脆弱性があるアンドロイドアプリの作り方

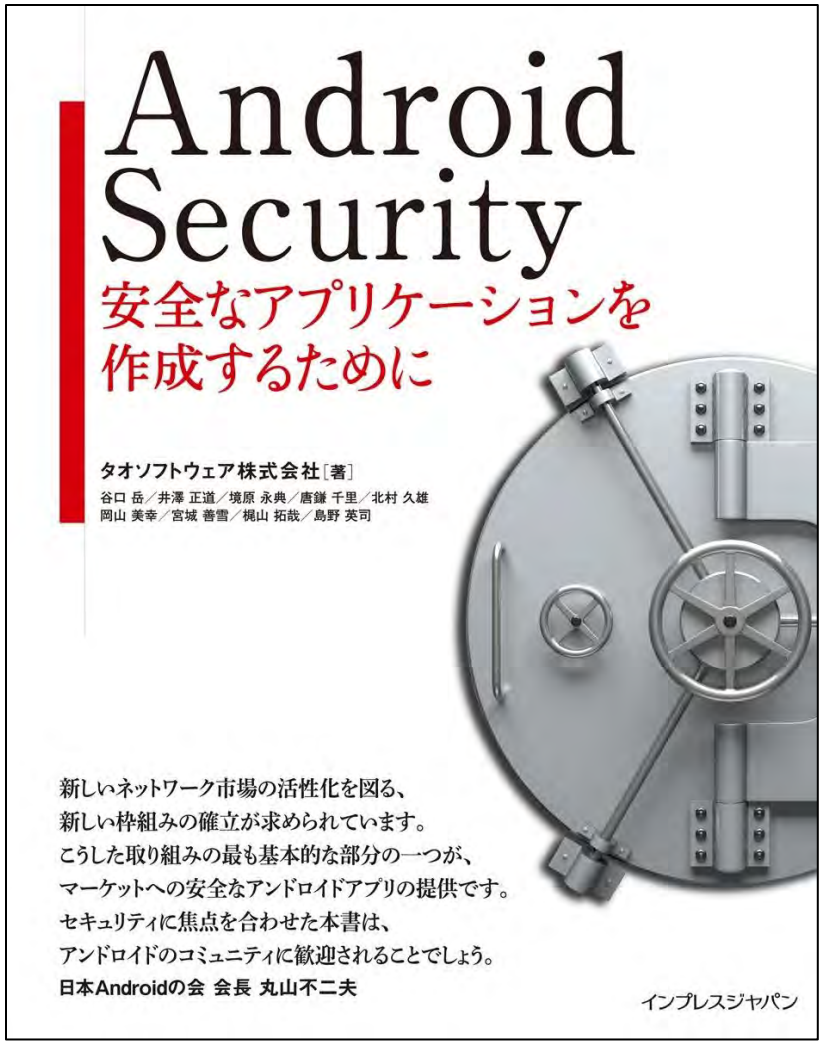
本資料は下記URLの最後の方からダウンロード可能です。
http://www.taosoftware.co.jp/android/android_security/

ABC2013 Spring
2013年3月15日(金) 13:00～13:40(40分)
E会場　28号館104教室
タオソフトウェア株式会社　谷口岳
Twitter:@tao_gaku

タオソフトウェア株式会社

- 日本の会社（Android専業）
- 独立系ソフトハウス
- Android発表と共に研究開発を開始
- 現在Android専業（受託開発）
- Androidマーケットにアプリを多数公開
- ブログにて開発者向け情報を発信
  - http://www.taosoftware.co.jp/blog/
- 雑誌他執筆、講演
- Twitter @tao_gaku

Android Security
安全なアプリケーションを作成するために



- ２０１２年１月１日発刊
- プログラマ向け
- アンドロイドのセキュリティに関してプログラマが注意すべき点が多くあるが、あまり認知されていなかったので本の執筆を行った。

- 資料
  - http://www.taosoftware.co.jp/android/android_security/
  - パワポ資料及びビデオ
- Think IT
  - 1章、2章、3章を掲載
  - http://thinkit.co.jp/book/2012/03/05/3463
- Amazon
  - http://www.amazon.co.jp/dp/4844331345/
- 電子版(DRMフリー)
  - http://www.impressjapan.jp/books/3134
  - Google Play Books,達人出版

JSSEC セキュアコーディングガイド

- 2012年06月11日初版
  - Android アプリのセキュア設計・セキュアコーディングガイド
  - PDF文書、およびサンプルコード
  - 文書
    - http://www.jssec.org/dl/android_securecoding.pdf
  - サンプルコード
    - http://www.jssec.org/dl/android_securecoding.zip
- JSSEC
  - 日本のキャリアやハードメーカ等が集まって作成されたセキュリティに関する団体

アンドロイド スマートフォン　プライバシーガイドライン
by タオソフトウェア

- 2012年10月9日初版
  - 2013/1/23 Version2.0
- アンドロイドアプリ提供者を対象にした、プライバシーポリシガイドライン
- 無料公開(ApacheLicense2)
  - http://www.taosoftware.co.jp/android/android_privacy_policy/

脆弱性があるアンドロイドアプリの作り方

セキュリティを考える視点

視点（立場）	セキュリティリスク	原因
個人ユーザ	個人情報漏洩 行動監視される 金銭的被害 端末紛失	マルウェアインストール フィッシングサイトアクセス 置き忘れ、盗難
企業ユーザ	トレードシークレット漏洩 顧客情報漏洩 業務システムへの不正アクセス （加えて各社員個人のリスクも）	マルウェアインストール フィッシングサイトアクセス 業務アプリに脆弱性 置き忘れ、盗難
アプリ開発者	データ漏洩 機能乗っ取られ 不正コピー・海賊版 著作権侵害	脆弱性（バグ） 対処不足 不要な通信 広告モジュール
サービスプロバイダ	不正アクセス 個人情報収集の疑いを受ける	脆弱性（バグ） 意識欠如（説明不足） メリット優先
ハードメーカ	ルート取得 OSに脆弱性 プリインアプリに脆弱性	脆弱性（バグ） 脆弱性（バグ） 意識欠如（市場優先）

Androidコンポーネント図

アプリの
セキュリティ

端末の
セキュリティ

Androidアプリ脆弱性の内訳

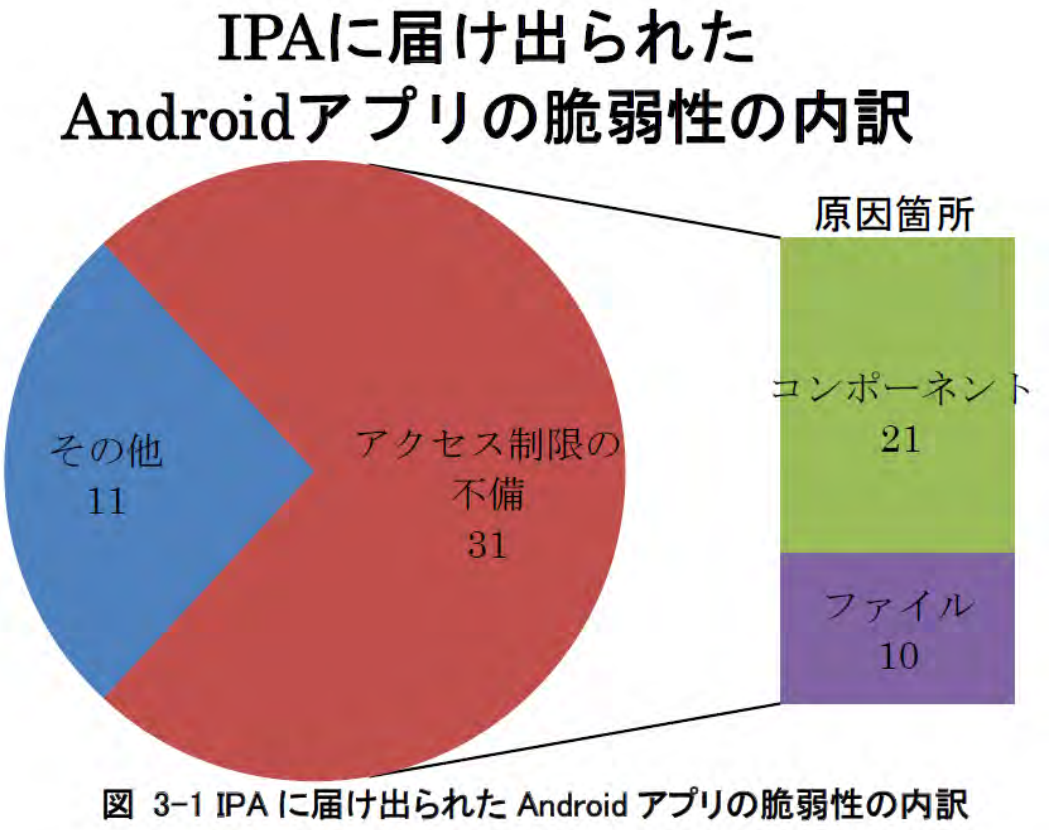


図 3-1 IPA に届け出られた Android アプリの脆弱性の内訳

- IPAに届け出られるAndroidアプリの脆弱性関連情報も増加傾向にある。届け出られた脆弱性は、アクセス制限の不備に関するものが7割以上であった。さらに分析した結果、これらはAndroidの仕組みを理解し、適切にアクセス制限の設定をしていれば防ぐことのできる脆弱性であることがわかった。(2012/6)

『Androidアプリの脆弱性』に関するレポート
http://www.ipa.go.jp/about/technicalwatch/20120613.html

ファイルのアクセス制御

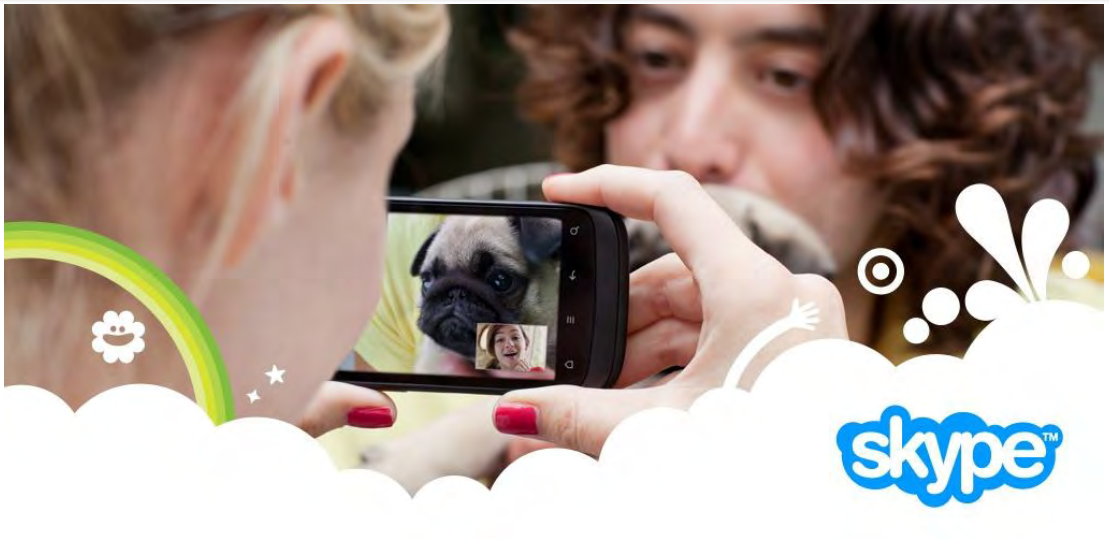
Androidのファイル

✓他のアプリからファイルが読める→重要なデータを読み取られる

✓他のアプリからファイルが書き込める→ハングアップ、アプリデータの改変

注意する事

- ファイルの作成方法
- ファイルの作成場所
  - アプリケーションデータディレクトリ
  - 外部記憶装置（ＳＤカード）



やっちゃった例

- 事件の概要
  - 2011年4月16日 Android版 Skypeに個人情報を取得できる脆弱性発見
  - 登録した個人情報、コンタクト、チャット履歴が閲覧可能な状態になっていた。
  - 4月21日に修正完了
- 原因
  - 設定ファイル、データベースのアクセス権限設定ミス
  - Windowsからの移植っぽい
    - Windowsだと同じユーザがインストールした他のアプリから参照可能
    - Androidだとだめ
  - 凡ミス

ファイルの作成方法

ファイルの作成は、MODE_PRIVATEを使う（デフォルトの動作）

- ファイルの作成方法
  - プリファレンスファイル
  - データベースファイル
  - 自分で作成するファイル（いつものJava）
  - ＮＤＫ
- ファイルパーミッションに注意
  - Linuxのファイルパーミッションと同じ
  - 自分自身のみ読み書き可能、他のアプリからの読み込み、書き込み可能等を指定できる

モード	説明
MODE_PRIVATE	作成したアプリのみ読み書き可能
MODE_WORLD_READABLE	他のアプリから読み込み可能（OS4.2で廃止）
MODE_WORLD_WRITEABLE	他のアプリから書き込み可能（OS4.2で廃止）

NECアプリの脆弱性事例 (SDカードに重要なデータを保存)

NEC Empowered by Innovation

ホーム 検索

ニュース 製品 ソリューション・サービス サポート・ダウンロード お問い合わせ NECについて

ホーム > 製品 > NEC製品セキュリティ情報 > セキュリティ情報 > NV12-008

NEC製品 セキュリティ情報

お知らせ

セキュリティ情報

影響のある製品

カテゴリ順

アルファベット順

日付順

掲載番号:NV12-008
脆弱性情報識別番号:JVN#05102851

Android版 嫁コレにおける端末識別番号の管理不備の脆弱性

概要

Android版 嫁コレには、IMEI(端末識別番号)をSDカードに保存する問題が存在します。
不正な他のAndroidアプリケーションを使用した場合、IMEIを取得される可能性があります。

対象製品

- IMEIをユーザ識別に使っていた
- テスト時にユーザ切り替えしやすいようにSDカードに書いていた
- http://www.nec.co.jp/security-info/secinfo/nv12-008.html

アプリケーションデータディレクトリ

他のアプリからのアクセスは通常できない

- アプリケーションデータディレクトリ
  - /data/data/パッケージ名ディレクトリ
  - ディレクトリの所有者読み込み、書き込み、実行
  - グループ読込み、実行
  - その他実行

外部記憶装置（SD）

外部記憶装置のファイルは全てのアプリケーションがアクセス可能

- パーミッション
  - Androidは書き込み用のパーミッションだけ定義されている（WRITE_EXTERNAL_STORAGE）
  - 読むだけならパーミッション不要（将来変更される）
- ファイルパーミッション
  - アプリケーションから外部記憶装置へファイルを書きだした場合、ファイルのオーナはsystem、グループはsdcard_rwが設定される。
  - アプリケーションディレクトリにファイルを書きだした場合と異なり、どのアプリケーションがファイル出力を行ってもオーナとグループは同じ値となる
  - アプリケーション毎にファイルやディレクトリのアクセス制御を行うことはできない
  - 外部記憶装置のファイルやディレクトリは全てのアプリケーションがアクセス可能であることを意味する

コンポーネントのアクセス制御

コンポーネント

- コンポーネントが複数集まってアプリケーションとなる

- ４つのコンポーネント
  - Activity　（画面表示やユーザ入力を受け持つ）
  - Service　（見えない場所動いて何かをする。例：データ収集）
  - Receiver　（見えない場所で、外部からメッセージを受け取る。）
  - Content Provider (外部からデータ(DB等)をアクセス可能にする）

- コンポーネントは外部アプリから呼び出し可能なので適切なアクセス制限をかけないと脆弱性を生む。
- 基本デフォルトで外部からアクセス不可となっている。

やっちゃった例

- 事件の概要
  - 他のアプリケーションからデータベースに変更を加え、Dropboxの公開用フォルダである「Public」フォルダにDropboxのアカウント情報が含まれている設定ファイルをアップロードさせたりすることが可能。
- 原因
  - ContentProviderは外部にデータを公開する仕組みなので、みんなに「公開する」がデフォルト値
  - android:exported=“false”を指定する必要があった。
- 詳しくは
  - ContentProviderのアクセス範囲 – Dropboxにおける脆弱性の修正
    - http://codezine.jp/article/detail/6286

AndroidManifest.xml セキュリティ設定

- コンポーネントのアクセス制限方法

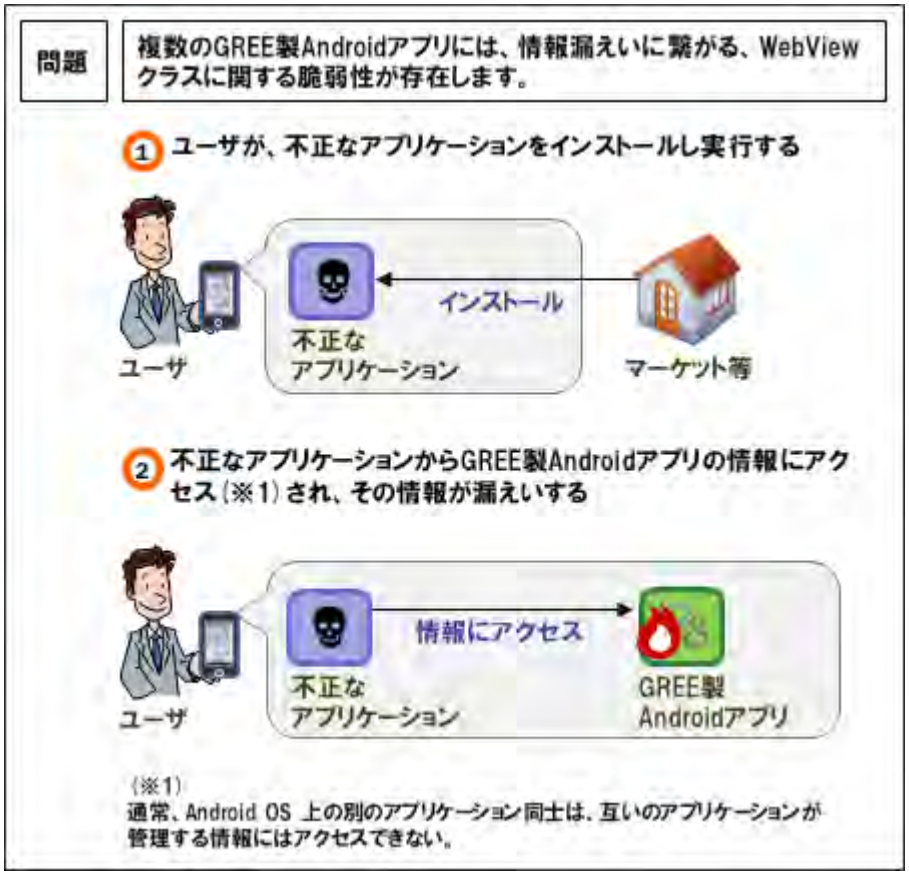
- android:exported
  - falseを設定した場合、他のアプリケーションから使用不可能になる
  - 自分自身かsharedUserId指定によって、同じユーザIDを持っているアプリケーションのみアクセス可能となる
- IntentFilterに注意
  - IntentFilterは外部公開する仕組み
  - IntentFilterが設定されている場合で、指定しない場合はtrue
  - IntentFilterが設定されていない場合で、指定しない場合はfalse

コンテントプロパイダのEXPORTEDに注意

- android:exported
  - 指定しない場合はtrue (4.2以降はfalse)
  - falseを設定した場合、他のアプリケーションから使用不可能になる
  - **注意：OSのバージョンによるExport値の動作の違い**

OSバージョン	Exportedにfalseを設定した時に外部からのアクセスが可能かどうか
1.6(API Level 4)	可能
2.1(API Level 7)	可能
2.2(API Level 8)	可能
2.3.1(API Level 9)	不可
2.3.3(API Level 10)	不可
3.0(API Level 11)	不可
3.1(API Level 12)	不可
4.0(API Level 14)	不可

GREEアプリの脆弱性（機能の乗っ取り）



ユーザが、不正な他の Android アプリケーションを使用した場合、当該製品のデータ領域にある情報が漏えいする可能性があります。

- JVN 複数の GREE 製 Android アプリにおける WebView クラスに関する脆弱性
  - http://jvndb.jvn.jp/ja/contents/2012/JVNDB-2012-000077.html
- CodeZine スマートフォンアプリへのブラウザ機能の実装に潜む危険―WebViewクラスの問題について　(http://codezine.jp/article/detail/6618)

出典：JVN 脆弱性対策情報データベースより

アプリケーション機能の悪用

- 脆弱性をつかれ、第三者アプリが悪さをしてユーザに迷惑がかかる
- 例
  - 電話をかけるアプリ
    - 電話をかけるロジックが外部から使用可能（チェック機能なし）
    - 他のアプリから権限なしで電話をかけほうだい（ダイアルQ2?)
  - ファイルをアップロードするアプリ
    - 他のアプリから無断でサーバにファイルがアップロードされる。
    - →他のアプリから無断でサーバ上のデータが書き換え可能という意味

ユーザデータ以外の守る物

ユーザデータ以外に守る物

1. アプリケーション内の著作物データ
   - 画像、動画、音声、文字列

2. アプリケーションロジック
   - 特殊なアルゴリズム、暗号化キー

1. アプリ内の著作権データを守る

- アプリケーション内のデータは総て簡単に抜き取り可能

1. PCと接続してAPKファイルの吸出し
2. Android端末上でAndroidアプリによる
   リソースの吸出し

タイトル：ブラックジャックによろしく
著作者名：佐藤秀峰
サイト名： Manga on Web
URL: http://mangaonweb.com

1-1 PCと接続してAPKファイル吸出し

- APKファイルはZIPファイル
  - 拡張子を変えて解凍すると構成ファイルを取り出すことができる。
- APKファイルの構造

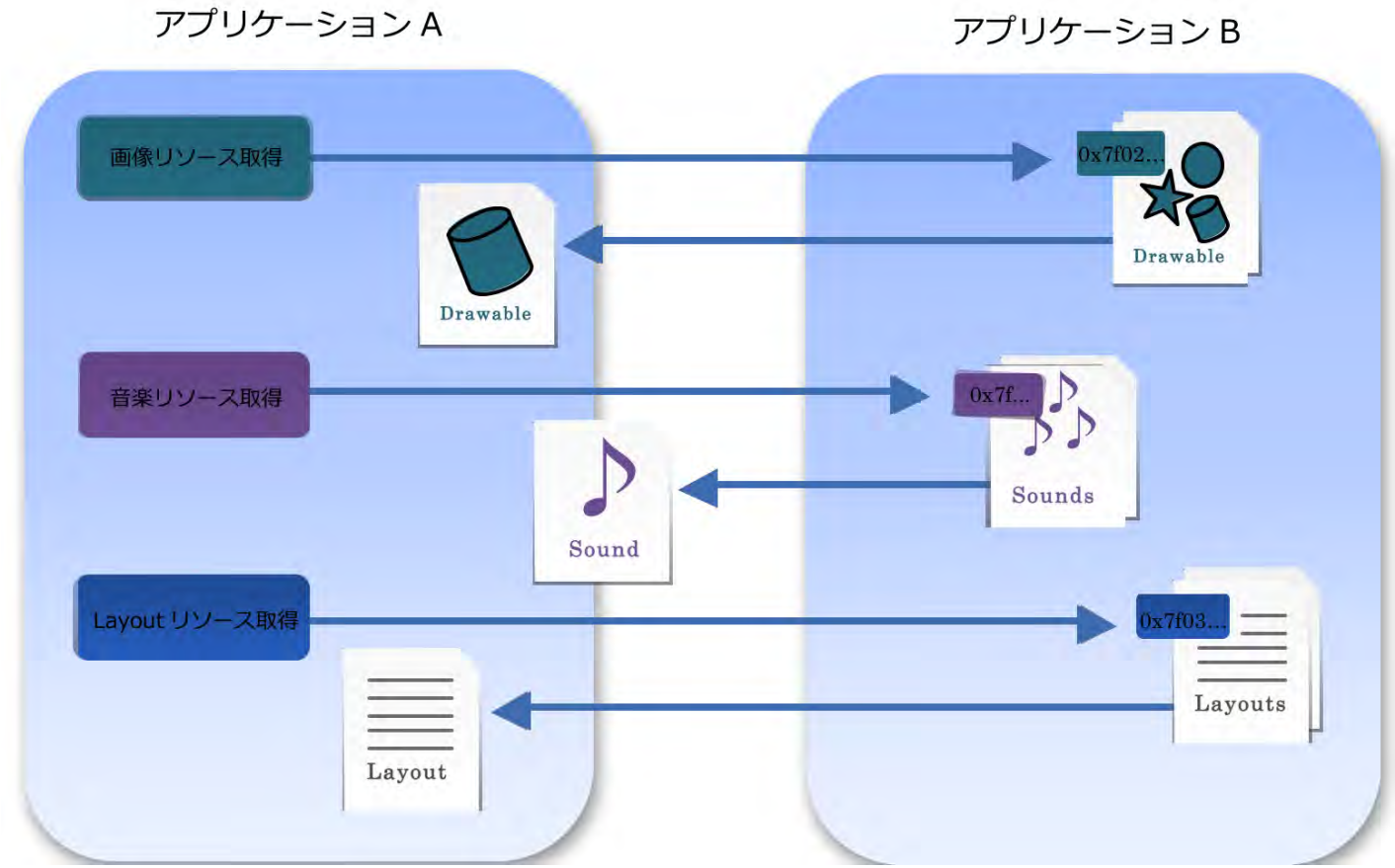
```
/trssreader.apk
┗AndroidManifest.xml　ー　Androidのマニフェストファイル
┗resources.arsc　ー　resディレクトリの内valuesディレクトリをまとめたもの
┗classes.dex　ー　プログラムコードをまとめたもの
┗META-INF　ー　署名関連のファイルが含まれている
┗/assets　ー　開発時のassetsディレクトリがそのまま含まれている
  ┗test.png
  ┗index.html
┗/res　ー　開発時のvaluesディレクトリ以外のresディレクトリがそのまま含まれている
  ┗/drawable
    ┗icon.png
  ┗/layout
    ┗main.xml
```

デモ

- 画像ファイルの取り出しを行います。

1-2 Android上でアプリによるリソースの吸出し

- PCと接続しなくてもアプリのリソース情報を取得可能
  - アプリケーション内に同梱されているデータは、全て取得可能



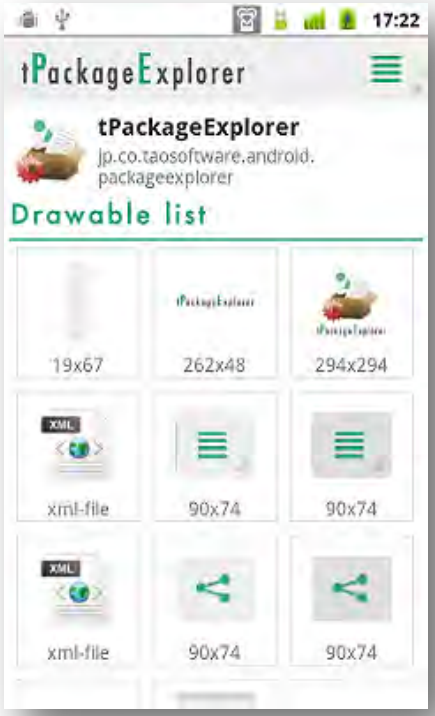
tPackageExplorer

- Androidアプリケーションのリソース情報を確認するアプリ
  - https://market.android.com/details?id=jp.co.taosoftware.android.packageexplorer
  - Google Playで「taosoftware」で検索

アプリー覧

AndroidManifest.xml

アプリ内画像の表示



デモ

2. アプリケーションロジック

- Javaで書かれているAndroidアプリケーションはソースコード解析が簡単
  - ツールも出回っており、解析を防ぐ事はできない
  - 簡単な手順で解析が可能
    1. APKファイル取得
    2. Dex2jar classes.dex
    3. JavaDecompiler
  - root化などの特殊な処理は必要ない
  - Proguard等の難読化ツールで時間稼ぎ
  - NDKにコードを移動（完全ではない）

デモ

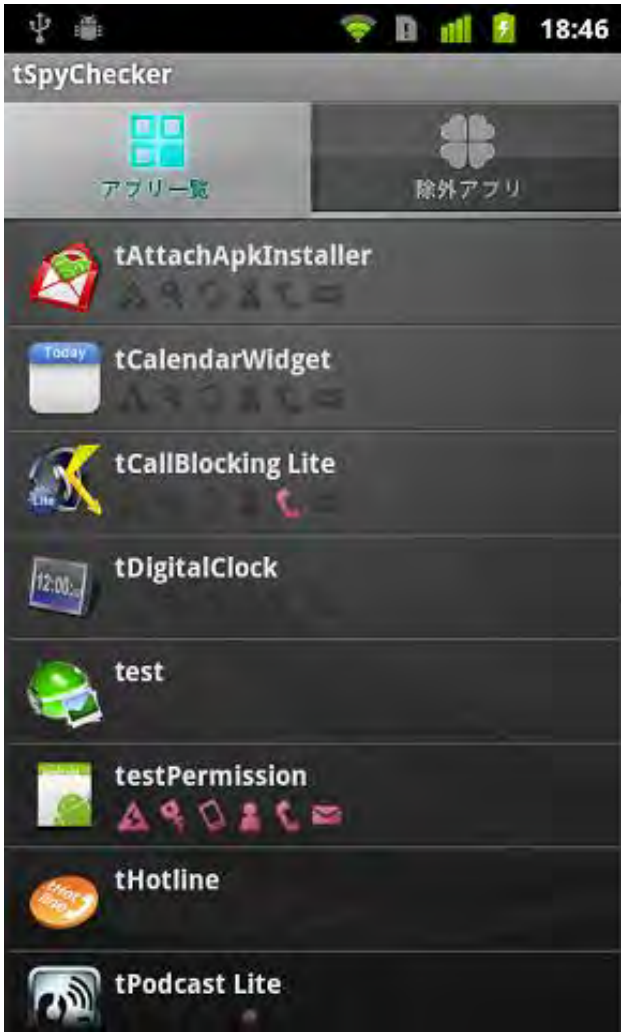
- Dex2jar
  - https://code.google.com/p/dex2jar/
- JavaDecompiler
  - http://java.decompiler.free.fr/

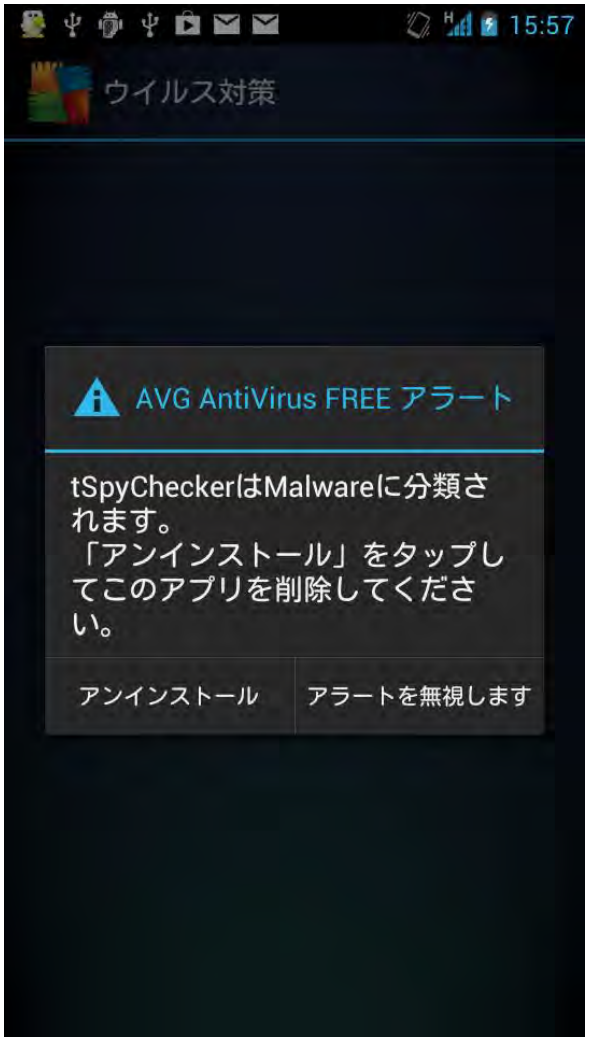
まとめ

- 前提
  - アプリケーション内のデータ、ロジックは総て簡単に抜き取り可能
- 対策
  - アプリケーション内には、コピーされると困るファイルは置かない。
  - パスワード等をリソース、コード内に格納しない。
  - 抜き取られても良い品質のファイルにする。（低画質、低音質）
  - 暗号化した画像ファイルを入れる（あまりよくないけど）
  - データはサーバーから持ってくる。
  - DRM使う？
  - 有償ツール使う？
- 結論
  - どれぐらいのレベルでデータを守るかによって実装方法が異なる。
  - 厳密にすればするほどお金がかかる。

マルウェアに間違えられる？

間違えられました！？





誤検知は怖い

- マルウェア (Malware) とは、不正かつ有害な動作を行う意図で作成された悪意のあるソフトウェアや悪質なコードの総称である。(by Wikiperia)

- マルウェア (Malware) とは、ウイルスチェックアプリがマルウェアと言ったものになっている。＼(゜ロ＼)(／ロ゜)／

- トレンドマイクロの誤検知、フリーソフトの開発を停止に追い込む
  - http://www.excite.co.jp/News/net_clm/20120711/Slashdot_12_07_11_0257259.html

使用している広告モジュールがマルウェア！？

マルウェアと認定される広告モジュールが入っていたら、そのアプリはマルウェアです。

- 広告モジュールが電話帳を参照して勝手にサーバに送っていないか？
- 電話帳を送るのは流石に少ないが、以下の物は良く送られている。
  - 電話番号
  - IMEI
  - ANDROID_ID

携帯電話は個人情報の宝庫

- 電話帳をサーバに送る人
  - 名簿屋さん
  - ソーシャル電話アプリ

区別しずらい。

- 広告屋さん
  - 個人に特化した広告を出すことができる
  - 渋谷にいるなら、渋谷近辺の広告を出す
  - 女性には女性向けの広告を出す
  - 年齢別の広告を出す
  - 子供がいる家庭には子供向けの広告を出す

現在問題になってます。
現在進行中…

- 怪しそうなグレーのアプリが多いため、マルウェア判断も大変

インストール時の「Permission確認画面」

- 現状
  - 何に「同意」をするのかわからない。
  - 一般ユーザはこの画面の内容を読まない・理解できる内容ではない。
  - アンドロイドに詳しい人でも、表示される内容で何がおこるのかわからない
  - ユーザが安心して使用できない。
- 対策
  - 法的に問題なければ良いは間違い、不信感を抱かれないようにする必要がある。
  - ユーザの情報の取り扱いについて、Google Play上や説明サイトで詳しく説明をする。(プライバシーポリシーの作成)
  - 重要な情報については、アプリケーション内で同意取得ダイアログ等を出すようにする。
  - 使わないパーミッションは使用しない

Permission画面

参考:不正アプリ供用事件の不起訴は何の立証が困難だったか
http://takagi-hiromitsu.jp/diary/20130129.html

利用者情報取り扱い

- 2012年8月に、総務省から「スマートフォン　プライバシー　イニシアティブ」が発表され、スマートフォンにおける、利用者情報の適切な取り扱い指針が示された。
  - http://www.soumu.go.jp/menu_news/s-news/01kiban08_02000087.html

スマートフォンの利用者情報等に関する連絡協議会（ＳＰＳＣ）が設立された。

何をやれば良いか

- アプリケーションごとにプライバシーポリシーを策定すると共に、一定の情報の取得については、個別の情報の取得について、同意取得を求める。

- プライバシーポリシードキュメントの作成
- 重要な情報を取得する時にダイアログでユーザに告知

アンドロイドスマートフォンプライバシーガイドライン作りました。無料公開(ApacheLicense2)
http://www.taosoftware.co.jp/android/android_privacy_policy/

まとめ

まとめ

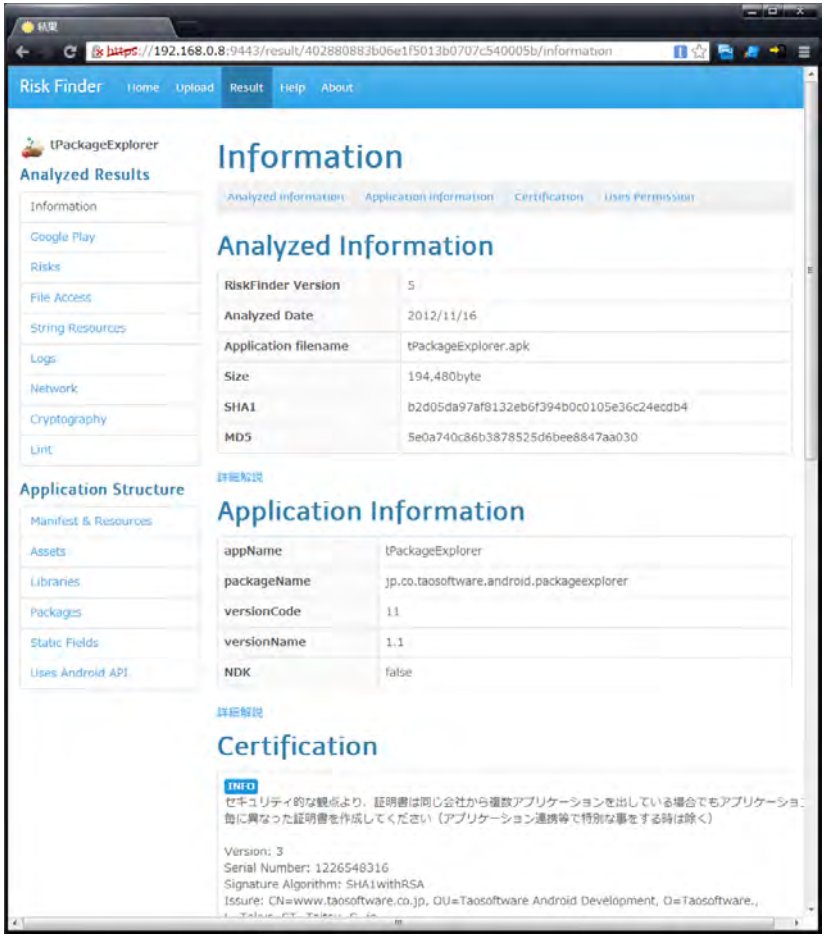
- ファイルのアクセス制御
- コンポーネントのアクセス制御
- アプリケーション内の著作権データ
- アプリケーションのロジック
- マルウェアに間違われないために
  - ライブラリに注意
  - プライバシーポリシーの作成

気を付ける所はまだまだありますが、細かい点までお話しすると2日ぐらいかかります。

Tao RiskFinder （脆弱性発見ツール）

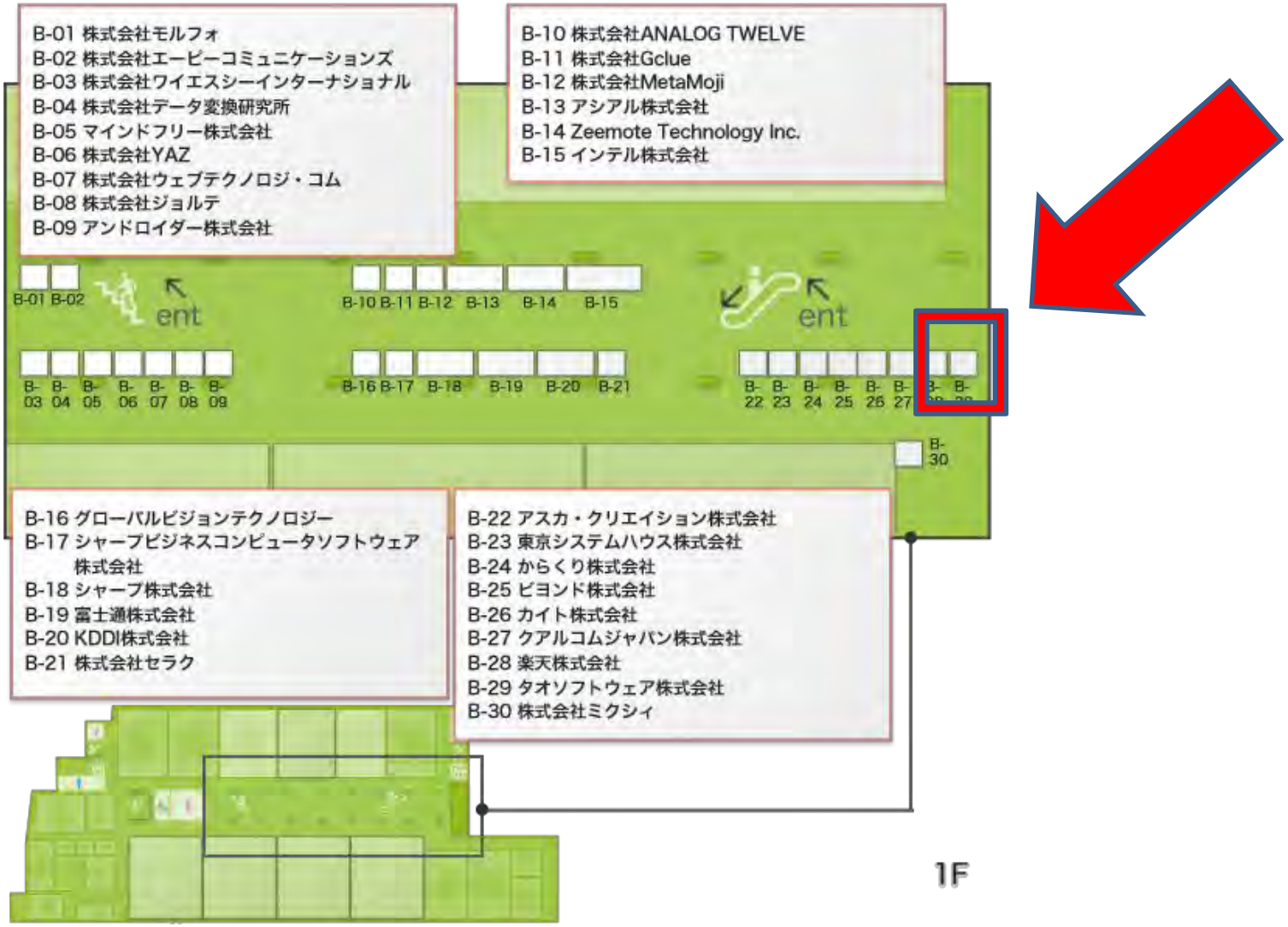
APKファイルをアップロードするだけで脆弱性レポートが作成されます。

講演をする中で、
「気を付ける事が沢山あるのは分かった。
でも全てのプログラマが理解するのは
難しい
何かいい方法はないか？」
という声があったので作ってみました。



1. プログラマでなくても使える
2. ソースコード不要
3. ウエブサービス型
4. 脆弱性以外も検出

バザーやってます。



ありがとうございました。

ABC2013 Spring
2013年3月15日(金) 13:00～13:40(40分)
E会場　28号館104教室
タオソフトウェア株式会社　谷口岳
Twitter:@tao_gaku