Android アプリケーションに対する情報フロー制御機構の提案

葛野 弘樹 †

† セコム株式会社 IS 研究所
181-8528 東京都三鷹市下連雀 8-10-16 セコム SC センター
h-kuzuno@secom.co.jp

あらまし Android ではアプリケーションに対する端末上の情報やデバイスへのアクセス制御を，パーミッションを管理し制御する．パーミッションのアクセス権限は，アクセス対象の端末上の情報やデバイスのみを示しており，複数パーミッションの組み合わせによる危険性は明示されない．そのため，ネットワーク通信と端末上の情報にアクセス可能なアプリケーションによる意図しない端末情報の外部送信が起こる問題がある．この問題を解決するためにアプリケーションの通信を監視しパーミッションの組み合わせにより発生する脅威を防ぐ手法を提案する．提案手法により，アプリケーションの送信情報を細かく制御し，意図しない端末情報の送信を防止する．

A Proposal of An Information Flow Checking System for Android Application

Hiroki Kuzuno†

†Intelligent Systems Laboratory, SECOM Co., Ltd.
SECOM SC Center, 8-10-16 Shimorenjaku, Mitaka, Tokyo 181-8528, Japan
h-kuzuno@secom.co.jp

Abstract Recently, Android are becoming popular smartphones OS. Android applications require permissions to access personal information or device resources. If application has multi permissions which have a combination of network connectivity with personal information accessibility, can transmit it across the network. In this paper, we propose a monitoring application to detect leakage of personal information which is caused by these applications. Our method inspects the application outgoing traffic and performs fine-graded information control flow to avoid unintended leakage.

1 はじめに

個人向け携帯端末として，スマートフォンが普及してきている．スマートフォンの OS としては，Apple Inc. の開発する iOS と，Google Inc. を中心に開発されている Android が広く利用されている．スマートフォン向けに開発されたアプリケーションは，App Store や Android Market 等のアプリケーションマーケットにて公開することが可能であり，多数のアプリケーションが提供されている．

携帯端末には，位置や端末情報などのユーザのセンシティブな情報が記憶されており，アプリケーションマーケットよりインストールしたアプリケーションによる情報漏洩 [9, 10] や盗聴の危険性 [13] が指摘され，DroidDream などスマートフォンを対象としたマルウェアも発見さ

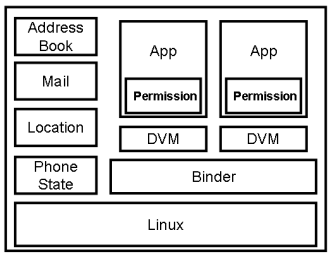

図 1: Android アーキテクチャの概要

れている [1].

Android では，Android アプリケーション (以下，アプリケーションとする) による位置や端末情報やデバイスへのアクセス権限をパーミッションとして管理しており，アプリケーションのインストール時にそのアプリケーションがアクセスする端末上の情報やデバイスを確認可能である．しかし，インストール時の確認画面では，パーミッションと対応した端末上の情報やデバイスの説明のみしか記載されないため，ユーザはアプリケーションが取得した情報がどのように扱われるかといった挙動を正しく把握できない可能性がある．その結果，複数のパーミッションの組み合わせによってはアプリケーションにより端末上の情報を外部へ送信され，情報漏洩に繋がる危険性がある．

本稿では，このような危険性に対応するためにアプリケーションの外部への通信を端末上で監視し，予め決められた情報フロールールに基づきアプリケーションの通信制御を行う手法を提案する．提案手法により，ユーザはアプリケーションが取得した端末上の情報の外部送信を把握することができ，意図しないタイミングで起る情報漏洩の未然防止を実現する．

2 背景知識と課題

2.1 Android のアーキテクチャ

Android のアーキテクチャの概要を図 1 に示す．Android は OS に Linux カーネル，ミドルウェアとして Binder，Dalvik VM，アプリケーション，位置情報等の端末上の情報から構成される．アプリケーションはサンドボックス化として，他のアプリケーションのデータへのアクセス制御を行うために固有のユーザ ID/グループ ID が割り当てられ，アプリケーション毎に起動された Dalvik VM 上で動作あるいはライブラリとして呼び出し実行される．アプリケーションから端末上の情報へのアクセスや他のアプリケーションとの通信は Binder を通じたプロセス間通信により行われる．

2.2 パーミッションによるアクセス制御

Android では，アプリケーションによる端末上の情報や機器へのアクセス権限をパーミッションとして管理しており，アプリケーションが Binder を通じてアクセスする際に正しいパーミッションがアプリケーションに付与されているかどうか確認しアクセス可否を判断することでアクセス制御を実現している．Android API Level 13 におけるパーミッションは 117 個 [2] あり，そのうち端末上の情報取得に用いられるパーミッションの一部を表 1 に示す．Android では，端末上に保存されている情報の多くをパーミッションが付与されていれば利用可能であり，ユーザのセンシティブな情報へアクセスするためのパーミッションが多く含まれている．

パーミッションは開発者の判断でアプリケーションの作成時に付与することが可能であり，ユーザは図 2 のようにインストール時にアプリケーションに付与されているパーミッションリストとアクセス対象の情報を確認することができる．

2.3 複数パーミッションによる課題

ユーザは図 2 に示したようにインストール時にパーミッションを確認できるが，アプリーケーション内のどの機能やタイミングにおいてパーミッションを必要としているか判断することは難しい．また，ネットワーク通信と端末上の情報を取得するパーミッションの組み合わせを要求しているアプリケーションでは，取得した端末上の情報を外部に送信する可能性が十分に考えられるが，それをユーザがインストール時に知

表 1: 端末上の情報取得に用いられる Android パーミッション

Permission	Description
ACCESS_FINE_LOCATION	GPS を利用した位置情報へのアクセス
READ_SMS	SMS メッセージの読み取り
READ_PHONE_STATE	携帯端末情報 (IMEI，IMSI, ICC-ID など) の読み取り
READ_LOGS	システムログファイルの読み込み
READ_HISTORY_BOOKMARKS	ブラウザの閲覧履歴とブックマークの読み取り
READ_CONTACTS	アドレス帳の読み込み
RECORD_AUDIO	音声の記録
PROCESS_OUTGOING_CALLS	発呼の監視，変更，停止
CAMERA	カメラデバイスへのアクセス

図 2: パーミッション確認画面の例

ることは困難であり，アプリケーションの種別や名前とは明らかに関係のないパーミッションを要求される場合，インストールを避けアプリケーションを使用しないよう自衛するしかない．

我々は，Android アプリケーションランキングサイトおよびいくつかの公式マーケットで推薦されているアプリケーションからネットワーク通信と位置情報，端末情報，音声，アドレス帳のパーミッションの組み合わせを調査した．その結果を表 2 に示す．組み合わせとしては，位置情報と端末情報を要求される場合が最も多く，いくつものアプリケーションが端末上の情報の利用を想定していることが分かる．

3 提案手法

3.1 Android アプリケーションに対する情報フロー制御

アプリケーションによる端末上の情報の外部送信により情報漏洩に繋がるかどうかは，アプリケーションが端末上の情報を取得した後の外部送信を行う状況に依存している．アプリケーションの中には提供する機能を利用する上で端末上の情報を必要とする場合があり，ユーザが情報の送信に同意した状況において，端末上の情報が外部に送信されるのは情報漏洩とは言いがたい．しかし，ユーザが意図していない状況やアプリケーションの利用にあたり必ずしも必要ではない端末上の情報を無断で外部に送信することは，ユーザには意図しない情報送信であり，情報漏洩に繋がる可能性がある．

そこで，ユーザの意図しない状況でのアプリケーションによる端末上の情報送信による情報漏洩を未然に防ぐために，アプリケーションのネットワーク通信を監視しあらかじめ決められた情報フロールールに基づき送信の可否を制御する手法を提案する．

提案手法では，アプリケーションの監視を，Android のカーネルである Linux において，アプリケーションが外部に送信するネットワークパケットを Android アプリケーションとして動作している情報フロー制御アプリケーションに転送することで実現し，監視対象アプリケーショ

表 2: Android パーミッションの組み合わせ (I:Internet, L:Location, P:Phone State, A:Audio, RC:Read Contacts)

Permissions					#	Applications
I	L	P	A	RC		
x		x			5	仕事効率化 01，カジュアル 04, カジュアル 01, ライフスタイル 01, 音楽&オーディオ 01, ニュース＆雑誌 01
x	x	x			9	ツール 02，ライフスタイル 03，健康&フィットネス 01，ツール 01, カジュアル 02，交通 01，アーケード&アクション 01, エンターテイメント 01，天気 01
x	x	x		x	2	ソーシャルネットワーク 01，ソーシャルネットワーク 02
x	x				3	ソーシャルネットワーク 04，ライフスタイル 02，天気 02
x	x		x	x	1	仕事効率化 02
x	x			x	1	ソーシャルネットワーク 03
x					4	ライフスタイル 04，ファイナンス，ソーシャルネットワーク 05, カジュアル 03

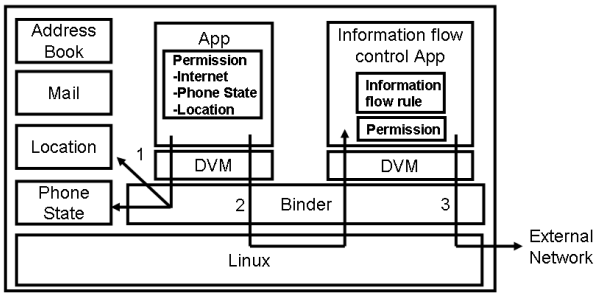

図 3: 情報フロー制御アプリケーションの概要

ンに自身の送信したパケットが監視されていることを知られること無く情報フロー制御を行う．情報フロー制御アプリケーションの概要を図 3 に示す．

情報フロー制御の手順は以下の通りである．

(1) 監視対象アプリケーションはパーミッションで許可された端末上の情報を取得

(2) 監視対象アプリケーションのネットワーク通信は情報フロー制御アプリケーションが全て中継

(3) 情報フロー制御アプリケーションは通信内容に端末上の情報が含まれていた場合，情報フロールールに基づき送信可否を判断，許可の場合は外部に転送，拒否の場合はブロック

なお，情報フロー制御アプリケーションにおける監視対象のプロトコルとしては HTTP を想定している．

3.2 情報フロールール

情報情報フロールールの記法の一部を図 4 に示す．情報フローの制御としては，送信の許可，拒否，および制限付き送信に対応している．制限付き送信は，送信する情報の値を書き換える際に用いる．情報の種類は，位置，端末情報，アドレス帳，メールなど Android パーミッション [2] から利用者の個人情報に関わる情報に対応している．ユーザ通知は，情報フロールールとマッチングした場合にユーザへのフィードバックに用いる．

情報フロールールの生成は，情報制御アプリケーションにより，監視対象アプリケーションの持つパーミッション情報を利用することでアプリケーションの取得可能な端末上の情報を特定し，ある程度自動化が可能である．通信先は，アプリケーションにおいて実際に通信が発生しないと把握することが難しいため，通信毎に情報フロールールをユーザの判断を元に修正する必要がある．

```
<rule-set>::= <rule> | <rule><rule-set>
<rule>::= <type> “application-id”
          <data-type><destination>
          <user-notify>
<type>::=“allow”|“deny”| <restrict>
<restrict>::=“restrict” <modify-rule>
          “value-range”
<modify-rule>::=“random”|“permutation”
<data-type>::= <location> | <phone-state>
          | <address-book> | <mail>
<location>::=“latitude” “graditude”
<phone-state>::=“IMEI” “IMSI” “ICC-ID”
          “device-info”
<address-book>::=“name” “phone-number”
          mail-address”
<mail>::=“from” “to” “body”
          “attached-file”
<destination>::=“address” | “port”
<user-notify>::=“true” | “false”
```

図 4: 情報フロールール BNF 記法

4 実験結果と評価

ネットワーク通信と端末上の情報取得のパーミッションを付与された Android アプリケーションによる端末上の情報の外部送信を調査し，提案手法を用いて検出することで情報フローを制御可能か評価を行った．調査対象は表 2 のアプリケーションとし，評価環境としては Nexus S，Android 2.3.4 を用いた．

調査および評価結果を表 3 に示す．調査対象アプリケーションでは，端末情報 (表 3 では Phone State)，続いて位置情報 (表 3 では Location) の送信が多く，付与されたパーミッションを利用して取得した情報を送信していることが確認された．また，全てのアプリケーションにおいてパーミッションを必要とせずに取得できる端末上の情報である Android のバージョン，端末名称，キャリア名称 (表 3 では Et Cetra) を外部送信しているのが確認できた．主な送信先はモバイル広告サイトで，ターゲット広告や統計のためのユーザ追跡に利用されている可能性がある．

提案手法による検出結果では，調査結果において確認できた端末上の情報の外部送信をほぼ検出できることを確認できた．これは，調査対象アプリケーションより送信されている情報の殆どが Android から取得できるデータ形式のままであり，容易に検出できることが大きい．そのため，提案した情報フロールールを利用しても送受信の制御を十分行えるといえる．しかしながら，暗号化通信や符号化を行われている場合は提案手法では検出が困難であり，アプリケーションによっては提案手法では未検知のまま情報送信が行われていることは十分に考えられる．

5 関連研究

個人の所有する計算機で使用しているアプリケーションによる情報漏洩の危険性は従来から指摘されており，アプリケーションの振る舞いや送信情報を検査する手法が提案されている [3, 4]．

Android アプリケーションでは，静的解析によりネットワーク通信のパーミッションと他のパーミッションの組み合わせが多いこと [8] や端末上の情報を外部に送信しているアプリケーションが存在することが報告 [10] されている．

アプリケーションによる情報漏洩を検出する試みとして，Android フレームワークのうちパーミッションの記法を拡張し，より細かくアクセス制御を実現する手法が提案されている [5, 6, 14]．また，アプリケーションの実装の不備により，アプリケーション間の通信を盗聴される危険性も指摘されており [12]，動的にアプリケーションの振る舞いを監視し Android フレームワークにおいて端末上の情報フローを制御し情報漏洩を防止する手法 [7] やログ情報を監視する手法 [15]，アプリケーションによる情報漏洩の有無を自動検査する試み [11] が提案されている．

先行研究のうち，Android フレームワークの変更が必要な場合は，Android のバージョンアップなどへの対応が困難である．また，ログ情報の監視では，ログを出力しないアプリケーションでは監視が難しい．提案手法では，アプリケーションとして他のアプリケーションのネットワーク通信を監視するためこれらの問題の影響は受けない．

表 3: アプリケーションによる端末上の情報送信および提案手法による検出結果 (L:Location, P:Phone State, E:Et Cetra(パーミッションを必要としない端末上の情報))

Application	Send Data			Detection		
	L	P	E	L	P	E
カジュアル 01			x			x
ライフスタイル 01		x	x		x	x
ツール 01			x			x
カジュアル 02	x	x	x	x	x	x
アーケード&アクション 01		x	x		x	x
ソーシャルネットワーク 01			x			x
ソーシャルネットワーク 02	x	x	x	x	x	x
ソーシャルネットワーク 03	x		x	x		x
ライフスタイル 02	x		x	x		x
ニュース＆雑誌 01		x	x		x	x
カジュアル 03			x			x

6 まとめ

本稿では，Android 端末上で動作するアプリケーションのネットワーク通信を監視し，端末上の情報の外部送信を情報フロールールに基づき制御する手法を提案した．提案手法により，アプリケーションに対してネットワークを通じて送信する情報を細かく制御することで，ネットワーク通信と端末上の情報取得のパーミッションを付与されたアプリケーションによる意図しない情報漏洩を未然に防止することができる．

今後は，提案手法における情報フロールールについてより適切に作成するための手法を検討予定である．また，アプリケーションの送信する情報が符号化，暗号化されていた場合，提案手法では検出が難しくなるため，Android フレームワークとどの程度連携するのが有効か検討していきたい．

参考文献

[1] Lookout, Mobile Security Report https://www.mylookout.com/mobile-threat-report, (2011).

[2] Android Developers, Manifest.permission, http://developer.android.com/reference/android/Manifest.permission.html, (2011).

[3] Heng, Y., Dawn, S., et al.: Panorama: Capturing System-wide Information Flow for Malware Detection and Analysis, *Annual Computer Security Applications Conference*, (2007).

[4] Jaeyeon, J., Anmol, S., et al.: Privacy Oracle: a System for Finding Application Leaks with Black Box Differential Testing, *ACM Conference on Computer and Communications Security*, (2008).

[5] Machigar, O., Stephen, et al.: Semantically Rich Applicaiton-Centric Security in Android, *Annual Computer Security Applications Conference*, (2009).

[6] William, E., Machigar, O., Patric, M.: On Lightweight Mobile Phone Application Certification, *ACM Conference on Computer and Communications Security*, (2009).

[7] William, E., Peter, G., et al.: TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, *9th USENIX Symposium on Operating Systems Design and Implementation*, (2010).

[8] David, B., H. Gunes, K., et al.: A Methodology for Empirical Analysis of Permission-Based Security Models and its Application to Android, *ACM Conference on Computer and Communications Security*, (2010).

[9] Manuel, E., Christopher, K., et al.: PiOS: Detecting Privacy Leaks in iOS Applications, *18th Annual Network & Distributed System Security Symposium*, (2011).

[10] William, E., Damiren, O., Patric, M., Swarat, C.: A Study of Android Application Security, *The Proceedings of The 20th USENIX Security Symposium*, (2011).

[11] Peter, G., Byung-Gon, C. and Landon, P, C., Jaeyeon, J.: Vision: Automating Privacy Testing of Smartphone Applications, *The Proceedings of the second international workshop on Mobile cloud computing and services*, (2011).

[12] Erika, C., Adrienne, P. F., Kate. G., David. W.: Analyzing Inter-Application Communication in Android, *The International Conference on Mobile Systems, Applications, and Services*, (2011).

[13] Roman, S., Kehuan, Z., et al.: Soundcomber: A Stealthy and Context-Aware Sount Trojan for Smartphones, *18th Annual Network & Distributed System Security Symposium*, (2011).

[14] Mohammad, N., Sohail, K., Xinwen, Z.: Apex: extending Android permission model and enforcement with user-defined runtime constraints, *The Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, (2011).

[15] 竹森 敬祐, 磯原 隆将, 窪田 歩, 高野 智秋.: Android 携帯電話上での情報漏洩検知, 暗号と情報セキュリティシンポジウム, (2011).