シャープのプライバシー保護への取り組み

2013/3/15

シャープ株式会社 通信システム事業本部
グローバル商品開発センター Gプロジェクトチーム
重田大助

自己紹介

- 名前:重田大助(@shigepon7)
- 業務:ソフトウエア開発(特にLinuxカーネル)
  - スタビリティ
  - パフォーマンスチューニング
  - セキュリティ
- 社外ではこんなところにも…
  - Androidセキュリティ・バイブル2012/2013
  - JSSECデバイスWGリーダー

今日のお話の概要

- 電話帳アクセスモニター機能のご紹介
  - 電話帳アクセスモニターとは何か？
  - なぜ作ったのか？
  - どういう方法で作ったのか？

電話帳アクセスモニターとは何か？

電話帳アクセスモニターとは何か？

- SH Developers Squareでの紹介

本機能は、アプリケーションが電話帳データにアクセスしたタイミングで、お客様にアクセスの発生を通知する機能です。また、端末の設定を変更することにより、お客様が電話帳データにアクセスする事を望まれないアプリケーションに対して、電話帳データへのアクセスをブロックすることもできます。

  - 搭載機種
    - SH-02E, SH-04E, 200SH, 203SH, DM014SH

https://sh-dev.sharp.co.jp/android/modules/others/#camonitor12w

ともかく電話帳アクセスモニターを使ってみよう

- 実際の動作を画面キャプチャーで説明します
  - 利用するアプリケーションは以下のもの

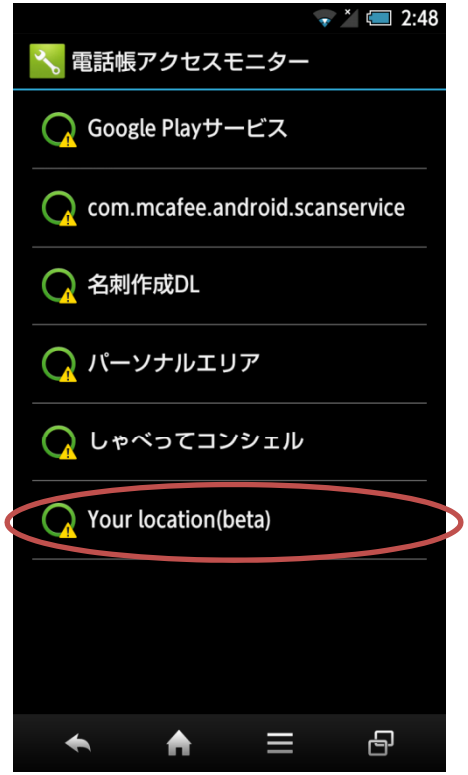
    - Your location(beta)
      SMSを使って相手端末の居場所を知ることができる簡易GPS位置確認アプリ(β版)です。

https://play.google.com/store/apps/details?id=jp.co.sharp.android.yourlocation

    - Your location(beta)は以下のWebサイトでソースコードも公開しています

https://code.google.com/p/your-location-beta/

電話帳アクセスモニターの動作画面



電話帳アクセスモニターとは何か？

- 要するに…

  「アプリケーションがいつ電話帳にアクセスしたかを可視化する機能」

- 例えばこういう使い方…
  - インストール時のパーミッションに電話帳へのアクセスが含まれているが私が利用したい機能は電話帳とは関係ないので実際に使ってみて電話帳へのアクセスを行っているかどうか確認する
  - インストール時のパーミッション確認がどうだったか覚えていないけど使っていたら教えてくれる

電話帳アクセスモニターとは何か？

- いつ動作するか？
  - 電話帳データの読み書きが発生した時点
- どんな動作をするのか？
  - 通知する
  - ブロックする
  - 通知してブロックする
- 動作に関する設定はどの単位で行うのか？
  - アプリケーション単位

電話帳アクセスモニターの購入時の動作

- ダウンロードアプリケーションへの動作
  - 電話帳データにアクセスした場合は通知する
  - 通知するだけでブロックは行わない

- プリインストールアプリケーションへの動作
  - 除外対象のアプリ以外はダウンロードアプリケーションと同じ
    - クラウドと同期処理を行うアプリケーションを電話帳アクセスモニターでブロックしてしまうとクラウドから電話帳データが消えてしまうのでそうしたアプリケーションは電話帳アクセスモニターの対象から除外

ブロックするとどうなるか？

- 電話帳を読み出した場合
  - 0件に見える
    =電話帳が空の時の動作と同じ
- 電話帳に書き込んだ場合
  - 書き込みができなかったように見える
    =ファイルシステムに空きがないなどのエラー発生時の動作と同じ

ソフトウエア的にどういう動作をするかは後程説明

なぜ作ったのか？

なぜ作ったのか？

- 不幸な事故が実際に発生してしまった
  - お客様の意図しないところで端末に保存された情報がネットワークに送信された問題
  - アプリのアンインストールがアプリ作成者の意図とは関係なく誘発された問題
- Androidのオープン性を妨げる事なくこうした事故を防ぐ取り組みの必要性を感じた

お客様の意図しないネットワークへの情報送信問題

- 概要
  - アプリケーションの内容とは関係なく、電話帳の読み取りとサーバへの送信を行っていたアプリが配布された問題

- なぜ広まったかに対する考察
  - インストール時のパーミッションの確認をしない人が多い
  - インストール時に確認してしまえばそれ以降は基本的にパーミッションを確認する手段がない

某アプリがアンインストールされた例

- 概要
  - アプリのアップデート時に機能追加を行うために電話帳へのアクセスの権限を追加したところアプリケーションのアンインストールが発生した
- なぜアンインストールされたかに対する考察
  - リテラシーのある人はインストール/アップグレード時のパーミッション権限が重要であると認識
  - マーケットアプリが「更新」「アンインストール」のいずれかを選択するようなUIであるため「アンインストール」が押しやすい位置にあった

Androidのパーミッションの確認について

- 不幸な2種類の事件
  - ユーザの意図とアプリケーションの動作が異なる
  - 必要な権限を利用しようとしただけでアプリケーションをアンインストール
- 事件が起きないようにできないか？
  - インストール時以外に権限を確認できればお客様が判断できるようになるのではないか？
    - 知らないアクセスに気付く
    - 使ってみてから確認できるようにする
  - 「電話帳」に限定して通知領域に通知するようにしてみた

なぜ端末メーカーが対応しないといけないのか？

- 標準の電話帳データに対するアクセスは端末の機能として提供されるから
  - 独自の電話帳データを用意するのであればアプリケーション側で通知や制限をするのは容易
  - どのアプリケーションからも参照できる標準の電話帳データに対するアクセスをどのように行うかはダウンロードアプリケーションで制御できない
  - あくまで手段を提供するのみで利用するかしないかはお客様がご自身で判断して利用できる形で提供する

どういう方法で作ったのか？

どういう方法で作ったのか？

- SH Developers Squareに書いています
  - 電話帳アクセスモニター機能は、マニフェストファイルに電話帳データベースへアクセスすることを宣言したアプリがContacts Providerへアクセスした際に、画面上部のステータスバーに以下の通知を表示します。（バックグラウンドで動作するサービスがアクセスした場合も表示します。）

- ContactsProviderに修正を行っています
  - ContentProviderとして実装されるアプリケーション実装の一部なのでAPIの互換性への影響が少ない

どのAPIを呼んだ時にチェックするのか？

- SH Developers Squareに書いています
  - **連絡先データの参照・削除・更新**
    （query/delete/update）
  - **連絡先データの登録**
    （insert）

SH Developers Squareの表記は禁止した場合の挙動だけのように読めますが、通知を出すかどうかも上記の4ヶ所で行います

どうやって通知しているのか？

- 単純にNotificationを利用
  - ContentProviderは他のアプリケーションのバックグラウンドで動作するものなのでそもそもフォアグラウンドに画面を出すような事はできない
  - アクセスを遅延させるような実装にするとアプリケーション側の動作に影響を与える懸念がある

どうやってブロックしているのか？

- データベースへのアクセスをする前にreturnするだけ
  - 各メソッドの戻り値はSH Developers Squareに記載

    - 連絡先データの参照・削除・更新
      （query/delete/update）
      - 戻り値が0として返却されます

    - 連絡先データの登録
      （insert）
      - 戻り値がnullとして返却されます。

アクセスしたアプリはどうやって識別するのか？

- Binder通信の相手を識別
  - ContentProviderはBinderを使ってプロセス間通信をするのでBinder.getCallingPid()を使えば相手のプロセスを識別する事ができる
  - プロセスが識別できれば後はPackageManagerからアプリの情報を取得する事が可能

設定画面に表示されるアプリはどれか？

- 電話帳へのアクセスをする権限を持つアプリ
  - 対象となる権限は以下の2つ
    - READ_CONTACTS
    - WRITE_CONTACTS
  - 設定対象外のアプリは設定画面に表示されない
    - 電話帳データのクラウドサービス連携を行うもの等

技術的な説明はこれだけです

- 技術的には難しいところはありません
  - AndroidのAPIの説明程度のレベル
  - 説明した内容を実装するのは非常に簡単
  - 説明していない範囲は以下のあたり
    - 設定UIをどう作るか
    - アプリ毎の設定をどう保存するか

- Google社に報告した上で搭載

まとめ

まとめ

- 電話帳アクセスモニター
  - アプリケーションがいつ電話帳にアクセスしたかを可視化し、選択すればブロックもできる機能
  - 技術的には非常にシンプルで同様の機能を他のデバイスに応用する事は容易
  - 不幸な事故をなくしアプリケーションをより広く使っていただくための端末メーカーとしての配慮

Androidの動向

- インストール時の権限確認だけではなく利用時の確認の動きがある
  - Android4.2よりアプリケーションから有料SMSの送信をする際に確認するダイアログを表示し、ユーザが許可してからでないと発信しない
    - ユーザの気付かないところで発生する海外における高額SMS発信アプリへの対策
- 電話帳アクセスモニターと基本的な考え方は同じであり、今後も重要な権限は手元で確認できるようになる機会は増えるだろうと予測