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の提案

123430029 戸田 尚希
渡邊研究室

1. はじめに

スマートフォンの普及に伴い，多くの人がインターネットを利用する機会が増えた．中でも Android が搭載されたスマートフォンが急速な普及をみせている．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．Android 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．

本稿では，ユーザによるマルウェアの発見やインストール時にアプリケーションの安全性の判断を補助するシステムを提案する．提案システムは，アプリケーションによる GPS 等の機能の利用や個人情報の外部サーバへの送信を可視化することにより，ユーザによるマルウェアの発見を可能とする．可視化された情報をもとに，ユーザはアプリケーションの正当性を判断し，必要であればアンインストールする．また，ユーザが危険であると判断したアプリケーションは，要注意アプリケーションとして，その情報をサーバに送信して蓄積する．サーバに蓄積された情報は，アプリケーションをインストールする時に確認することができる．これにより，これからインストールするアプリケーションの安全性の判断を補助することができる．

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．表 1 は，MacAfee のホームページを参考に独自の判断で分類したものである．マルウェアが及ぼす被害は，情報送信被害，端末内における被害，ダウンロード被害，その他の被害に分類できる．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．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．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．その他の被害とは，上記分類に当てはまらない被害である．上記の分類の中では，情報送信被害と端末内における被害の原因となるマルウェアが数多く報告されている．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，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からである．このような情報が外部へ漏れることにより犯罪などに利用される可能性もある．そのため，情報送信被害を発見し，被害を防ぐことが Android 端末における最大の課題を解決することに繋がる．

表 1: マルウェアによる被害の分類

被害の種類	構成要素
情報送信被害	IMEI，IMSI をサーバに送信
	SMS を外部サーバに送信
	プレミアム SMS への SMS 送信
	位置情報をサーバに送信
	連絡先リストをサーバに送信
	写真をサーバに送信
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
	特定の SMS メッセージの削除
	通話内容を SD カードに保存
	端末の再起動
ダウンロード被害	アプリを/system にインストール
	アプリをアンインストール
その他の被害	SMS メッセージの監視
	ルート権限の取得

2.2 パーミッション機構の課題

Android にはセキュリティ面を考慮して，パーミッション機構という仕組みがある．パーミッション機構とは，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一覧をユーザに表示する仕組みである．この仕組みは，マルウェア対策に特化した仕組みではないため，マルウェアに関してはセキュリティ対策が不十分である．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．

- インストール時にパーミッションから，マルウェアを発見するのは難しい．(課題 1)
- インストールしたマルウェアを見つけ，被害の拡大を防ぐことは難しい．(課題 2)

Android において，マルウェアの感染を防ぎ，被害を食い止めることができる可能性があるのは，アプリケーションのインストール時である．マルウェアを感染前に防ぐことができれば，Android におけるセキュリティ対策の中で最も効力のある対策である．しかし，現状のパーミッション機構では，ユーザがその表示からアプリケーションの動作を予想してマルウェアであるかを判断するのは困難である．また，アプリケーションインストール後については，マルウェアを検知して対策を施すことは行われない．そのため，マルウェアが一旦インストールされてしまえば，GPS 等の機能の利用や個人情報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．

2.3 市販のセキュリティ対策ソフトの課題

市販のセキュリティ対策ソフトの課題としては以下の 2 点が挙げられる．

- 新種や亜種のマルウェアを検出が不十分であること．
  パターンマッチングがマルウェア検出の主流であるため，定義ファイルに定義されていないマルウェアは検出できない．また，ヒューリスティック検知においても検出できないマルウェアが存在する．
- 保存先によってはセキュリティ対策ソフトのスキャンができないこと．
  一般権限で動作するセキュリティ対策ソフトは「/data/app」もしくは「SD カード」しかスキャンできない．そのため，「/data/app private」や「/system」におけるマルウェアに関しては，検出・削除ができない．

3. 提案方式

Android では，正規アプリケーションとマルウェアの区別が難しい．そのため，Android を対象にしたマルウェアをセキュリティ対策ソフト等の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方式とした [1]．本提案では，市販のセキュリティ対策ソフトと併用することにより，2.2 節で述べた課題 1，課題 2 を解決する．

3.1 提案方式の動作概要

図 1 に提案方式の動作概要を示す．本提案では，アプリケーションが GPS 等の機能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 には，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機能がある．このバッファに保存されたログを，ビジュアル化プログラムが Logcat コマンドを用いて参照する．この際，参照したログが定義ログと一致する場合，GPS 等の機能が利用された，または個人情報が送信されたとみなし，その旨をトーストにて表示する．トーストとは，画面に一時的に表示するユーザ通知機能である．トーストには，アプリケーションの挙動の内容とそのアプリケーション名を表示する．ユーザはその表示が意図していない内容であった場合，アプリケーションを削除すべきかどうかを判断する．同時に，ユーザが要注意であると判断したアプリケーション情報をサーバに送信する．

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．ユーザは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，ユーザ自身でマルウェアへの対策をとることができる．

3.2 定義ログの決定

本提案では，Android 端末にインストールされているアプリケーションの挙動をアプリケーションレベルで検知する方法として Logcat ログに着目した．アプリケーション挙動の中でも GPS とカメラを利用する際に出力される Logcat ログを調査し，提案方式における定義ログを決定した．定義ログとして決定した GPS 利用時に出力される Logcat ログを図 2 に示す．

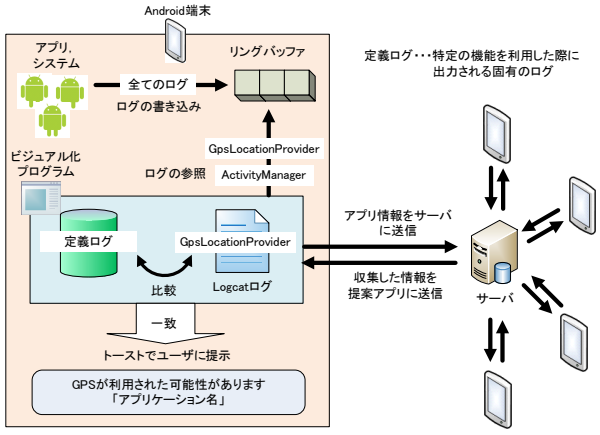


図 1: 提案方式の動作概要

```
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'
```

図 2: GPS 利用時に出力される Logcat ログ

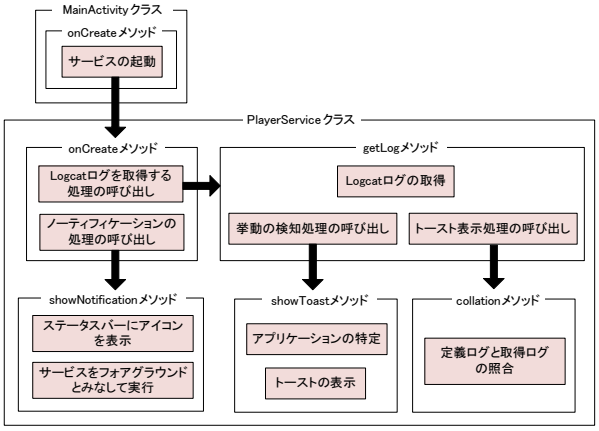


図 3: ビジュアル化プログラムの構成

4. 実装と検証

今回は，GPS の利用とカメラの利用をユーザに通知する実装を行った．本提案が実装された Android4.0 を搭載した HTC Corporation の ISW13HT で，GPS とカメラの機能を利用したところ，それらの機能の利用を知らせるトーストが表示されることが確認できた．

Android ではメモリの不足等により，サービスとして実行中のアプリケーションが OS によって終了させられてしまうことがある．この問題を解決するために，startForeground API を利用した実装を行なった．(図 3) startForeground API は，サービスで実行されているアプリケーションをフォアグラウンドとみなして実行させることができる．Android では，フォアグラウンドとして実行されているアプリケーションは，通常のバックグラウンドで実行されるプロセスよりも優先度が高くなり，強制終了の可能性を低くすることができる．

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． 今後の課題としては，サーバの実装を進め，本提案の有効性を示す予定である．

参考文献

[1] 戸田尚希，他：”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の検討”，マルチメディア, 分散, 協調とモバイル (DICOMO2013) シンポジウム論文集，Vol.2013，No.1，pp.1348-1354，Jul.2013.

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の提案

名城大学大学院
理工学研究科　情報工学専攻
渡邊研究室
123430029　戸田　尚希

研究背景

- スマートフォンの普及
  - Android OSのシェアが増加
- Android端末をターゲットにしたマルウェアが出現
  - パーミッションを悪用したアプリケーション
- Android端末におけるマルウェアの自動感染は殆どない
  - Windows PCよりは安全
- 現在もAndroidマルウェアは増加傾向

パーミッションを悪用するマルウェアの被害を防止する必要がある

研究目的

- Androidのセキュリティにおける課題
  - システム側の課題
    - → パーミッション機構の課題
    - → 市販のセキュリティ対策ソフトの課題
  - ユーザ側の課題
    - → ユーザのセキュリティ意識が低い

目的

システム側，ユーザ側，双方の課題を解決し，
マルウェアの被害を防止する

マルウェアによる被害の分類

- McAfeeのHP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

	被害内容
情報送信被害	位置情報を外部サーバへ送信
	写真撮影と写真データを外部サーバに送信
	電話番号を外部サーバに送信
	住所録を外部サーバに送信
	SMSメッセージの取得，プレミアムSMS番号への送信
端末内における被害	通話内容を記録してSDカードに保存
	他の実行中のプロセスの強制終了
	端末の再起動
ダウンロード被害	アプリケーションを/systemにインストール
その他の被害	ルート権限の取得

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

マルウェアによる被害の分類

- McAfeeのHPを参考
- 4つに被害を分類

ユーザにとって最も深刻な被害
理由：犯罪に利用される可能性があるため

	被害内容
情報送信被害	位置情報を外部サーバへ送信
	写真撮影と写真データを外部サーバに送信
	電話番号を外部サーバに送信
	住所録を外部サーバに送信
	SMSメッセージの取得，プレミアムSMS番号への送信
端末内における被害	通話内容を記録してSDカードに保存
ダウンロード被害	アプリケーションを/systemにインストール
その他の被害	ルート権限の取得

情報の「取得」と「送信」により成り立つ

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

Androidにおけるセキュリティ対策

- パーミッション機構
  - アプリケーションが使用する機能をユーザに表示し，許可を求める仕組み
  - マルウェアに特化したセキュリティ対策ではない
- パーミッション機構の課題
  - インストール時のパーミッションからマルウェアを発見するのは難しい
  - インストールしたマルウェアを見つけ，被害拡大を防ぐことは難しい

アプリケーションが使用したい権限を表示

感染を防ぎ，感染後の被害拡大を防ぐ対策が必要

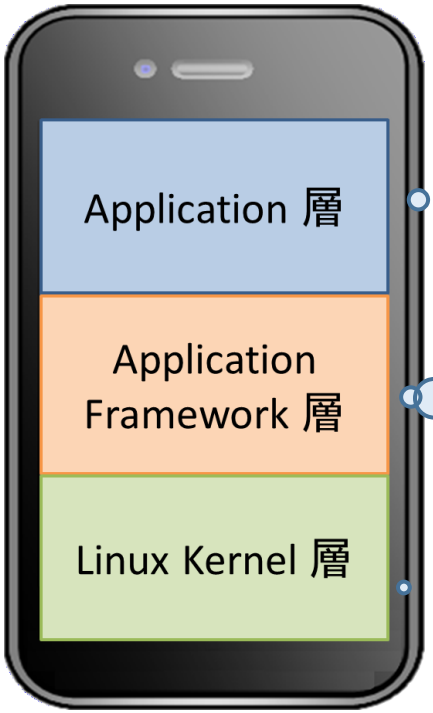
市販のセキュリティ対策ソフト

- 市販のセキュリティ対策ソフトの例
  - 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，McAfee Mobile Security等
- パターンマッチングがウイルス検出の主流
- ヒューリスティック検知と併用しているソフトもある
- 課題
  - 新種・亜種のマルウェアに対応しきれない
  - スキャンできないフォルダ，ファイルがある

関連研究

マーケット

1. アプリケーションの情報漏洩を検知する事前審査ツールの提案



2. 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
3. ホワイトリストを用いてアプリケーションをチェックする提案

4. パーミッションの動的承認を可能にする提案
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

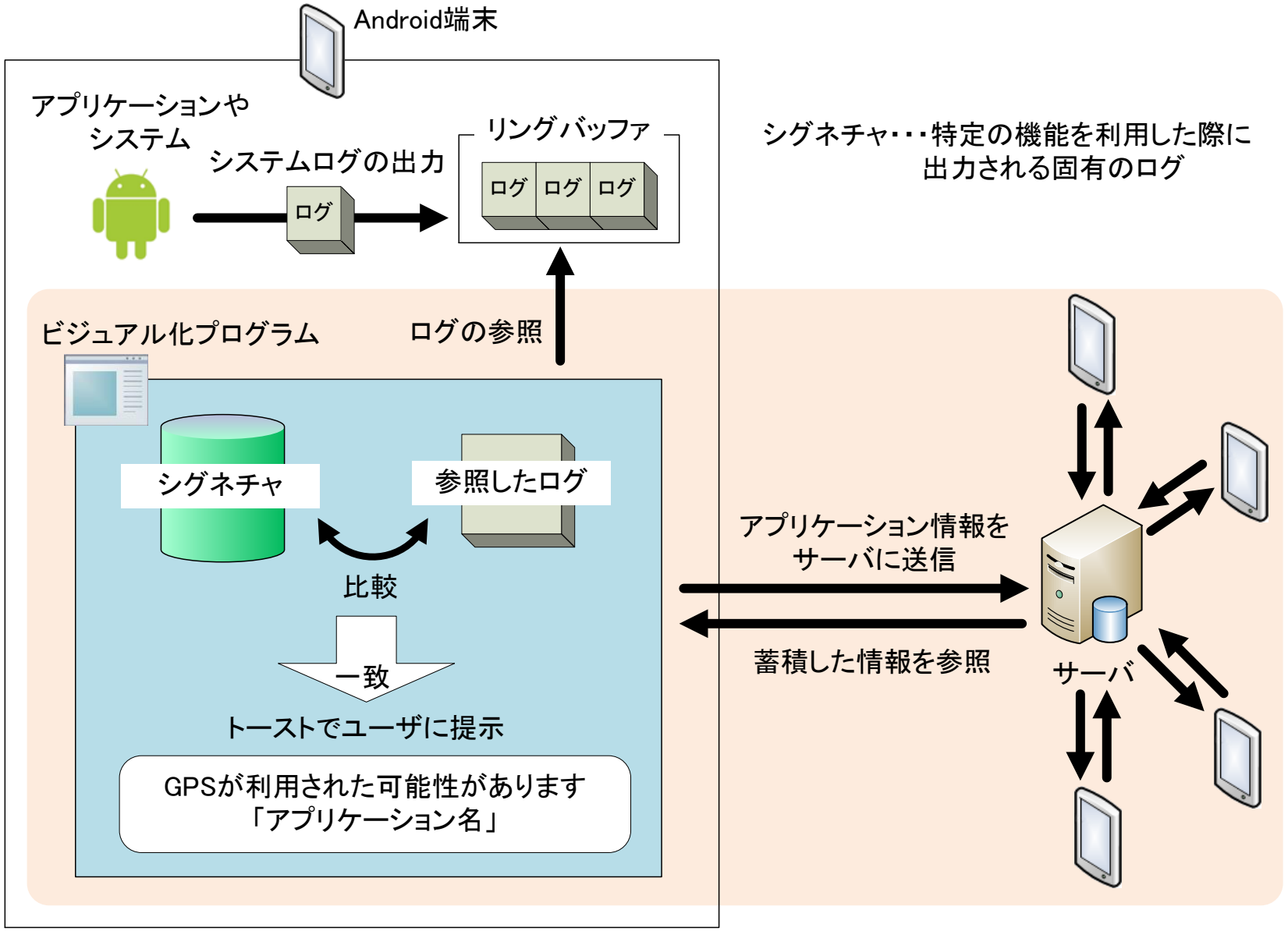
6. 端末情報を持つHTTPパケットの通信制御を行う提案

提案方式

ユーザによるマルウェアの発見や，アプリケーションの安全性の判断を補助するシステム

- マルウェアの被害拡大を防ぐ
  - アプリケーションの挙動をユーザに通知
  - アプリケーションの正当性を判断し，削除するなど対策をとる
- マルウェアの感染を防ぐ
  - 要注意アプリケーションの情報をユーザ間で共有
  - アプリケーションのインストール時に安全性判断の補助
- ユーザのセキュリティ意識の向上
  - アプリケーションの安全性を考える機会を与える
- バックグラウンドで動作するシステムとして実装

提案方式の動作概要


Android端末
アプリケーションや
システム
システムログの出力
ログ
リングバッファ
ログ ログ ログ
シグネチャ・・・特定の機能を利用した際に
出力される固有のログ
ビジュアル化プログラム
ログの参照
シグネチャ
参照したログ
比較
一致
トーストでユーザに提示
GPSが利用された可能性があります
「アプリケーション名」
アプリケーション情報を
サーバに送信
蓄積した情報を参照
サーバ

シグネチャの定義

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ログを知る必要がある

- シグネチャとして利用するログの調査
  - 情報送信被害の中でも，Android端末の機能の利用を検知するログについて調査
  - GPSとカメラの利用を検知するログを確認
- 使用実機

(1) IS03, Android 2.2,シャープ株式会社
(2) HTC J(ISW13HT),Android 4.0,HTC Corporation

シグネチャの決定

- GPSを利用した際に出力されたログ
  - Android2.2を搭載したIS03のログ

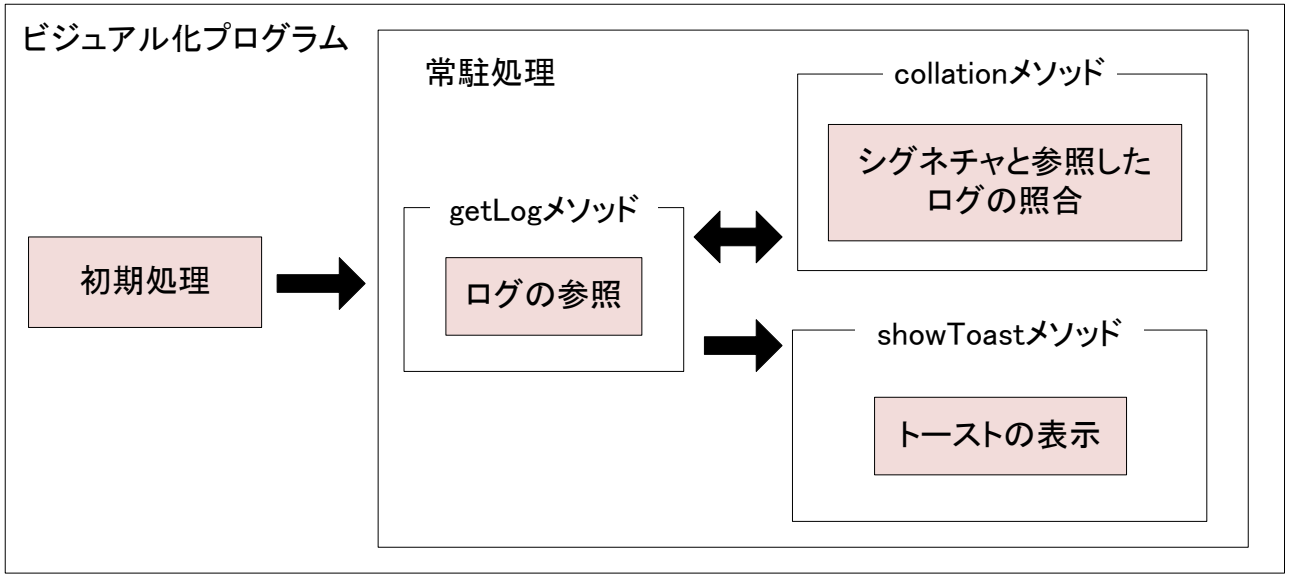
```
2012-09-16 22:30:14.323 D 280/GpsLocationProvider: setMinTime 0
2012-09-16 22:30:14.323 D 280/GpsLocationProvider: startNavigating
```

- カメラを利用した際に出力されたログ
  - Android2.2を搭載したIS03のログ

```
11-22 15:33:12.668: I/ActivityManager(280): Starting activity:
Intent{act=android.intent.action.MAIN cat=[android.intent
.category.LAUNCHER] flg=0x10200000 cmp=jp.co.sharp.android.camera/
.stillimagecamer.Camera bnds=[322,320][470,508] }
～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～～
11-22 15:33:13.204: I/ActivityManager(280): Displayed activity
jp.co.sharp.android.camera/.stillimagecamera.Camera: 473ms (total
473 ms)
```

実装

- ビジュアル化プログラムをAndroidアプリケーションとして実装



- 初期処理
  - バックグラウンドで動作するビジュアル化プログラムをフォアグラウンドとみなして実行させる処理

Androidアプリケーションの既存課題

- バックグラウンドで動作するアプリケーションは正常に実行されていない可能性がある
- その理由
  - メモリ不足等で優先度の低いアプリケーションをAndroidが強制終了してしまう
- プロセスの優先度
  - バックグラウンドで実行…優先度（低）
  - フォアグラウンドで実行…優先度（高）
- バックグラウンドで動作するアプリケーションは強制終了されやすい

提案方式ではstartForeground APIの利用により，Androidによるプロセスの強制終了を避ける

実装の検証 (2)

- ビジュアル化プログラムが正常に常駐しているかを確認
- 確認方法
  - １時間に１度ログを出力するテストプログラム
  - ログを取得し続けるビジュアル化プログラム

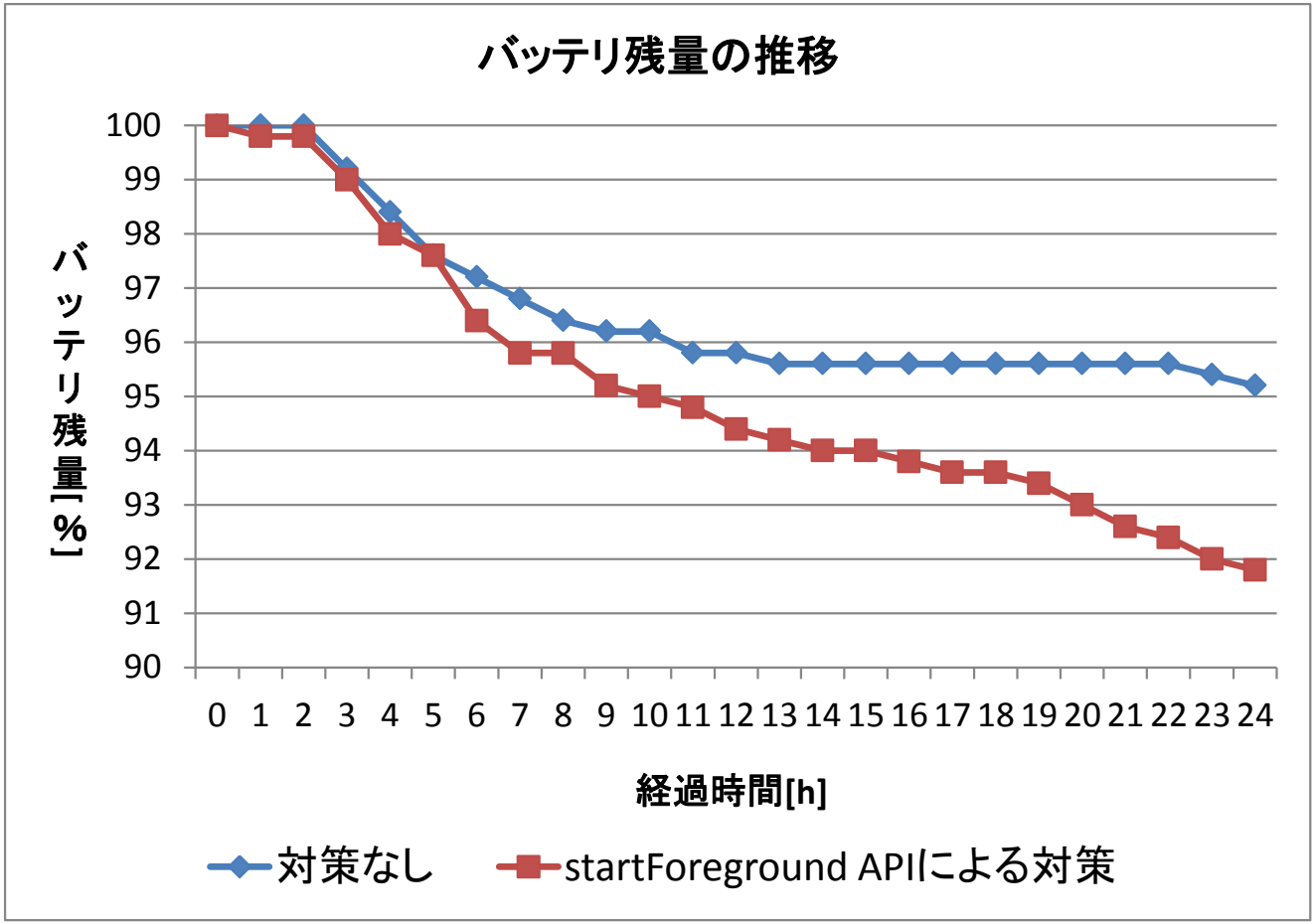
  } 同時実行
  - 実行させた期間：1日 × 5回
  - 想定される取得ログ数：25個※
- ログの取得状況をビジュアル化プログラムが正常に常駐しているかどうかの目安とした

startForeground API	ログ取得成功数	ログ取得率
実装なし	14.2個	56.8%
実装あり	23.4個	93.6%

※テストプログラム実行直後にもログが出力されるため25個

実装の検証 (3)

- 実装の違いによるバッテリ残量の推移を確認
  - 1日のバッテリ残量の推移を5回測定



まとめ

- ユーザによるマルウェアの発見や，アプリケーションの安全性の判断を補助するシステムを提案
- 提案システムの効果
  - パーミッション機構の課題を解決
  - 市販のセキュリティ対策ソフトの課題を解決
  - ユーザのセキュリティ意識の向上
- 今後の課題
  - サーバ部分の実装
  - Logcatログ以外でアプリケーション挙動を検知する方法の調査