アプリケーションの挙動を可視化することによるセキュリティ対策

戸田 尚希　鈴木 秀和　旭 健作　渡邊 晃

名城大学大学院理工学研究科

ec e, e , e e e , c,
, , e , e , , , , , e
, , e c c e ce, ec , e e

1 はじめに

インターネットの発展に伴い，誰でも自由にインターネットを利用できるようなり，近年ではスマートフォンと呼ばれる携帯端末が普及してきた．中でも，Android スマートフォンが急速な普及をみせている．Android はオープンソースであり，メーカーで独自に OS の拡張が可能である．アプリケーションの開発環境は無料で構築でき，誰でも自由にアプリケーションを開発することができる．また，作成したアプリケーションを Market 上に公開する際に事前審査がないため，誰でも気軽に公開できる．このような利点がある反面，マルウェアと呼ばれるアプリケーションの開発・公開も容易にできてしまう点が課題となっている．近年では，Android 端末をターゲットにしたマルウェアが数多く確認されている[1]．Android 端末の今後の更なる普及を考えると，マルウェア対策は重要な課題である．

Android におけるマルウェアは，アプリケーションに内包された形で存在する．Android 端末は，マルウェアが内包されたアプリケーションをインストールすることによってマルウェアに感染する．

Android では，セキュリティモデルとしてパーミッション機構が採用されている．パーミッション機構とは，アプリケーションが利用する機能や情報をユーザに表示して承認を求める仕組みである．また，Android のアプリケーションは Dalvik VM と呼ばれる仮想マシン上で実行される．アプリケーションは，パーミッションが付与されてはじめて，仮想マシン外にある機能へのアクセスが許可される．マルウェアは，マルウェアとしての動作を行うために必要以上にパーミッションを要求する．そのため，アプリケーションインストール時にユーザがパーミッションを確認することでマルウェアを感染から防ぐことができる可能性もある．このことから，インストールするアプリケーションにさえ気を付けていれば，Android は通常の PC よりは安全であるといえる．

パーミッション機構の課題として，アプリケーションに付与したパーミッションがどのタイミングで利用されているかユーザは知ることができない点，付与されたパーミッションの組み合わせでアプリケーションがどのような動作を行うのかユーザが予測することは困難な点がある．そのため，複数のパーミッションの組み合わせによって，Android 端末内の位置情報等の機密情報が外部に送信される情報漏洩がユーザの知らないところで起こってしまう危険性がある．

Android をターゲットにしたマルウェア対策の既存技術としては，ノートンモバイルセキュリティ，ウイルスバスターモバイル for Android，McAfee Mobile Security 等の Android 端末向けのウイルス対策ソフトがある．ウイルス対策ソフトのウイルス検出の基本はパターンマッチングである．パターンマッチングは，ウイルスの特徴をパターンとして取り出し，定義ファイルとする．それを検索対象となるファイルと照合することでウイルスを検出する．しかし，未知のウイルス等の定義ファイルにないパターンは検出できないという課題がある．

本提案では，Android 上で動作するアプリケーションによる位置情報等の機密情報の利用や外部サーバへの送信をユーザに対して可視化する．可視化した結果，個々のユーザが判断した要注意アプリケーションの情報をサーバで管理することにより，複数のユーザに対してアプリケーションのインストールにおける安全性の判断を補助するシステムを提案する．

本提案では，個人情報や位置情報といった機密情報の利用や外部サーバへの送信等，本来ユーザが関知しないアプリケーションの動作を可視化することにより，ユーザが理解できる形で示すことができる．これにより，ユーザ自身でアプリケーションの動作の正当性を考える必要があるため，セキュリティ意識の向上を期待できる．さらに，要注意アプリケーションの情報を複数のユーザで共有することにより，要注意アプリケーションをインストールの段階で防ぐことができる．

以下，2 章で Android セキュリティの課題を述べる．3 章で，関連研究についての説明を行い，4 章で提案方式について述べる．5 章でまとめる．

2 Android セキュリティの課題

2.1 マルウェア被害の分類

表 1 に既存のマルウェアが Android 端末に対してどのような被害を及ぼしているかを分類した．マルウェアが及ぼす被害は，端末内における被害，情報送信被害，ダウンロード被害，他の被害に分類できる．端末内における被害は，端末内部の設定を変更する被害である．情報送信被害は，端末内の情報を外部へ送信する被害である．ダウンロード被害はアプリケーションを勝手にダウンロードすることによる被害である．この被害は，端末内における被害と情報送信被害の発生に繋がる可能性を持っている．他の被害とは，端末内における被害，情報送信被害，ダウンロード被害に該当しない被害とした．中でも，端末内における被害と情報送信被害の原因となるマルウェアが数多く報告されている[2]．この 2 種類の被害を比較すると，ユーザにとって被害が大きいのは情報送信被害である．その理由は，携帯端末である Android 端末は PC よりも端末の利用頻度が高い分，位置情報やアドレス帳等，よりプライベートな情報が保存されている．このような情報が外部へ漏れることで犯罪などに利用される可能性もある．そのため，情報送信被害の対策が Android 端末におけるセキュリティ上の最大の課題であるといえる．

表 1　マルウェアによる被害の分類

	被害内容
情報送信被害	IMEI，IMSI をサーバに送信
	送受信される SMS メッセージを外部サーバに送信
	プレミアム SMS の番号への SMS 送信
	現在の GPS ロケーションのアップロード
	電話番号，連絡先リストをサーバに送信
	写真撮影とアップロード
端末内における被害	デフォルトブラウザのホームページを変更
	デフォルトブラウザにブックマークを登録
	ホーム画面にショートカットを追加
	ネットワーク設定の変更
	他の実行中のプロセスの強制終了
	特定の SMS メッセージの削除
	通話内容を記録して SD メモリカードに保存
	電話の再起動
ダウンロード被害	Android アプリケーションを/system ディレクトリにインストール
	APK ファイルのインストール・アンインストール
他の被害	送られてくる SMS メッセージの監視
	完全自動で攻撃者の Web サイトにアクセス
	ルート権限の取得

2.2 パーミッション機構の課題

Android では，セキュリティモデルとしてパーミッション機構が採用されている．パーミッション機構の課題としては以下の点が挙げられる．

課題1　アプリケーション動作中のどのタイミングでパーミッションが利用されるのかユーザは知ることができない．

課題2　複数のパーミッションを組み合わせることで起こるアプリケーションの動作をユーザが予想することは難しい．

Android では，アプリケーションインストール時にそのアプリケーションが要求しているパーミッションはユーザに表示されるが，ユーザはその表示からアプリケーションの動作を予想しなければならない．また，インストール後もパーミッション利用のタイミングはアプリケーションに委ねられているため，位置情報の外部サーバへの送信や，ユーザの意図しない SMS の送信等が行われていてもユーザは知ることができない．

3　関連研究

Android のセキュリティに関連する研究の一覧を図 1 に示す．図 1 は，これらの技術が，アプリケーションを提供する Market，端末内の Application 層，Application Framework 層，Linux Kernel 層，そしてユーザ自身のどこで主な対策を行っているのかという点についてまとめたものである．

3.1 Market における対策

文献[3]では，アプリケーションを提供する Market が安全になれば，多くのユーザの携帯端末を保護できるという考えの下，アプリケーションの事前審査ツールを提案している．この提案では，アプリケーションが出力する Logcat ログに注目している．Logcat ログとは，アプリケーションやシステムから出力されるログである．取得した Logcat ログと正規表現で表現された重要情報の組み合わせシグネチャ，広告シグネチャとを照らし合わせて，情報漏洩の痕跡や広告表示の痕跡を検知する．この提案は，アプリケーションを提供する Market 側で運用されることを想定しているため，2.2 節における課題 1，課題 2 は解決できない．さらに，システム導入の敷居が高いことが課題である．

3.1 Application 層における対策

文献[4]では，法人向けの Android 端末のセキュリティ保護を目的とした研究が行われている．この提案では，法人管理者が予めアプリケーションを実行して，その動作ログを安全性の評価を行うセンター局へ送信する．センター局は動作ログの解析を行い，脅威の低いアプリケーションのホワイトリストを作成する．各端末では，独自のアプリケーションがこのホワイトリストをチェックし，インストールするアプリケーションが安全かどうかを判断する．また，アプリケーションが認定されていない場合，ユーザに対してアンインストールを促す．この提案は，ホワイトリストに存在するアプリケーションのみがインストールできる仕組みなので，2.2 節における課題 1，課題 2 は解決できない．さらにこの提案は，法人向け Android 端末に対する提案であり，一般ユーザに対してはこの考えを適用することは非現実的である．

文献[5]では，アプリケーションインストール時にパーミッションに基づいた危険性検知と危険性提示を行うことで，ユーザのアプリケーション導入の判断支援を行う．この提案では，パーミッションの組み合わせに着目している．特定のパーミッションの組み合わせを持つアプリケーションでは，位置情報の流出等の危険がある．そのため，位置情報の流出等の危険性を示すアイコンを表示するユーザインタフェースを作成している．これにより，アプリケーションをインストールする際のユーザ判断を補助する．パーミッションの組み合わせにより起こりうる事態を，アイコンを用いてユーザに示すため，2.2 節における課題 2 を解決できる．しかし，いつパーミッションが利用されたかユーザに知らせることは出来ないため，課題 1 は解決できない．危険性検知において，マルウェア 180 個，Android Market 上から入手したアプリケーション 261 個を用いて評価を行ったところ，マルウェアの検知率が 95%と高い検知率であった．しかし，マルウェアでない Android Market から入手したアプリケーションも 62%を危険性のあるアプリケーションとして検知した．したがって，この提案は，誤検知率の高さが課題である．

3.1 Application Framework 層における対策

文献[6]では，個人情報等を扱う API への割り込みと，割り込みを受ける制御アプリケーションを Android に実装することによって，API ごとのリアルタイム動的制御方式を提案している．この提案では，パーミッションに保護されている個人情報を扱う API へフックポイントを設ける．アプリケーションが実行されてフックされると，拡張した Android Framework により，独自の制御アプリケーションによって定義されたセキュリティポリシーリポジトリが参照される．セキュリティポリシーの設定によって個人情報や位置情報へのアクセスを許可したり，拒否したりすることができる．こ

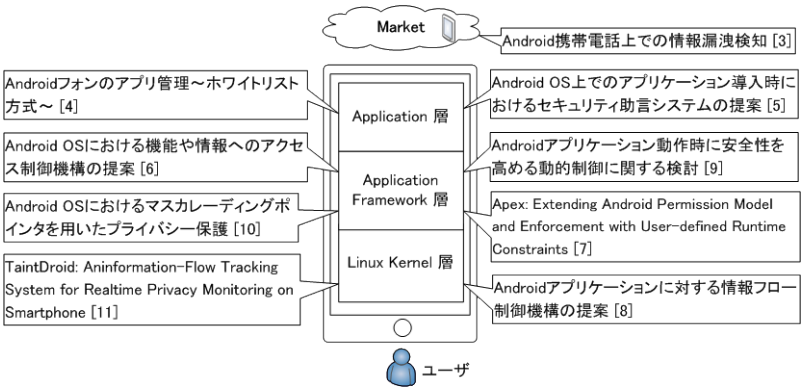


図 1 Android のセキュリティ対策における関連研究

の提案は，パーミッションが必要な機能を利用する毎に，ユーザに許可・拒否の判断を求めるので，2.2 節における課題 1 は解決できる．しかし，パーミッションの組み合わせによるアプリケーションの動作をユーザに提示することはできないため，課題 2 は解決できない．また，API が呼び出されるたびにセキュリティポリシーを参照するため，個人情報等への参照に時間がかかるという課題がある．

文献[7]では，アプリケーションが要求する全てのパーミッションを許可しなければアプリケーションをインストールできないという問題点を解決している．この提案では，アプリケーションによって要求されるパーミッションを条件付で許可する等，ユーザが許可するパーミッションを選択できるフレームワークを提案している．そのため，ユーザはアプリケーション実行時のパーミッション利用のタイミングや，複数パーミッションによるアプリケーションの動作を知ることはできない．したがって，2.2 節における課題 1，課題 2 は解決できない．また，パーミッションを拒否してもアプリケーションを実行できるため，アプリケーションによっては，正常に動作しない可能性があるという課題がある．

3.1 Linux Kernel における対策

文献[8]では，アプリケーションにより送信される HTTP パケットを Linux Kernel が独自の情報フロー制御アプリケーションに転送し，そのパケットを解析することで，個人情報や位置情報等の機密情報が含まれていた場合に通信制御を行う提案をしている．この提案は，アプリケーションの動作ではなく，送信されるパケットに注目した提案であるので，2.2 節における課題 1, 課題 2 は解決できない．さらに, Linux Kernel を改造する必要があるため，システム導入の敷居が高いという課題がある．また，パケット内の情報を解析する必要があるため，暗号化されたパケットへはこの提案方式は適用できない．

4 提案方式

Android では，必要以上にパーミッションを要求し，情報の漏洩等を引き起こす不正アプリケーションが存在する．また，広告表示のため等の正当な理由で，本来の目的以上のパーミッションを要求する正規アプリケーションも存在するため，これらを区別することは難しい．そのため，不正アプリケーション自体をシステム側で検知することは困難である．そこで，Android をターゲットにしたマルウェアをシステム側で対策を施すには限界があると考え，本提案ではユーザの補助を行うような方式とした．本提案は，アプリケーション層で実装を行っている．そのため，Android のバージョンや機種に関わらず，提案方式を適用できる．また，提案方式を Android 端末に適用することにより，2.2 節で挙げた課題 1，課題 2 を同時に解決できる．

4.1 提案方式の動作

本提案では，Android 上で動作するアプリケーションによる位置情報等の機密情報の利用や外部サーバへの送信をユーザに対して可視化する．その結果，個々のユーザが判断した要注意アプリケーションの情報をサーバに蓄積する．この情報をもとに，複数ユーザのアプリケーションインストール時における安全性の判断を補助する．

図 2 に提案方式の動作を示す．本提案では，アプリケーションが位置情報等を利用する際に出力される固有のログを定義ログとして予め定義する．Android では，アプリケーションやシステムのログをリングバッファに一時的に保存する．本提案では，このバッファに保存されたログを，Logcat コマンドを用いて参照をする．この際，参照したログが定義ログと一致する場合，位置情報等の機密情報を利用された，または送信されたとみなす．そして，その旨を簡単なメッセージを一時的にユーザに表示する機能であるトーストを利用して表示する．ユーザはその表示が意図していない内容であった場合，そのアプリケーションを要注意アプリケーションとしてサーバに通知する．サーバには，要注意アプリケーション情報を蓄積して，複数のユーザで情報を共有する．新たにアプリケーションをインストールする際に，このサーバに蓄積された情報を参考にすることにより，これからインストールするアプリケーションが他のユーザからどのように思われて

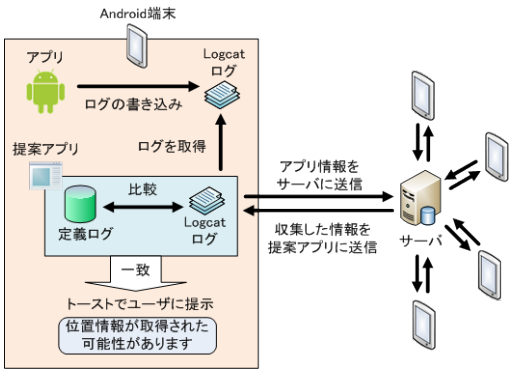


図 2 提案方式の概要

表 2　関連研究との比較

	課題 1	課題 2
提案方式	**解決できる**	**解決できる**
文献[3]	解決できない	解決できない
文献[4]	解決できない	解決できない
文献[5]	解決できない	**解決できる**
文献[6]	**解決できる**	解決できない
文献[7]	解決できない	解決できない
文献[8]	解決できない	解決できない

```
2012-09-16 22:30:14.323 D 280/GpsLocationProvider: setMinTime 0
2012-09-16 22:30:14.323 D 280/GpsLocationProvider: startNavigating
2012-09-16 22:30:14.343 D 280/GpsLocationProvider: Acquiring wakelock
2012-09-16 22:30:42.463 D 280/GpsLocationProvider: TTFF: 28122
2012-09-16 22:30:45.993 W 18051/KeyCharacterMap: Can't open keycharmap file
2012-09-16 22:30:45.993 W 18051/KeyCharacterMap: Error loading keycharmap
file '/system/usr/keychars/SH_touchpanel.kcm.bin'.
hw.keyboards.65541.devname='SH_touchpanel'
2012-09-16 22:30:46.003 W 18051/KeyCharacterMap: Using default keymap: /
system/usr/keychars/qwerty.kcm.bin
```

図 3　GPS 利用時の Logcat ログ

いるのかを知ることができる．その結果，アプリケーションのインストールを控えるなど，ユーザ自身で不正なアプリケーションへの対策をとることができる．また，本提案ではユーザ自身がトーストで表示された内容を確認して，その表示内容を適切であるか，不適切であるかを判断しなければならない．そのため，ある程度，ユーザのセキュリティ意識を高めることも期待できる．

4.3 関連研究との比較

Android はパーミッション機構によってセキュリティを確保している．しかし，2.2 節で挙げたようなパーミッションの課題は Android のセキュリティ上の課題として存在する．

本提案では，アプリケーションが位置情報等の機密情報を利用したことをユーザに見える形で表示するため，課題 1 を解決できる可能性がある．また，本提案ではユーザによる予想が難しい，パーミッションの組み合わせによるアプリケーションの挙動を可視化することができるため，課題 2 を解決できる可能性がある．以上のように，本提案を採用することで，パーミッション機構の課題を解決できる可能性がある．

2.2 節における課題を関連研究と提案方式がそれぞれ解決できるか比較を行った．表 2 に関連研究と提案方式の比較結果を示す．文献[5]が課題 2 を，文献[6]が課題 1 を解決しているが，2 つ同時には解決できていない．提案方式では 2.2 節で挙げた課題を同時に解決している．

4.2 Logcat ログの調査

本提案では Logcat ログから機密情報の利用や送信などを検知する．今回は位置情報利用時にどのようなログが出力されるのかを調査し，提案方式における定義ログを決定した．

出力されるログには，アプリケーションのデバッグの際に開発者が出力させるようにしたログと，システムから出力されるログの 2 種類がある．前者は，アプリケーション開発者に依存するため，後者に注目して出力ログの解析を行った．

今回，Logcat ログの確認をした実機は Android2.2 を搭載したシャープ株式会社の IS03 である．図 4 は GPS 情報を取得するアプリケーションを実行した際に確認できたログの一部である．図 4 の下線部の GpsLocationProvider とは，GPS を利用する際に使用されるクラスである．Android のバージョンと機種を，Android4.0 を搭載した HTC Corporation の HTC J に変更しても同様のログを得られたことから，GPS の利用を検知するには GpsLocationProvider に関するログを定義ログとして利用できると考えられる．

5　まとめ

本提案では，Logcat ログから位置情報等の利用や外部サーバへの送信を検知し，その挙動をトーストで可視化する．その情報から，ユーザ自身に要注意アプリケーションかどうかの判断を行わせ，要注意と判断したならばユーザはサーバに報告し，その情報を蓄積する．蓄積した情報をユーザ間で共有することにより，新たにインストールするアプリケーションの正当性を判断するユーザを補助するシステムを提案した．

今後は，位置情報以外の定義ログを決定するために有用なログの調査をしつつ，提案方式の有効性を確認するための実装を行う予定である．

参考文献

[1] McAfee 脅威レポート，2012 年第 2 四半期 http://www.mcafee.com/japan/media/mcafeeb2b/international/japan/pdf/threatreport/threatreport12q2.pdf

[2] McAfee 最新ウイルス一覧 http://www.mcafee.com/japan/security/latest.asp

[3] 竹森，他，”Android 携帯電話上での情報漏洩検知”, The 2011 Symposium on Cryptography and Information Security Kokura, Japan, Jan.2011.

[4] 竹森，他，“Android フォンのアプリ管理 ～ホワイトリスト方式～”，Vol.2011-CSEC-53 No.2, May.2011.

[5] 松戸, 他,”Android OS 上でのアプリケーション導入時におけるセキュリティ助言システムの提案”, Vol.2012-CSEC-56 No.12, Feb.2012.

[6] 川端，他，“Android OS における機能や情報へのアクセス制御機構の提案”，Computer Security Symposium 2011, Oct.2011.

[7] Mohammad, other, “Apex:Extending Android Permission Model and Enforcement with User-defined Runtime Constraints”, ACM Symposium on Information, Computer and Communications Security, 2009.

[8] 葛野，“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”，Computer Security Symposium 2011, Oct.2011.

[9] 林，他，“Android アプリケーション動作時に安全性を高める動的制御に関する検討”，情報処理学会第 74 回全国大会，6Z-6, March.2012.

[10] 上松，他，“Android OS におけるマスカレーディングポインタを用いたプライバシー保護”，Vol.2012-CSEC-57 No.18, May.2012.

[11] William, other, “TaintDroid:An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones”, 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010).

Androidアプリケーションの挙動を可視化することによるセキュリティ対策

名城大学大学院　理工学研究科
戸田　尚希，鈴木　秀和，旭　健作，渡邊　晃

研究背景

- スマートフォンの普及
  - Android OSのシェアが増加
- Androidをターゲットにしたマルウェア出現
  - パーミッションを悪用したアプリケーション
- Androidにおけるマルウェアは自動感染しない
  - PCよりは安全
  - それでも，マルウェアに感染してしまう
- ユーザのセキュリティ意識が低い

研究目的

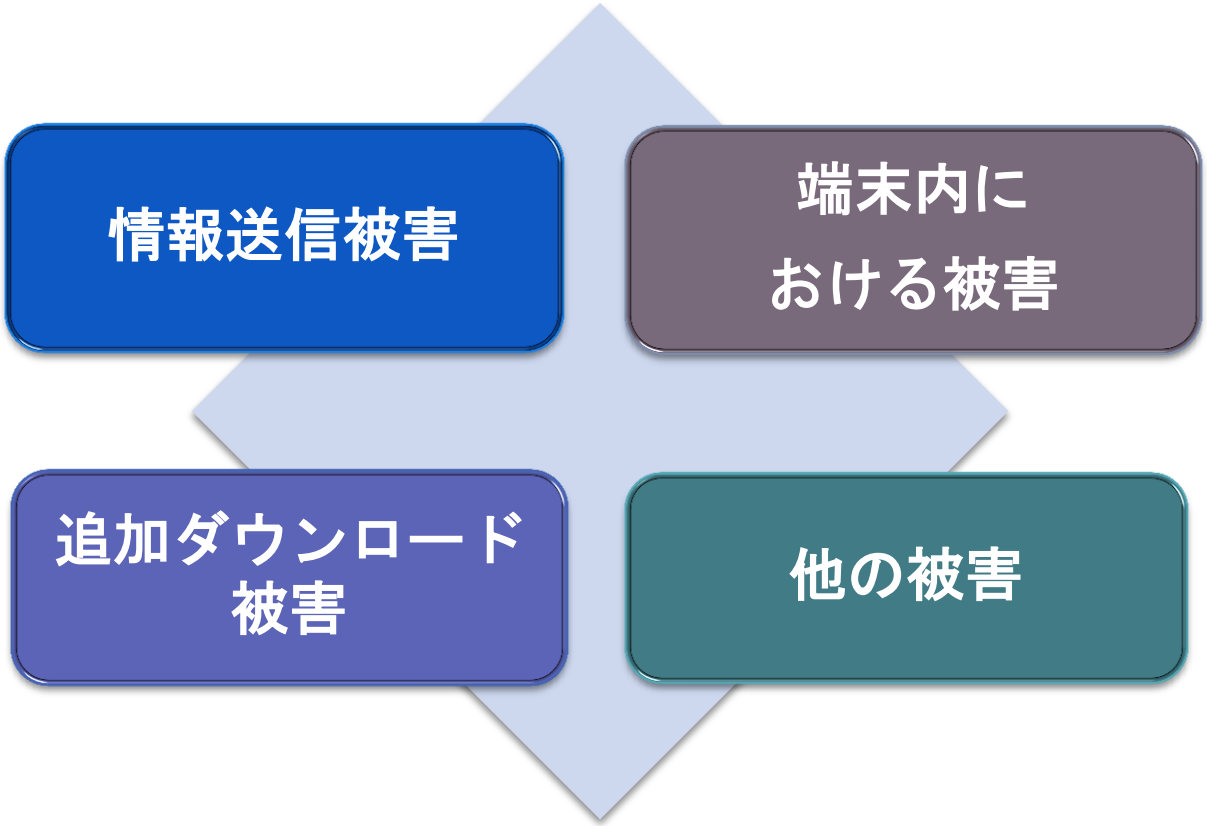
- Androidにおける問題点
  - パーミッション機構の課題（システム側の問題）
  - ユーザのセキュリティ意識が低い（ユーザ側の問題）
- 関連研究では…
  - システム側のセキュリティ対策が主流

目的

システム側，ユーザ側，双方の問題を解決することで，
Androidセキュリティを向上させる

マルウェア被害の分類

- McAfeeのホームページを参考にマルウェアが及ぼす被害を調査
- 4つに被害を分類



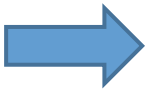
マルウェアによる被害 (1)

	被害内容
情報送信被害	プレミアムSMS番号へのSMS送信
	位置情報を外部サーバへ送信
	Android_id，電話番号，連絡先リストを外部サーバに送信
	写真撮影とアップロード
端末内における被害	ネットワーク設定の変更
	他の実行中のプロセスの強制終了
	通話内容を記録してSDカードに保存
	端末の再起動
追加ダウンロード被害	アプリケーションを/systemディレクトリにインストール
	APKファイルのインストール・アンインストール
他の被害	ルート権限の取得
	完全自動で攻撃者のWebサイトにアクセス

参考：http://www.mcafee.com/japan/security/latest.asp

マルウェアによる被害 (2)

- 被害の分類から分かったこと
  - 情報送信被害，端末内における被害を及ぼすマルウェアが多い
- ユーザにとって深刻な被害は？



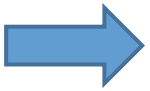
情報送信被害

理由

送信情報が犯罪に利用される恐れがある

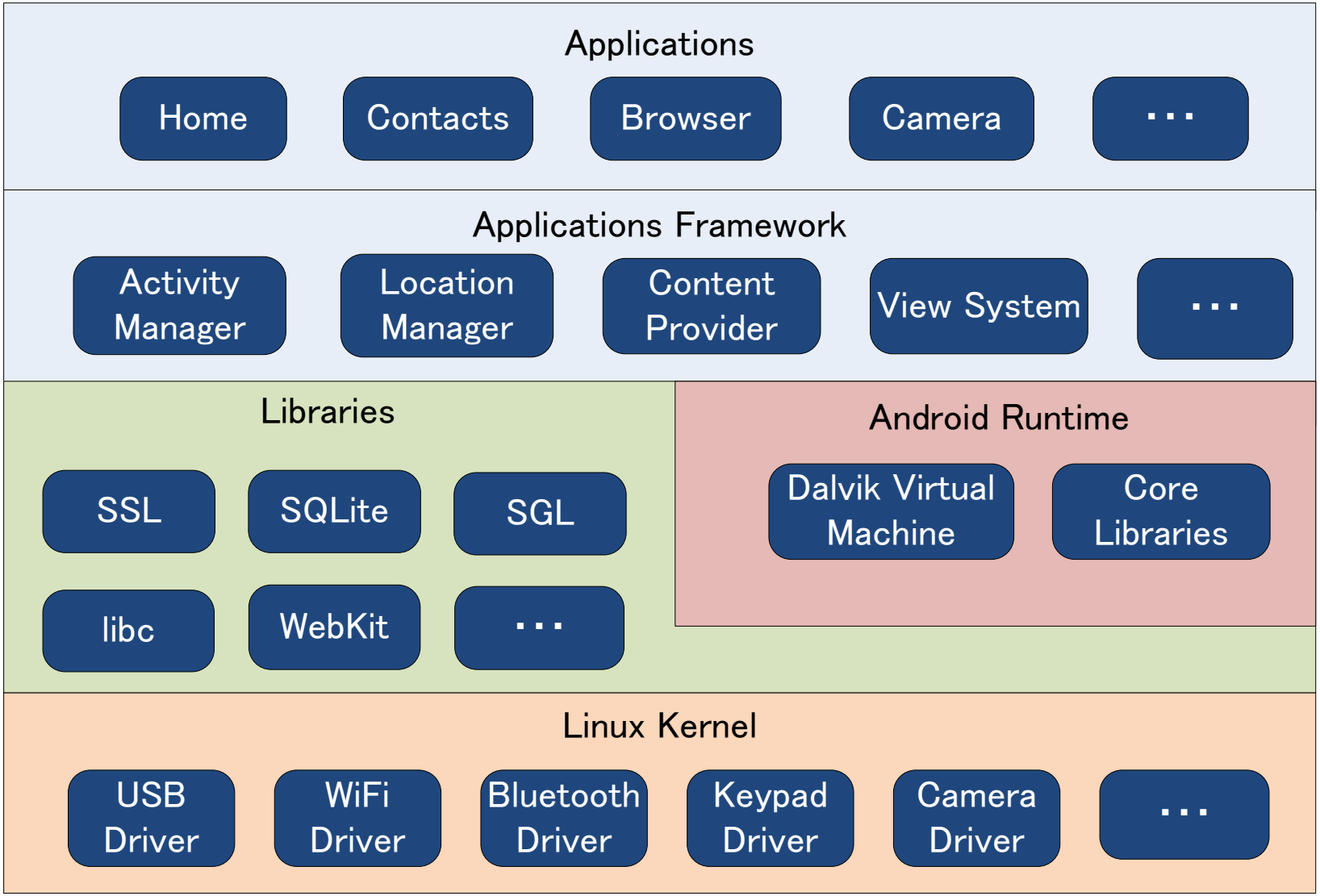
- 情報漏洩
  - ユーザの気付かないところで起こる被害

重要情報の取得 ＋ 端末外への送信

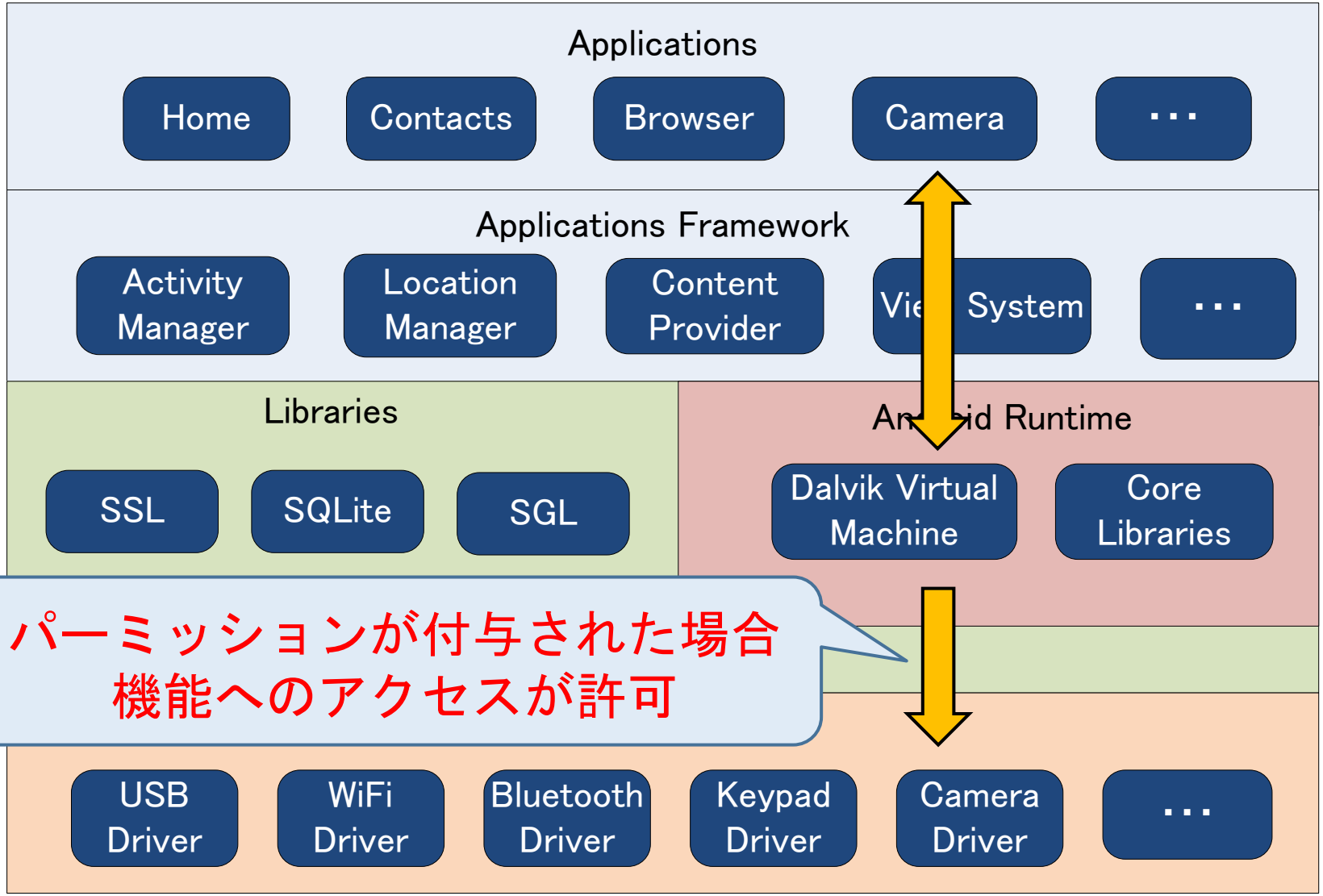


片方でも気付くことが出来れば，対策が打てる

Androidの仕組み

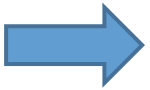


Androidの仕組み



パーミッションとは？

- Androidのセキュリティ機構
- アプリケーションが必要とする権限(パーミッション)を開発者が明示し，ユーザに承認を求める仕組み



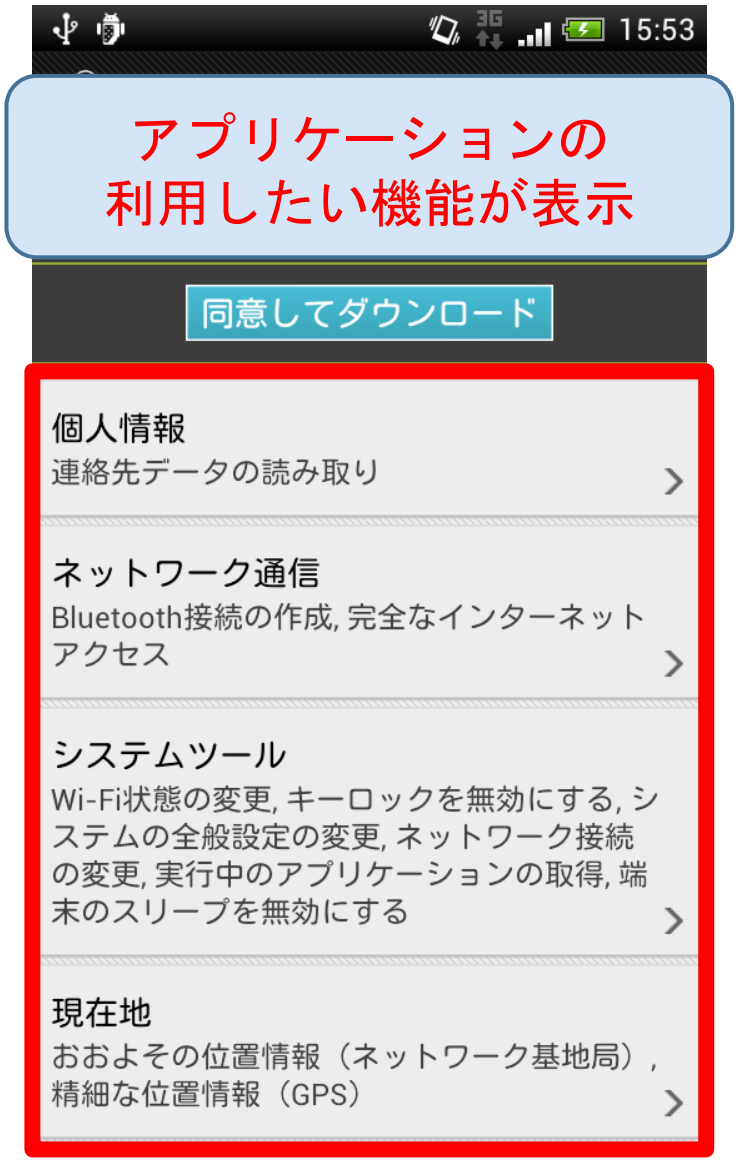
承認した場合にのみインストール可能

- パーミッションの例

パーミッション	用例
ACCESS FINE LOCATION	GPSを利用した位置情報へのアクセス
READ CONTACTS	アドレス帳の読み込み
INTERNET	インターネットアクセス
SEND SMS	SMSの送信

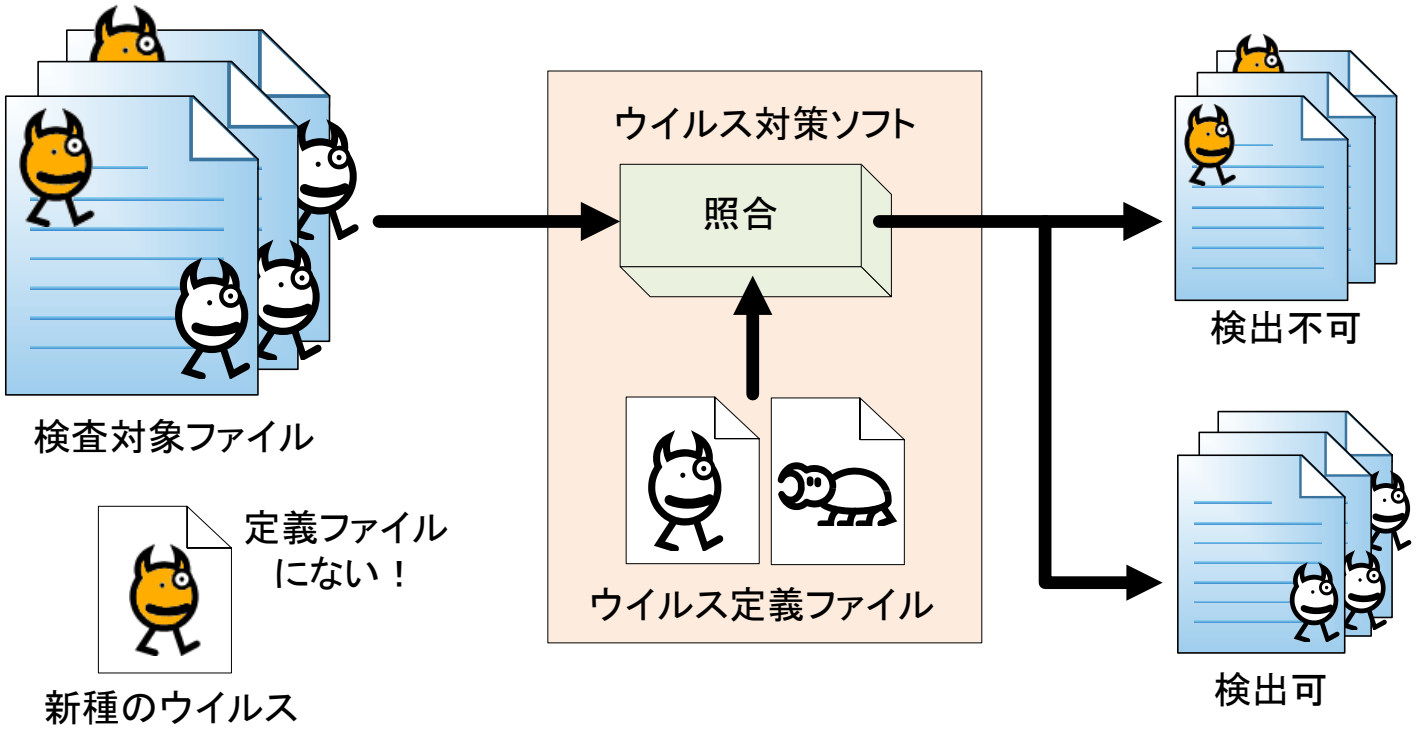
パーミッションの課題

- GPS等の情報がいつ利用されたのかユーザは知ることが出来ない
- 表示されるパーミッションからアプリケーションの詳細な動作はわからない



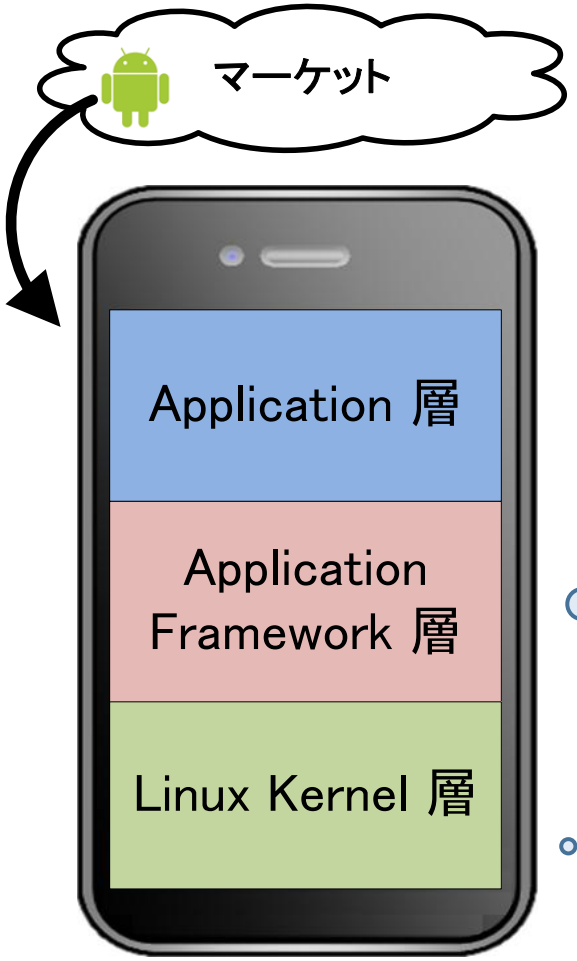
既存技術

- Android向けのウイルス対策ソフト
  - ノートンモバイルセキュリティ，McAfee Mobile Security等
  - パターンマッチングがウイルス検出の主流



新種・亜種のウイルスに対応できない

関連研究

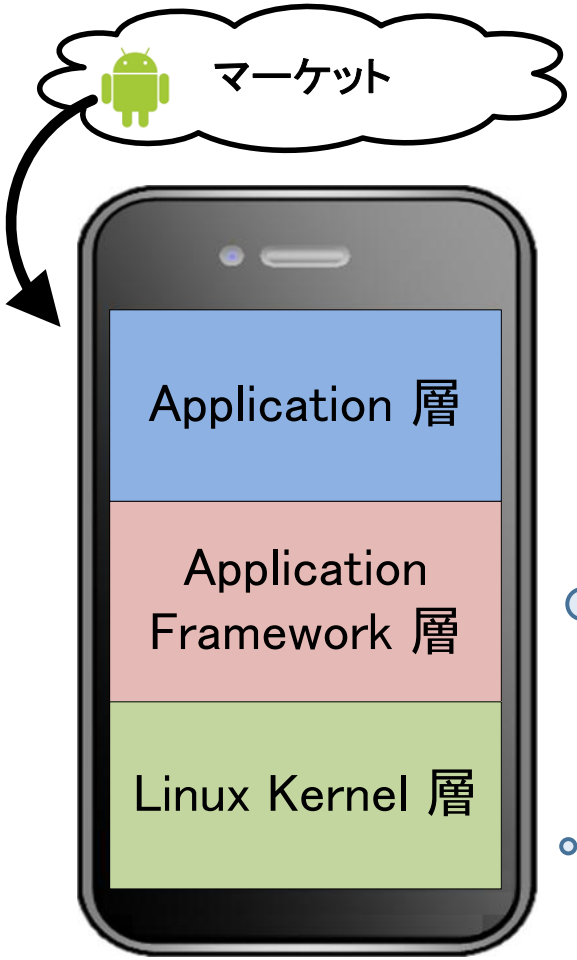


1. インストール時にアプリの安全性を助言する提案
2. ホワイトリストを用いてアプリをチェックする提案

3. パーミッションの動的承認を可能にする提案
4. アプリに機密情報そのものを保持させない提案
5. パーミッションの個別承認，条件付承認を実現する提案

6. 端末情報を持つHTTPパケットの通信制御を行う提案

関連研究



1. インストール時にアプリの安全性を助言する提案
2. ホワイトリストを用いてアプリをチェックする提案
3. パーミッションの動的承認を可能にする提案
4. アプリに機密情報そのものを保持させない提案
5. パーミッションの個別承認，条件付承認を実現する提案
6. 端末情報を持つHTTPパケットの通信制御を行う提案

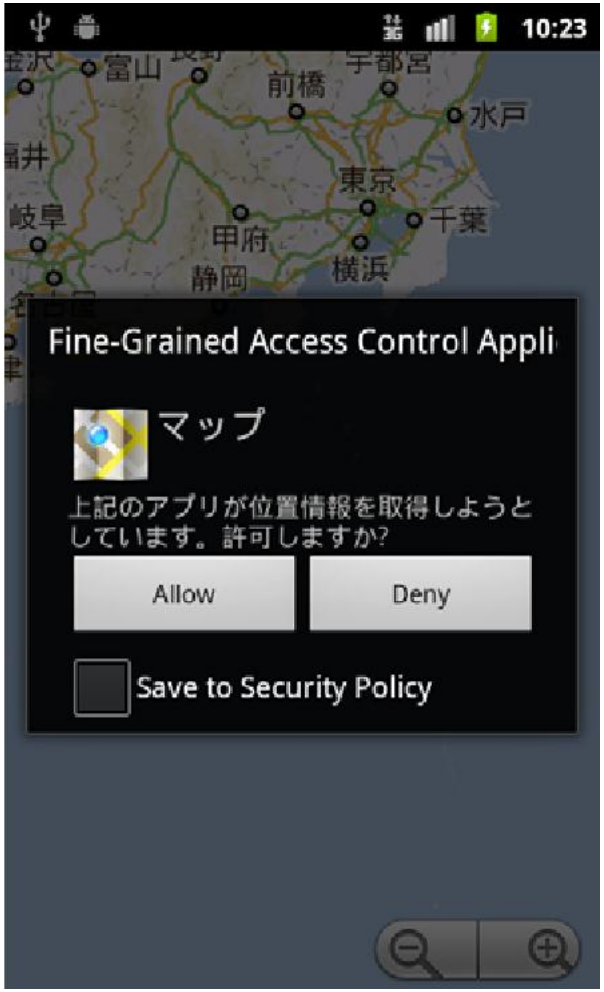
Application層での対策

- インストール時にアプリケーションの安全性を助言する提案
  - 危険性検知と危険性提示を実現
  - パーミッションから危険性を検知し，アイコンで表示
  - マーケットでの評判やダウンロード数を表示
- 利点
  - アイコンを用いてアプリケーションの動作の予測をユーザに示すことが出来る
- 欠点
  - GPS等の情報がいつ利用されたのかユーザは知ることが出来ない

参考：松戸（岩手県立大学），他：Android上でのアプリケーション導入時におけるセキュリティ助言システムの提案，Vol.2012-CSEC-56,No.12,Feb.2012

Application Framework層での対策

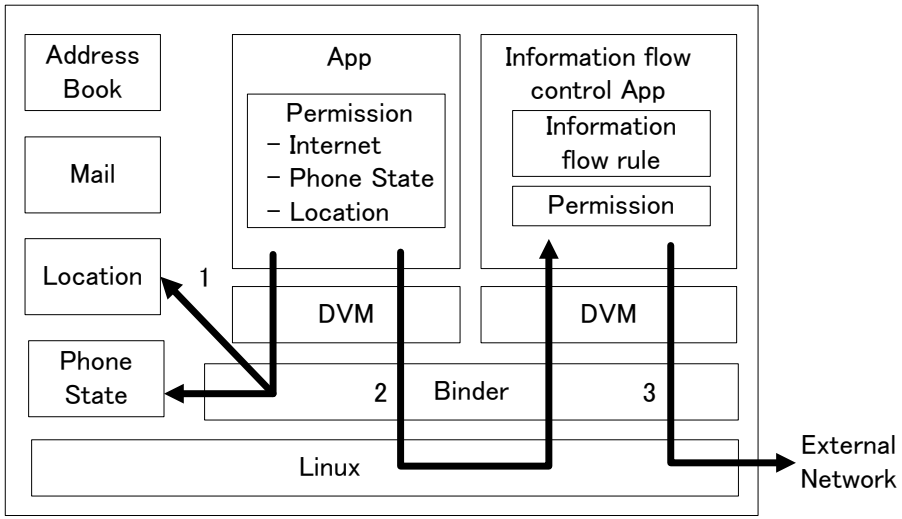
- パーミッションの動的承認を可能にする提案
  - パーミッションで保護されているAPIをフックすることで実現
  - セキュリティポリシーの設定により承認ダイアログを省略
- 利点
  - GPS等の情報がいつ利用されたのかユーザは知ることが出来る
- 欠点
  - インストール時にアプリケーションの詳細な動作をユーザに示せない



参考：川端（KDDI研究所），他：Android OSにおける機能や情報へのアクセス制御機構の提案，Computer Security Symposium 2011.

Linux Kernel層での対策

- 端末情報を持つHTTPパケットの通信制御を行う提案
  - HTTPパケットの送信に注目した情報漏洩対策
  - パケットをLinux Kernelで独自アプリケーションへ転送
  - パーミッションをもとに生成された情報フロールールに基づき通信を制御
- 利点
  - 情報漏洩自体を防ぐことが出来る
- 欠点
  - パーミッション機構の問題を解決できない



参考：葛野（セコム株式会社IS研究所）：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，Computer Security Symposium 2011.

提案方式

アプリケーション挙動の可視化と
アプリケーションインストール時のユーザ補助

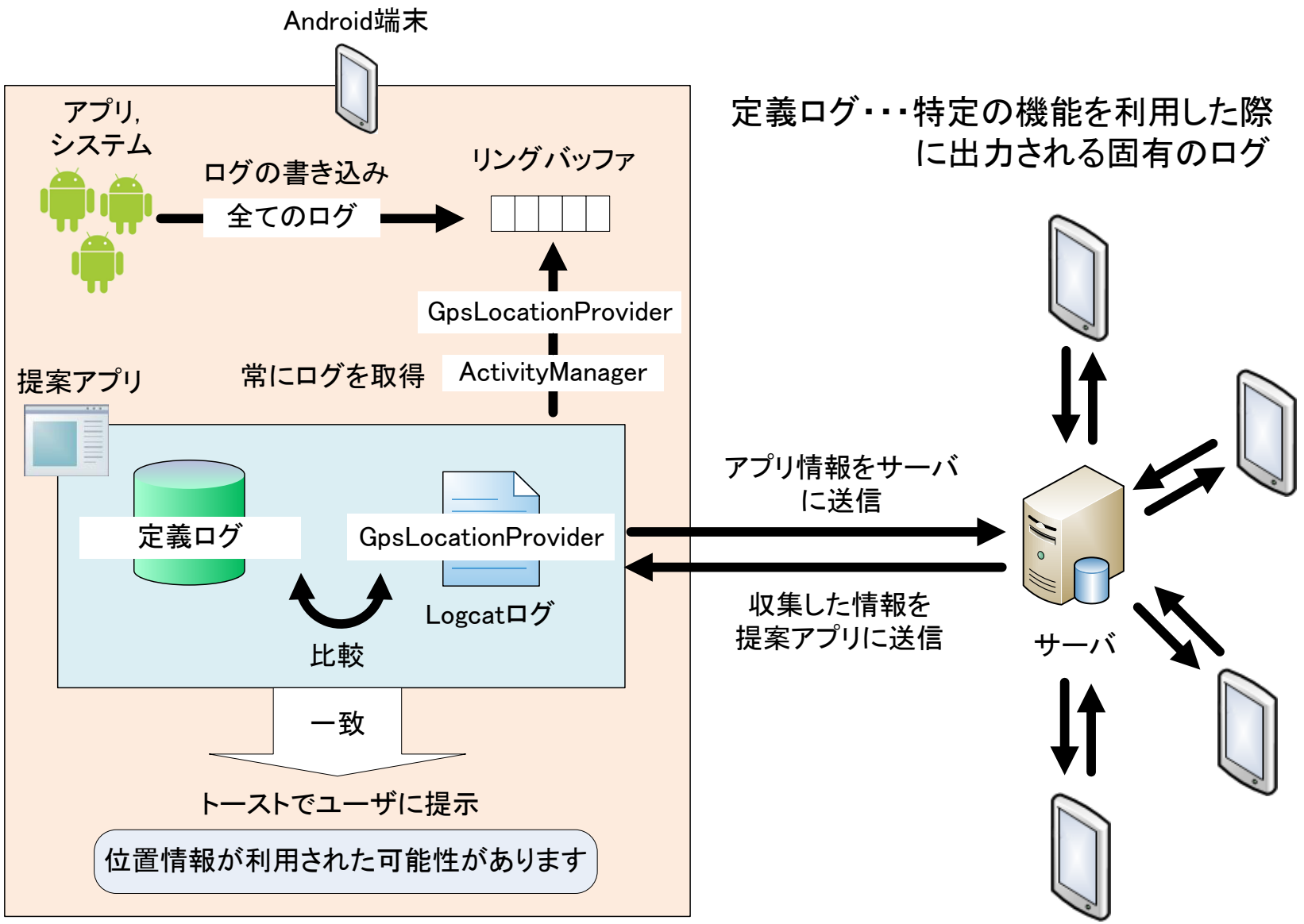
- 挙動の可視化
  - ユーザのアプリケーションの正当性の判断補助
  - ユーザのセキュリティ意識の向上
- ユーザ補助
  - 要注意アプリケーションの情報共有
  - アプリケーションの安全性の判断補助

本提案では挙動の検知にLogcatログを利用

Logcatログ

- アプリケーションやシステムがリングバッファに一時的に保存するログ
- Logcatコマンドを利用すればログの参照が可能
- Logcatログの種類
  - アプリケーション開発者が出力させたログ
  - システムが出力するログ

提案方式の動作概要


Android端末
アプリ,
システム
ログの書き込み
全てのログ
リングバッファ
GpsLocationProvider
常にログを取得
ActivityManager
提案アプリ
定義ログ
GpsLocationProvider
Logcatログ
比較
一致
トーストでユーザに提示
位置情報が利用された可能性があります
定義ログ・・・特定の機能を利用した際
に出力される固有のログ
アプリ情報をサーバ
に送信
収集した情報を
提案アプリに送信
サーバ

定義ログ

Android端末の機能の利用や外部への送信の際に
出力される固有のLogcatログを知る必要がある

- 定義ログの決定
  - GPSを利用した際に出るLogcatログを調査
  - GPSの利用を検知
- 使用実機

(1) ISO3, Android 2.2,シャープ株式会社
(2) HTC J(ISW13HT),Android 4.0,HTC Corporation

定義ログの決定 (1)

- IS03のLogcatログ
  - GpsLocationProviderに注目

TTFF: GPS初期位置算出時間

```
2012-09-16 22:30:14.323 D 280/GpsLocationProvi      setMinTime 0
2012-09-16 22:30:14.323 D 280/GpsLocationProvide    startNavigating
2012-09-16 22:30:14.343 D 280/GpsLocationProvider: Acquiring wakelock
2012-09-16 22:30:42.463 D 280/GpsLocationProvider: TTFF: 28122
2012-09-16 22:30:45.993 W 18051/KeyCharacterMap: Can't open keycharmap file
2012-09-16 22:30:45.993 W 18051/KeyCharacterMap: Error loading keycharmap file
'/system/usr/keychars/SH_touchpanel.kcm.bin'. hw.keyboards.65541.devname='SH_touchpanel'
2012-09-16 22:30:46.003 W 18051/KeyCharacterMap: Using default keymap:
/system/usr/keychars/qwerty.kcm.bin
2012-09-16 22:30:46.313 D 280/GpsLocationProvider: stopNavigating
2012-09-16 22:30:46.373 D 280/GpsLocationProvider: Releasing wakelock
```

定義ログの決定 (2)

- HTC JのLogcatログ
  - IS03と同様のログが確認

```
2012-09-24 13:02:39.504 D 434/GpsLocationProvider: setMinTime 0
～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～
2012-09-24 13:02:39.514 D 434/GpsLocationProvider: startNavigating
～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～
2012-09-24 13:02:45.741 D 434/GpsLocationProvider: TTFF: 6204
2012-09-24 13:02:45.741 D 434/GpsLocationProvider: [GICON] +++++
GPS_FIX_CHANGE_ACTION +++++  (true)
～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～
2012-09-24 13:02:46.692 D 434/GpsLocationProvider: In
handleEnableLocationTracking. enable false
2012-09-24 13:02:46.692 D 434/GpsLocationProvider: stopNavigating
```

GpsLocationProviderを定義ログとして決定

サーバで管理する情報

- 要注意アプリケーションの情報
  - アプリケーション名
  - パッケージ名
  - パーミッション情報
  - ユーザが不審に思った動作

ユーザからの情報提供
により得られる情報

- マーケットから得られる情報
  - ダウンロード数
- その他
  - 要注意アプリケーションの報告件数

サーバ自身の処理
により得られる情報

上記の情報をサーバで管理し，
提案アプリに対して提供する

サーバで管理する情報

- 要注意アプリケーションの情報
  - アプリケーション名
  - パッケージ名
  - パーミッション情報
  - ユーザが不審に思った動作

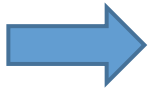
- マーケットから得られる情報
  - ダウンロード数
- その他
  - 要注意アプリケーションの報告件数

上記の情報をサーバで管理し，
提案アプリに対して提供する

情報を提供された提案アプリのイメージ図

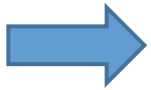
提案方式のポイント

- アプリケーション挙動の可視化
  - GPS等の情報がいつ利用されたのかユーザは知ることが出来ない（システム側の問題）



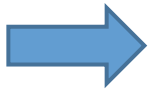
GPS等の機能を利用をトーストでユーザに提示することで解決

- ユーザによるアプリケーション挙動の審査
  - ユーザのセキュリティ意識が低い(ユーザ側の問題)



アプリケーションの正当性について考える機会をユーザに与えることで解決

- 要注意アプリケーション情報の共有
  - 表示されるパーミッションからアプリケーションの詳細な動作はわからない（システム側の問題）



インストール前に要注意アプリケーション情報を確認することで解決

まとめ

- 提案方式で実現できること
  - パーミッション機構の課題を解決
  - ユーザのセキュリティ意識の向上
  - アプリケーション実行時の正当性判断の補助
  - インストール時の安全性判断を補助
- 今後の課題
  - 位置情報以外の定義ログとして使えるログの調査
  - 提案方式の実装