Taint タグが付加される．

このように，変数や配列に付与された Taint タグの伝搬を追跡することで，AP が取得した利用者情報をを細粒度で追跡できる．

4.2 伝搬経路把握機能

本機能は，AP 間の利用者情報の伝搬経路を把握する機能であり，Taint タグを利用することで実現している．TaintDroid は，Taint タグを伝搬させることで AP が利用する利用者情報を追跡する．しかし，TaintDroid は，利用者情報がどの AP を介して伝搬したかの情報を保持しない．このため，複数の AP が連携し利用者情報を漏洩させた場合，利用者情報の漏洩に関わったすべての AP を特定することができない．

本機能では，インテント処理をフックすることで，AP 間で利用者情報がやり取りされた際に，利用者情報の送信元，送信先の AP のパッケージ名，やり取りされる値，および Taint タグを取得する．また，取得した Taint タグからどの種類の利用者情報がやり取りされた値に含まれているか判断できる．これらの取得した情報をログに出力することで記録する．これにより，ログに出力された情報から利用者情報の伝搬経路を把握できる．また，利用者情報が端末外部に送信される際，記録している情報を利用して，利用者情報の漏洩に関わったすべての AP のパッケージ名を取得できる．利用者情報の伝搬経路に関わる情報は，漏洩検知機能が利用者情報の漏洩を検知した際に，制御 AP に渡され，利用者に提示される．

4.3 漏洩検知機能

本機能は，端末外部への利用者情報の送信を検知し，制御 AP を呼び出す機能であり，Taint タグを利用することで実現している．

AP による利用者情報の漏洩を防止するためには，AP が端末外部に情報を送信する動作を検知する必要がある．このため，端末外部への情報の送信に関わる OpenSSLSocket.java と Posix.java 内にフック箇所を実装することで，その動作の検知を実現した．

本機能では，フック箇所において，AP が端末外部への情報の送信を行う際，送信される値に付加されている Taint タグを取得する．例えば，OpenSSLSocket.java 内では，端末外部に送信されるデータの書き込み処理を行うメソッド（write）が実行された際，その引数として渡された値を取得する．また，取得した値に付与されている Taint タグを取得する．その後，取得した Taint タグから送信を要求した情報に利用者情報が含まれているか否かを識別し，含まれていた場合，利用者情報の漏洩と判断し，取得した Taint タグから利用者情報の種類を特定する．

また，このとき，利用者情報を端末外部に送信する動作を行ってた AP のパッケージ名を取得する．取得したパッケージ名を用いて，伝搬経路把握機能

が保持している情報から取得したパッケージ名に該当する情報を検索する．

該当するパッケージ名が存在しない場合は，利用者情報を端末外部に送信する動作を行った AP のみを利用者情報の漏洩に関わった AP とする．

該当するパッケージ名が存在した場合，伝搬経路把握機能で保持している AP 間でやり取りされた際の利用者情報の送信元，送信先の AP のパッケージ名，やり取りされる値，および Taint タグを取得する．これにより，複数の AP が連携し利用者情報を漏洩させた場合でも利用者情報の漏洩に関わったすべての AP を特定できる．その後，制御 AP を呼び出し，本機能と伝搬経路把握機能で保持している情報を制御 AP に送信する．

4.4 制御 AP

4.4.1 表示する情報

制御 AP は，利用者に警告画面を提示し，利用者の判断結果を取得する AP である．本 AP では，漏洩検知機能から送信された情報を利用者に分かりやすい形式に変更し，警告画面として利用者に提示する．警告画面の例を図 2，図 3，および図 4 に示し，利用者の判断を支援するために利用者に提示する情報を以下に示す．

(1) 利用者情報を外部に送信する AP 名

(2) 利用者情報を取得した AP と利用者情報を外部に送信する AP が別の場合（複数の AP が連携している場合），すべての AP 名

(3) AP が送信する利用者情報の種類（電話番号など）

(4) AP が取得した利用者情報の危険性の説明

(5) AP に対する警告画面の表示の有無の選択項目

(6) AP に対する処理内容の選択項目

(7) AP に対する処理内容の詳細

(8) 送信先の提示（IP アドレス）

(9) 漏洩時の日時

利用者の判断を支援する情報について，既存研究では，利用者に提供する情報としてパッケージ名や UID を提示する場合が多い．しかし，これらの情報では利用者は，AP が利用者情報を端末外部に送信する動作を許可するか否かの判断が難しい．このため，本手法では，利用者の判断を支援する情報として AP 名と送信される利用者情報を提示する．また，提示する情報量が多い場合，利用者が判断しづらくなる．このため，表示する警告画面は図 2 の様に 1 画面のみで警告の内容と処理選択ができるようにした．

図 2 警告画面

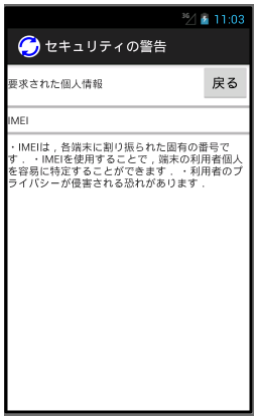

図 3 利用者情報の危険性の説明

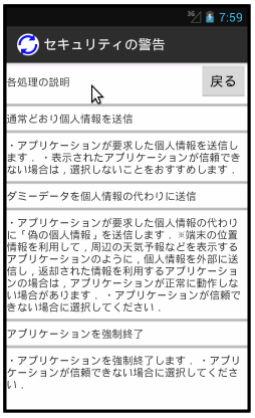

図 4 処理内容の詳細

さらに，利用者情報が端末外部に漏洩した場合の危険性を確認できることが利用者の判断を支援するうえで重要である．このため，図 2 の「危険性の説明」ボタンから，図 3 のように AP が取得した利用者情報が漏洩することによる危険性を確認できるようにする．なお，利用者情報が漏洩することによる危険性は，これらの情報を利用者が必要だと判断した場合に提供できるように，画面遷移を用いて提供する．

次に，利用者の負担を減らす工夫について述べる．本手法では，利用者情報が端末外部に送信されようとするたびに利用者に許可を求める．しかし，許可

を求める頻度が高くなると利用者の負担が増加する．このため，利用者に許可を求める頻度を抑制する必要がある．本手法では，図 2 のチェックボックスで表示されている項目のように，利用者が AP ごとに警告の有無を選択できるようにする．これにより，警告画面の表示頻度を抑制し，利用者の負担を減らす．

さらに，利用者情報を漏洩させた AP に対する処理選択において，利用者が処理選択を判断しやすいことが重要である．このため，警告画面における AP に対する処理選択の際，図 2 のラジオボタンで表示されている項目のように，選択項目を三つに限定し，簡単な選択で処理内容を決定できるようにする．また，AP に対する処理内容の詳細は，図 2 の「詳細」ボタンから図 4 のように確認できるようにする．なお，AP に対する処理内容の詳細は，これらの情報を利用者が必要だと判断した際に提供できるよう，画面遷移を用いて提供する．

4.4.2 処理の流れ

制御 AP は，漏洩検知機能から送信された情報のうち AP のパッケージ名から対応する AP 名を取得する．パッケージ名から対応する AP 名を取得する処理の流れを以下に示す．

(1) 端末にインストールされている AP のパッケージ名とアプリ名を取得・保持

(2) 漏洩検知機能から送信された AP のパッケージ名を取得

(3) (2) で取得したパッケージ名をキーとして (1) で取得した情報から該当するアプリ名を取得

上記の処理で取得した AP 名と利用者情報の種類を警告画面に提示し，AP が端末外部へ利用者情報を送信する動作を許可するか否かを利用者に尋ね，利用者の判断結果を取得する．その後，制御機能に取得した判断結果を送信する．

なお，AP 名を取得する際に，対応する AP のアイコンも取得できる．AP のアイコンを利用することで，利用者により分かりやすい警告画面の提示が可能だと考える．

4.5 制御機能

制御機能は，制御 AP で取得した利用者の判断結果に従って，AP が利用者情報を端末外部に送信する動作を制御する機能である．本機能は，漏洩検知機能の箇所に実装している．AP の処理の制御には，AP の処理を制御しない場合（送信を許可）と，AP を強制的に終了させる制御方法以外に，利用者の意図しない利用者情報の漏洩を防止し，かつ AP を継続して利用できる制御方法が必要である．このため，本手法では，以下を 3 つの制御方法を利用者に提供する．

（1）通常動作

利用者情報の端末外部への送信を許可する場合に選択する．利用者情報は，AP の動作により端末外部に送信される．

（2）ダミーデータの送信

利用者情報の端末外部への送信を許可せず，AP の動作を継続したい場合に選択する．利用者情報が端末外部に送信される際，送信される利用者情報の代わりに，対応するダミーデータを端末外部に送信する．これにより，AP の動作を終了せず，利用者情報の漏洩を防止できる．

（3）強制終了

利用者情報の端末外部への送信を許可しない場合に選択する．強制終了は，AP の動作を強制的に終了させ，利用者情報の漏洩を防止する．

なお，ダミーデータは，以下の二つの方法のどちらかで置換する．一つ目の方法は，null 文字列のデータに置換するものである．これは，端末外部に送信される利用者情報が，カメラ，マイク，およびログの場合に，使用する．二つ目の方法は，固定の値を返すものである．これは，端末外部に送信される利用者情報が，端末の位置情報，電話番号，および，IMEI（International Mobile Equipment Identifier）などの場合に使用する．

ただし，ダミーデータを送信することにより，AP が正常に動作しない場合がある．例えば，利用者の位置情報を取得し，その場所の天気予報を表示する AP が外部にダミーデータを送信したとする．このとき，AP は，利用者の正確な位置情報を取得できないため，利用者がいる場所の天気予報を表示できない．

4.6 期待される効果

本手法の実現により，以下の効果が期待できる．

(1) 誤検知の削減

(2) 利用者情報の漏洩の防止

表 2 ゲスト OS の評価環境

ディストリビューション	Ubuntu 10.04 LTS
カーネル	Linux ubuntu 2.6.32-42-generic
仮想 CPU 数	4
メモリ	4 GB

表 3 Android Emulator の評価環境

Android バージョン	Android 4.1
TaintDroid バージョン	TaintDroid 4.1(updated Dec 6, 2012)

(3) 利用者情報の漏洩に関わったすべての AP 名の取得

(4) AP の動作を妨げない利用者情報の漏洩の防止

(5) 利用者の負担の軽減

5 評価

5.1 評価項目

評価では，ゲスト OS 上の Android Emulator で行った．ゲスト OS と Android Emulator の評価環境を表 2 と表 3 に示す．また，以下に評価の目的と評価項目を示す．

（評価 1）実アプリを用いた追跡精度の比較

実際のアプリを用いて，TaintDroid と本手法での追跡精度を比較し，評価した．また，本手法で利用者情報が端末外部に送信される際の検知タイミングと処理を評価した．

（評価 2）複数の AP が連携した場合の利用者情報の漏洩防止

複数の AP で連携した利用者情報の漏洩を防止する実験について述べる．これにより，利用者が利用者情報の伝搬経路を把握できることを示す．

（評価 3）AP の実行から漏洩検知までの処理時間の測定

テスト AP を作成し，Android，TaintDroid，および本手法における AP の実行から動作終了までの処理時間を測定し，比較した．これにより，本手法の導入により発生するオーバヘッドを示す．

なお，AP 単位の利用者情報の漏洩防止の有効性は，文献 [9] で報告したため，省略する．

5.2 実アプリを用いた TaintDroid と本手法との追跡精度の比較

Google Play[2] 内の各カテゴリの無料アプリトップ 20 までの AP を取得し，その中からランダムに 100 個の AP を選択した．本手法を用いることで，選択した 100 個の AP が端末外部に利用者情報を送信する動作を検知できるか評価した．

表 4 実アプリの動作結果

インストール不可	8 個
正常に動作しない	17 個
利用者情報の送信なし (検知なし)	6 個
利用者情報の送信あり (検知あり)	69 個

表 4 は，選択した 100 個の AP を TaintDroid と本手法のそれぞれで動作させた結果を示している．

利用者情報の送信を検知した際に送信されていた利用者情報は，IMSI（International Mobile Subscriber Identity）と IMEI である．IMSI は，利用者情報の送信を検知した AP のすべての AP で送信されていた．また，利用者情報を送信するタイミングは，AP の起動時である．なお，IMEI を送信する AP は少なく，利用者情報の送信を検知した AP(69 個) の内 9 個である．

IMSI と IMEI 以外の利用者情報の送信に関して，GPS などのセンサ情報の取得は，エミュレータ上で AP がセンサ情報を取得できない．このため，GPS などのセンサ情報に関する利用者情報の送信は検知できていない．また，AP がセンサ情報を取得できないことにより，AP が正常に動作しない場合があった．

本評価において，TaintDroid と本手法を比較した結果，TaintDroid と本手法の追跡精度は同等である．これは，本手法が TaintDroid を基礎として使用しているためである．検知のタイミングについて，本手法は，AP が利用者情報を外部に送信するタイミングでその動作を検知し，動的に利用者が送信の実行可否を制御できる．また，TaintDroid と本手法の追跡精度は同等であるが，本手法は，利用者情報を追跡する際に利用者情報の伝搬に関わった AP 名などの TaintDroid が保持していない情報を保持している．

なお，本手法において，AP が利用者情報を外部に送信する動作を強制的に終了した場合，その後の処理は AP 側の実装に依存する．例えば，本評価で利用者情報の送信を検知した 69 個の AP において，AP が端末外部に利用者情報を送信する動作を強制的に終了させた場合，60 個の AP で再度利用者情報の送信を行う動作が実行された．このように，本手法を用いた場合でも，何度も利用者情報の送信を行

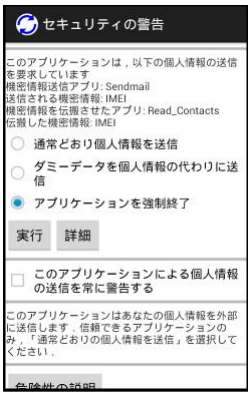

図 5 複数の AP が連携した場合の警告画面

う動作が実行される場合があることが分かった．

5.3 複数の AP が連携した場合の利用者情報の漏洩防止

2 つの AP が連携した場合の利用者情報の漏洩を防止する例を示す．1 つ目の AP（ReadContacts）は IMEI を取得したのち 2 つ目の AP（SendMail）に取得した情報を送信する．2 つ目の AP は，送信された情報（IMEI）を端末外部に送信する．

図 5 は，上記の AP を動作させた際の警告画面である．利用者情報の漏洩に 2 つの AP が関わっていることが分かる．なお，今回は，2 つのアプリが連携した場合の例を示したが，2 つ以上のアプリが連携した場合も同様に利用者情報の漏洩に関わったすべての AP の AP 名が，図 5 の警告画面に表示される．

以上のことから，本手法では，複数の AP が連携した場合であっても利用者情報の漏洩に関わった AP をすべて把握できており，利用者は利用者情報の漏洩に関わった AP を正確に把握できる．

5.4 処理時間の測定比較（オーバヘッド）

テスト AP を作成し，Android，TaintDroid，および本手法における AP の起動から動作終了までの処理時間を測定し比較した．

作成したテスト AP は，実行後，ボタン処理などの人による操作を行わず，電話番号を取得し，その後，端末外部に電話番号を送信するものである．

評価結果を表 5 に示す．本手法は，TaintDroid と比べて AP 起動から動作終了までの処理時間において 9 秒弱処理が遅くなっている．これは，AP が利用者情報を送信する際，その動作を検知し制御しているためである．

本手法で発生するオーバヘッドは，利用者が体感できるものである．しかし，AP が利用者情報を端末外部に送信する際に発生するものであり，許容できる範囲のものであると考える．

表 5 処理時間の測定比較 (オーバヘッド)(単位 : ms)

測定タイミング	Android	TaintDroid	本手法
AP 起動から動作終了まで	4353.0	4435.2	13225.0
AP 起動から画面表示	94.3	132.4	128.0
情報送信の完了まで	4258.6	4302.8	13097.0
情報の取得	7.3	8.8	9.5
メッセージ送信処理	4248.6	4291.6	13084.8
送信するテキストの設定	44.0	47.2	52.3
コネクト処理	2579.0	2344.4	4983.5
メッセージの送信	1733.0	1845.2	7978.3
その他	56.2	54.8	59.3

6 関連研究

端末外部に利用者情報を漏洩させるマルウェアへの対策が重要視されており，この問題に対処するために，AP の解析手法や利用者情報の漏洩防止手法が研究されている．AP の解析手法として，DroidScope [13] や文献 [14] の手法が提案されている．DroidScope は，仮想化技術を用いた Android マルウェア解析手法であり，Java コンポーネントとネイティブコンポーネント間での利用者情報の伝搬追跡やいくつかの解析ツールを解析者に提供している．また，仮想マシン上で動作する Android システムの変更が不要であるという特徴がある．また，文献 [14] の手法は，TaintDroid を拡張し，利用者情報の対象範囲と追跡範囲を拡張した手法であり，従来の TaintDroid では追跡できなかったデータベースを介した利用者情報の送信を検出できる．

本手法では，利用者情報の漏洩防止を目的としており，利用者情報の漏洩を検知するために，動的解析手法である TaintDroid を利用している．このため，DroidScope と比べて，利用者情報の伝搬追跡範囲は狭いが実際の Android 端末上で動作させることが可能であり，実際の Android 端末の情報が必要な AP の動作の検出が可能であるという特徴がある．また，文献 [14] の手法の様に TaintDroid を拡張した手法と本手法を併用することで，利用者情報の伝搬追跡範囲を拡張することが可能である．

7 おわりに

本稿では，文献 [9] の手法の実現方式と評価について述べた．本手法は，端末内部の個人情報の伝搬を追跡し，個人情報が端末外部に漏洩する際に利用

者の判断に従って AP の動作を動的に制御する．これにより，端末外部への個人情報の漏洩を防止できることを示した．また，本手法の Android での実装について述べ，追跡精度と処理のオーバヘッドを評価した結果を示した．

参考文献

[1] Android, 入手先〈http://www.android.com/〉(参照 2013-11-29)

[2] Google Play, 入手先〈https://play.google.com//store〉(参照 2013-11-29)

[3] Zhou, Y. and Jiang, X. :Dissecting Android Malware: Characterization and Evolution, Proceedings of the 33rd IEEE Symposium on Security and Privacy (Oakland 2012), (2012).

[4] Symantec，日本の Android ユーザーから利用者情報を盗み出す”The Movie”マルウェア入手先〈http://www.symantec.com/connect/ja/blogs/android-movie〉(参照 2013-11-29)

[5] 産経ニュース，スマホアプリで 76 万件分の利用者情報流出か「全国電話帳」インストールに注意入手先〈http://sankei.jp.msn.com/affairs/news/121006/crm12100617380008-n1.htm〉(参照 2013-4-10)

[6] Sakamoto, S., Okuda, K., Nakatsuka, R. and Yamauchi, T. : DroidTrack: Tracking and Visualizing Information Diffusion for Preventing Information Leakage on Android, Journal of Internet Services and Information Security (JISIS), vol.4, no.2, pp.55-69 (2014).

[7] 林 里香，後藤 厚宏：Android アプリケーション利用の安全性を高めるアプリケーション動作の「見える化」，コンピュータセキュリティシンポジウム 2012(CSS2012) 論文集，vol.2012，no.3，pp.130–137(2012).

[8] Beresford, A.R., Rice, A., Skehin, N. and Sohan, R. : MockDroid: Trading privacy for application functionality on smartphones, Proc. 12th Workshop on Mobile Computing Systems and Applications (HotMobile) (2011).

[9] 小倉 禎幸，山内利宏：細粒度の情報追跡による機密情報送信の動的制御手法，情報処理学会研究報告，vol.2013-CSEC-62，no.20，pp.1-7 (2013).

[10] Enck, W., Gilbert, P., Chun, B.G., Cox, L.P., Jung, J., McDaniel, P. and Sheth, A.N. : TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones, Proc. 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI) (2010).

[11] 梶原 直也，堀 良彰，櫻井 幸一：情報フロー追跡を用いた Android 端末における情報送信制御，2013 年暗号と情報セキュリティシンポジウム (SCIS2013) 論文集，電子媒体 (2013).

[12] Hornyack, P., Han, S., Jung, J., Schechter, S. and Wetherall, D. : These aren't the droids you're looking for: retrofitting android to protect data from imperious applications, Proc. 18th ACM conference on computer and communications security (2011).

[13] Yan, L.K. and Yin, H. : DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis Proc. 21st USENIX conference on security symposium (2012).

[14] 名雲 孝昭 , 秋山 満昭 , 針生 剛男：ContentProvider を用いた利用者情報送信の動的解析手法に関する検討，情報処理学会研究報告，vol.2013-CSEC-60, no.52，pp.1-6 (2013).