UBITEQ
UBIQUITOUS TECHNOLOGY

Androidアプリケーションの脆弱性発生ポイントとその傾向

株式会社ユビテック ソフト評価・検証室
発表者氏名：田久保　順
共著者氏名：八丁　勝利

目次

UBITEQ
UBIQUITOUS TECHNOLOGY

はじめに

1. Androidアプリの検証方法の妥当性と網羅性の検討

2. セキュリティ検証結果と脆弱性事例

3. Androidと他OSのセキュリティ比較

参考文献

はじめに

・株式会社ユビテックについて

・ソフト評価・検証室について

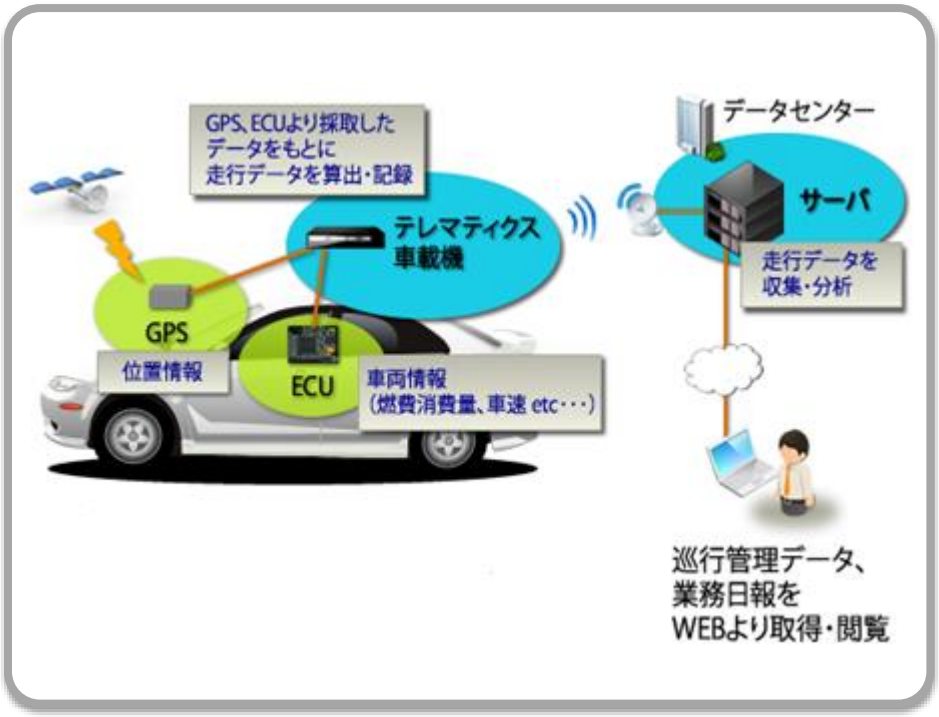
株式会社ユビテックについて

UBITEQ
UBIQUITOUS TECHNOLOGY

このような製品を自社開発し、サービス展開しています。

http://service.ubiteq.co.jp/utq_service/index.html

省エネソリューションサービス



カーソリューションサービス

ソフト評価・検証室について

UBITEQ
UBIQUITOUS TECHNOLOGY

● 自社開発システムの評価検証

リリース製品の品質確保が至上命題（いかに事前に不具合を無くすか）

テーマ
- ユビテックの「端末(センサー)＋ネットワーク型」システムの品質評価
- 不具合情報の統計分析とテスト項目へのフィードバック手法

● 評価サービスの展開

スマートフォン、データ通信デバイス、ホームセキュリティ機器など、モバイル端末を中心に検証サービスを展開

テーマ
- ユーザの行動分析による評価品質確保とテストの効率性の両立
- 新製品、新機能に対する試験観点、試験要件のスピード策定
- セキュリティをテーマとした検証手法の開拓、研究

1. Androidアプリの検証方法の妥当性と網羅性の検討

- 概要
- 研究対象範囲
- 攻撃経路（インタフェース）の整理
- セキュリティ検証メニューの整理

概要

UBITEQ
UBIQUITOUS TECHNOLOGY

Androidアプリはデバイスの重要機能にアクセスできたり、端末が持つ個人情報など利用でき、他のアプリとも連携して動作することから、自分のアプリの機能が他のアプリから悪用されることがないかといったことを十分にチェックしないと、重大なセキュリティインシデントに発展するリスクがある。

開発者が注意すべきセキュリティのチェックポイントについて、検証メニューとして整理。実際にPlayストアで配布されているアプリで検証を実施し、市場に出回っているアプリのセキュリティ問題の実態について調査。

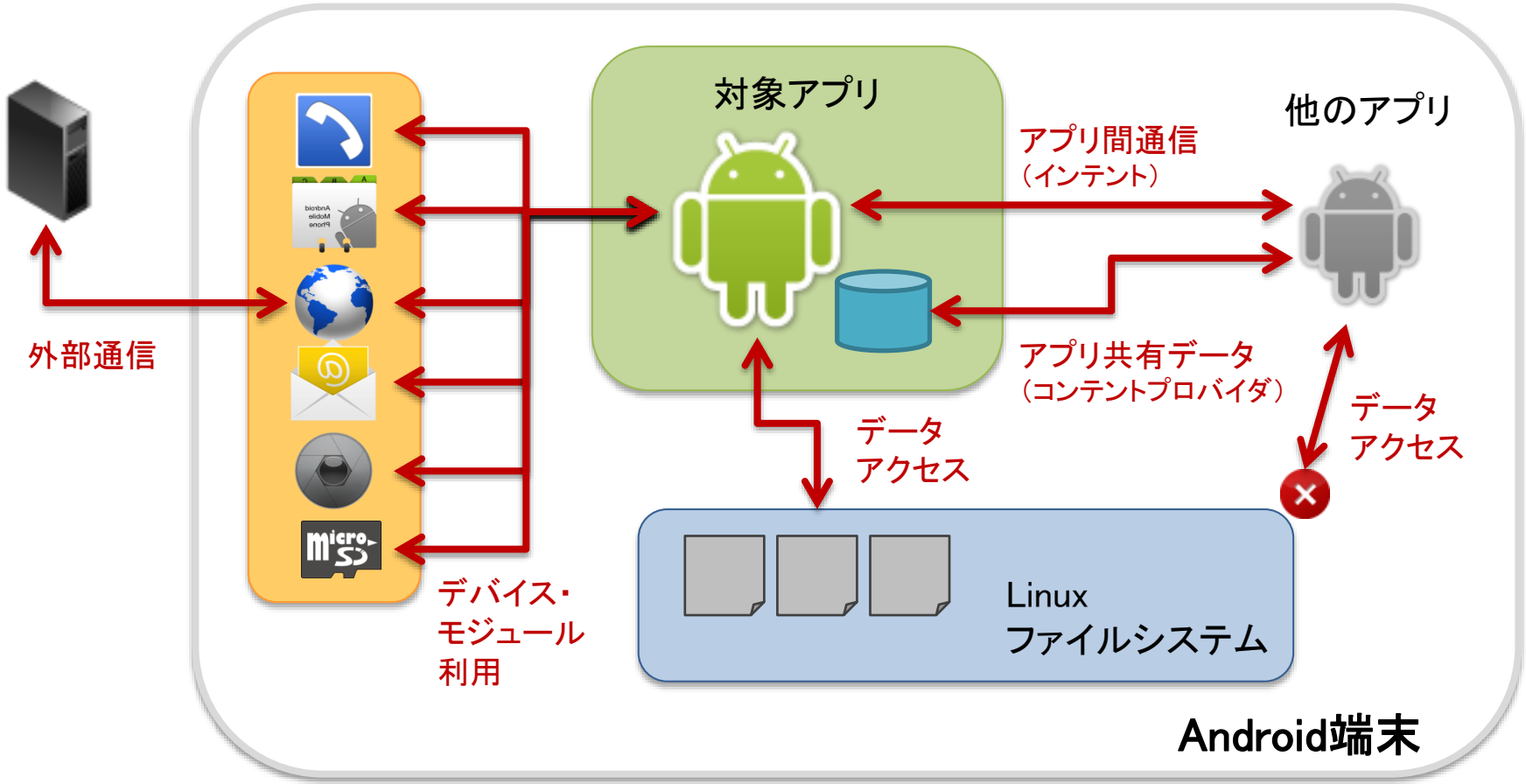
研究対象範囲

UBITEQ
UBIQUITOUS TECHNOLOGY

攻撃経路（インタフェース）の整理

UBITEQ
UBIQUITOUS TECHNOLOGY

・検証の網羅性を考える上で、アプリのインターフェイスを整理
・外部とつながる部分がセキュリティ攻撃を受ける危険性あり（赤線の箇所）



セキュリティ検証メニューの整理

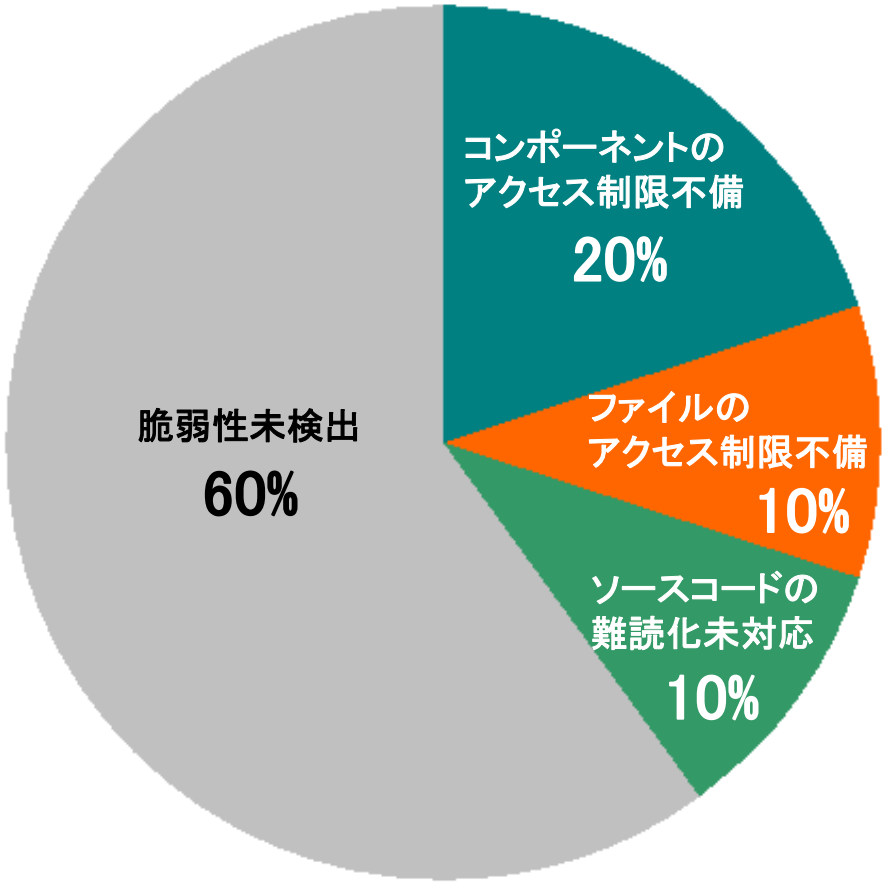
UBITEQ
UBIQUITOUS TECHNOLOGY

検証メニュー	検証したいセキュリティリスク
Android Permission、 独自定義Permission	攻撃アプリから踏み台利用されることにより、デバイスの重要機能を不正に利用される。メールやインターネット経由でユーザの機密情報が流出する危険性　など。
データ格納ディレクトリの妥当性	重要データを盗み見・改ざんされることにより、情報流出やイタズラなどのトラブルにユーザが巻き込まれる危険性　など。
ファイル・ディレクトリのアクセス権	
インテント・エクストラパラメータ	アプリ間の処理から機密情報が盗み見まされる危険性　など。
ソースコードの難読化対策	重要な機密技術の流出、ソースコード窃取によるコピーソフト氾濫の危険性　など。

2. セキュリティ検証結果と脆弱性事例

- 結果の概要
- 事例1： パーミッション設定の不備
- 事例2： SDカード利用の盲点
- 事例3： リバースエンジニアリング対策の不備
- 調査まとめ

結果の概要



Playストアより無料の10アプリ（ブラウザ・SNS・音声通話アプリなど）を選択し、セキュリティ検証を実施した結果、4アプリ（40%）から何らかの脆弱性を検出。

パーミッション（許可）設定不備のリスク

UBITEQ
UBIQUITOUS TECHNOLOGY

Androidアプリがデバイスの重要な機能や情報にアクセスする場合、インストール時に使用許可をユーザに求める。ユーザの許可がない限りインストールできない。

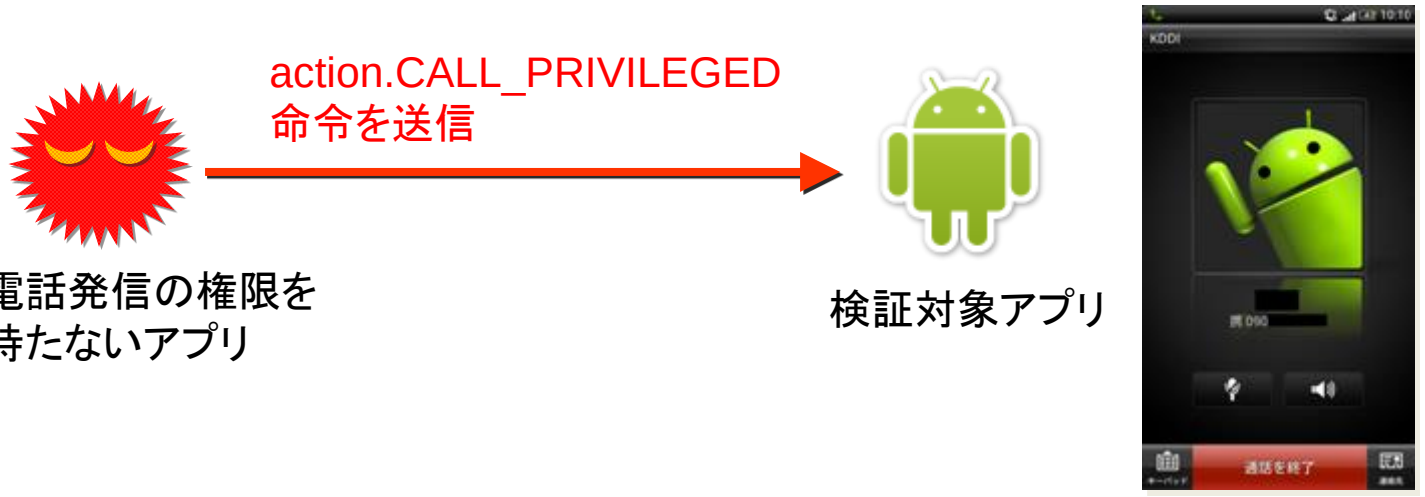
ユーザの許可を得ていないアプリでも、Androidのアプリ連携機能を悪用することで、許可を得ているアプリを経由してデバイスの重要な機能や情報にアクセス可能な場合がある（＝踏み台攻撃）。

踏み台攻撃を受けないように、アクセス制限を適切に設定しておかなくてはならない。

事例1： パーミッション設定の不備

UBITEQ
UBIQUITOUS TECHNOLOGY

某インターネット通話アプリは、電話発信権限を持たないアプリからでも電話発信要求を受け付けている。踏み台に利用されると、ユーザは金銭被害を被ったり、勝手にイタズラ電話をかけられてトラブルに巻き込まれるリスクがある。



命令に従って電話発信

パーミッション設定の対策

UBITEQ
UBIQUITOUS TECHNOLOGY

1.電話発信要求の受付を、自アプリもしくは同一ベンダーのアプリに制限する

```
- <activity android:name="A" android:launchMode="singleTask" android:configChanges="ke
    android:windowSoftInputMode="adjustResize" android:exported="false">
  - <intent-filter android:icon="" android:priority="0">
      <action android:name="android.intent.action.CALL_PRIVILEGED" />
```

アクションの実行はアプリ自身か同じSharedUserIDのアプリケーションに限定するようにマニフェストファイルを設定する。

2.他アプリが電話発信の許可を持っていることを確認する

```
- <activity android:name="A" android:launchMode="singleTask" android:configChanges="ke
    android:windowSoftInputMode="adjustResize" android:permission="CALL_PRIVILEGED">
  - <intent-filter android:icon="" android:priority="0">
      <action android:name="android.intent.action.CALL_PRIVILEGED" />
```

ユーザから CALL_PRIVILEGED の使用が許可されているアプリからのみ電話発信要求を受け付けるように設定する。

事例2： SDカード利用の盲点

UBITEQ
UBIQUITOUS TECHNOLOGY

某インターネット通話アプリは、ユーザが登録した電話番号の情報をそのままSDカードに保存している。電話番号が流出したり、電話番号が書き換えられて身に覚えのない番号にかけてしまうリスクを有する。

機密データ保存の対策

UBITEQ
UBIQUITOUS TECHNOLOGY

1.SDカードに重要なデータを保存しない

SDカード内のファイルやディレクトリはユーザ（アプリ）ごとのアクセス制限はできないため、誰でも閲覧や改ざんが可能。
重要なデータは、端末内のアプリ固有のデータ領域に、適切なアクセス権を設定して保存することが望ましい。

2.SDカードに重要なデータを保存する場合は、暗号化する

この場合、暗号化・復号鍵の管理はSDカードではなく、端末内のアプリ固有のデータ領域か、セキュアなネットワーク上で行うこと。

事例3： リバースエンジニアリング対策の不備

UBITEQ
UBIQUITOUS TECHNOLOGY

Java実行バイナリは逆コンパイルで元ソースコードを取り出すことが可能。これを防ぐことはできない。

ソースコードがそのまま読めてしまうと、暗号などの重要な技術が漏洩したり、コピーアプリがバラまかれるなどのリスクがある。

リバースエンジニアリング対策

UBITEQ
UBIQUITOUS TECHNOLOGY

1.読みづらくする

ソースコードの難読化対策（クラス名や変数名を意味のない文字に置換し、可読性を低くする）により、リバースエンジニアリングの難易度を上げる。

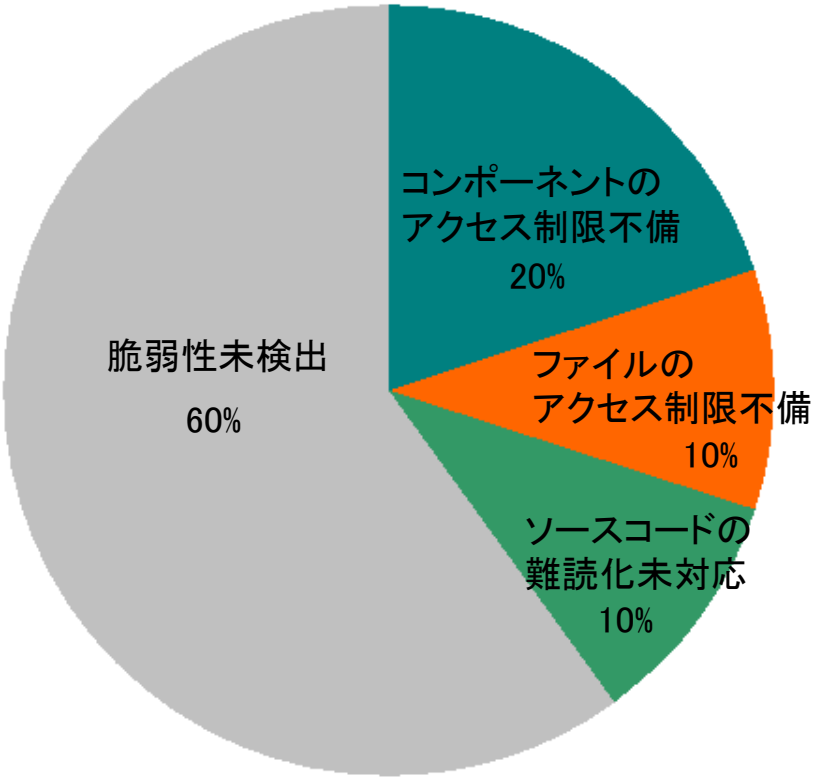
2.読めなくする

秘密にすべき処理をネイティブコードで実装する（ただし、デメリットとしてはハードウェアごとにアプリ開発が必要）。

3.改ざんチェックを行う

改ざんされたソースコードが用いられていないかを確認するため、アプリ起動時に常にサーバと連動してアプリ認証を行う。

調査まとめ



検証で見つかった問題の大半は、デバイス利用の設定漏れや、SDカードへの不用意なデータ保存など、基本的なものが多い。開発者はJava言語の知識はあってもAndroidのセキュリティシステムについて十分な知識がなく、脆弱性を作り込んでいるものと考察される。

アプリ開発において、セキュリティ上気をつけるべきポイントと、具体的な事例情報を整理することで、基本的な問題（しかし、市場において多くの問題）は防げるものと思われる。

3. Androidと他OSのセキュリティ比較

- iPhoneにおける脆弱性事例
- 第3OS（FirefoxOS）の脆弱性リスク
- Androidアプリの脆弱性の背景
- 今後のスマートフォンのセキュリティ考察
- 発表の終わりに

iPhoneのセキュリティ事例

UBITEQ
UBIQUITOUS TECHNOLOGY

	報告	概要
ワームソフト 「Ikee」	2009年11月	・Jailbreakした端末に限定し感染 ・影響は背景の壁紙を変更する極めて害の低いもの
ハッキングツール 「iPhone/Privacy.A」	2009年11月	・Jailbreakした端末に限定し、感染。 ・ユーザデータの殆どをコピー可能な状態に書き換える ※手口は「Ikee」と同じくデフォルトのSSHを使用し、ハッキングを仕掛けるもの。
ユーザ操作による ロック画面の解除	2013年9月	・対象は「ios6.1.3」「ios7β 版」「ios7」のiPhone、iPad ・マルチタスキング画面を経由し、ロック画面を迂回してアプリを起動するもの。 ※脆弱性ではあるが、本質的なセキュリティシステムに穴が見つかった訳ではない。

2件はJailbreak端末が対象。深刻な脆弱性のニュースは殆どない。

第3OS（FirefoxOS）の脆弱性リスク

UBITEQ
UBIQUITOUS TECHNOLOGY

	Android	FirefoxOS
危険なアプリが出回るリスク	高	低
	誰でも端末の重要機能や情報にアクセスできるアプリが作れる。Playストアではマルウェア検知プログラムによる検査があるが、人手によるレビューはない。	端末の重要機能にアクセスするアプリは、マーケットの人手によるレビューを受ける必要がある。また、マーケット以外のアプリはできることが非常に限定される。
踏み台攻撃のリスク	高	低
	パーミッション設定不備により、端末機能乗っ取りや機密情報漏洩の危険性がある。また、WebViewを利用したXSS攻撃などの危険性がある。	重要な機能の使用はユーザの同意を必要とする。情報を抜かれる可能性はあるが、漏洩させるための経路（インターネットなど）の使用についてはマーケット審査で厳しくチェックされている。XSS対策はOSレベルで実装。
リバースエンジニアリング悪用のリスク	高	調査中
	難読化対策ツールなどを使わなければ悪用されるリスクが高い。	

Androidアプリの脆弱性の背景

UBITEQ
UBIQUITOUS TECHNOLOGY

1. 複雑なセキュリティモデル

   Javaが理解できればアプリが作れるため、Android特有のセキュリティシステムを理解せずにアプリを開発

   ※アプリ開発のコストが下がり、セキュリティ対策に手が回らないという背景も？

2. リベラル・フリーダムな設計思想

   アプリ開発者にとっての利便性を追求した基本設計。Intentによるアプリやデバイス制御の自由度が高い（ユーザー確認を必要とせずに電話発信など重要な機能を利用可能、など）

3. アプリの審査方針

   iOSやFirefoxOSのような「事前審査制＋人手によるレビュー」ではなく、「事後審査制＋プログラムによる自動診断」。
   参入の障壁は低い反面、品質や安全面が担保されていない。

発表の終わりに

UBITEQ
UBIQUITOUS TECHNOLOGY

Androidを入り口として、iPhone、FirefoxOSと比較研究を行った結果、Androidは例外として「スマホは危険」という図式が必ずしも正しいとは言えないのではないか。

△スマートフォンは、セキュリティ上危険

↓

○Androidは、セキュリティシステムを理解せずに
開発されたアプリが危険
＋iPhoneは（Jailbreakしなければ）結構安全
＋Firefoxも（セキュリティポリシーを貫ければ）それなりに安全

将来的には、モバイルアプリのセキュリティ対策も基本ポリシーが洗練され、検証手法もよりベーシックなソフトウェア開発に対するアプローチへと回帰していくのではないだろうか。

参考文献

UBITEQ
UBIQUITOUS TECHNOLOGY

- IPAテクニカルウォッチ「Androidアプリの脆弱性」に関するレポート
  （独立行政法人情報処理推進機構　技術本部　セキュリティセンター）

  ⇒ 最初の入り口として最適。初心者にも分かりやすく整理されています

- Androidアプリのセキュア設計・セキュアコーディングガイド
  （一般社団法人日本スマートフォンセキュリティ協会 セキュアコーディンググループ）

  ⇒ サンプルコードとルールブックが記載され、充実の内容です

- Android Security 安全なアプリケーションを作成するために
  （タオソフトウェア株式会社[著]、インプレスジャパン発行）

  ⇒ 上記2つの資料の内容とあわせて、不明点を解決できます

ご清聴、有り難うございました。

Androidアプリに限らず、第３のOSについても
更に研究活動を継続していく予定です。
今後ともどうぞよろしくお願い致します。