MITB in Android

Fourteenforty Research Institute, Inc.
株式会社 フォティーンフォティ技術研究所
http://www.fourteenforty.jp

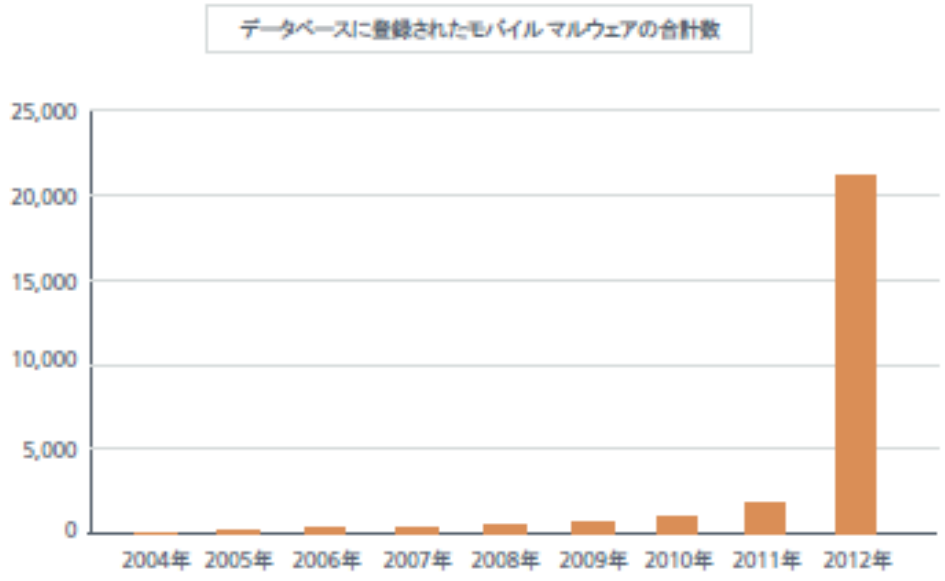
Ver 2.00.01

アジェンダ

- 背景
- MITBとは
- MITB in Androidの可能性
- Firefox for mobileを用いたMITB
- デモ
- まとめ

背景：Androidの普及とMan in the Browser

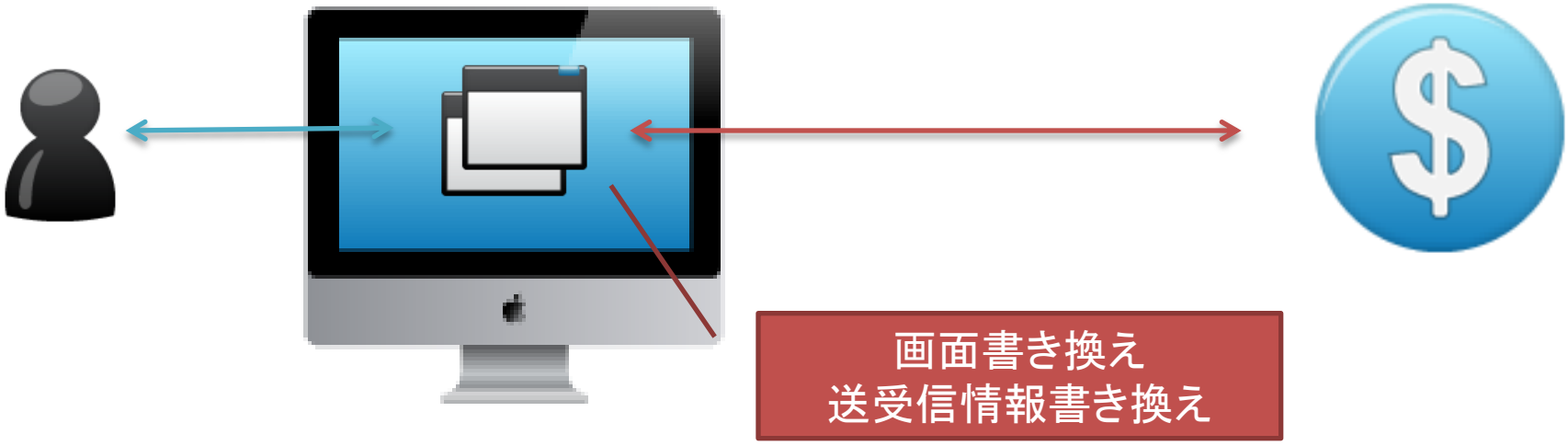
- Androidマルウェアの増加
- 従来からのWindows PCマルウェアの攻撃手法の高度化
  - オンラインバンクを狙ったMan in the Browser (MITB)



McAfee脅威レポート 2012年第3四半期より転載

Man in the Browser (MITB)とは

- ブラウザ内に侵入して、画面を書き換える、送信されるデータを書き換える、パスワードを盗むなどを行う攻撃手法
- 主にオンラインバンクへのアクセスを監視、ユーザーの入力の搾取、改ざんを行う
- 二要素認証を用いても、**正規のセッション、パスワードを攻撃時に用いる**こともできるため防げない



MITBの現状（海外）

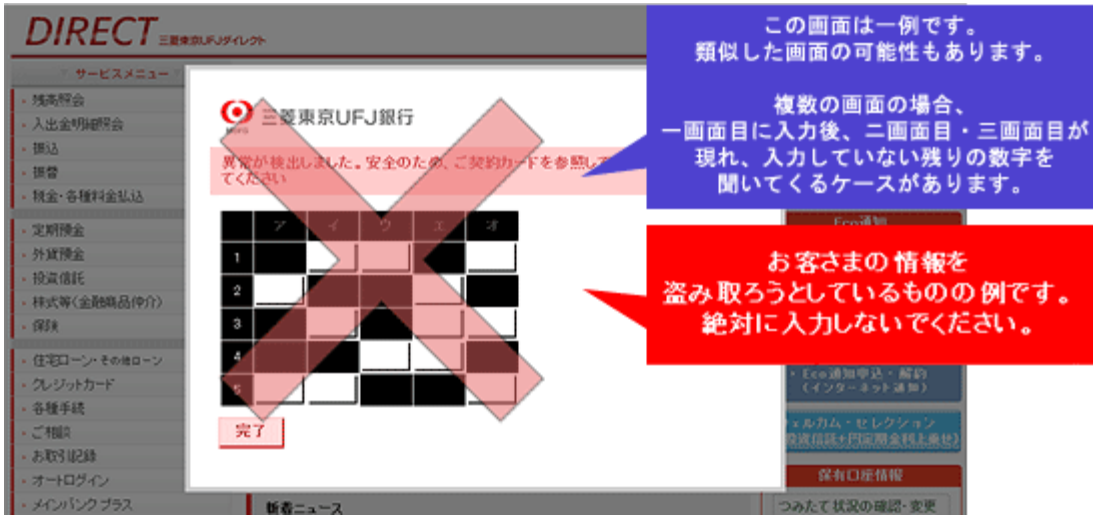
- Operation High Roller※
  - 2012年 US、ヨーロッパを中心に行われたMITB攻撃
  - 2ヵ月間で最大で20億ユーロの被害が発生（およそ2,000億円）

※McAfeeホワイトペーパー Operation High Rollerより

MITBの現状（国内）

- ZeusやSpyeyeといったツールキット
  - 2012年、国内でも銀行などを対象にした攻撃が現実に起きている
    - 三菱東京UFJ銀行
    - 三井住友銀行
    - みずほ銀行

    など



画面は http://www.bk.mufg.jp/info/phishing/ransuu.html より引用

脅威予測

スマートフォンユーザーの増加

スマートフォンによるオンラインバンク利用者の増加

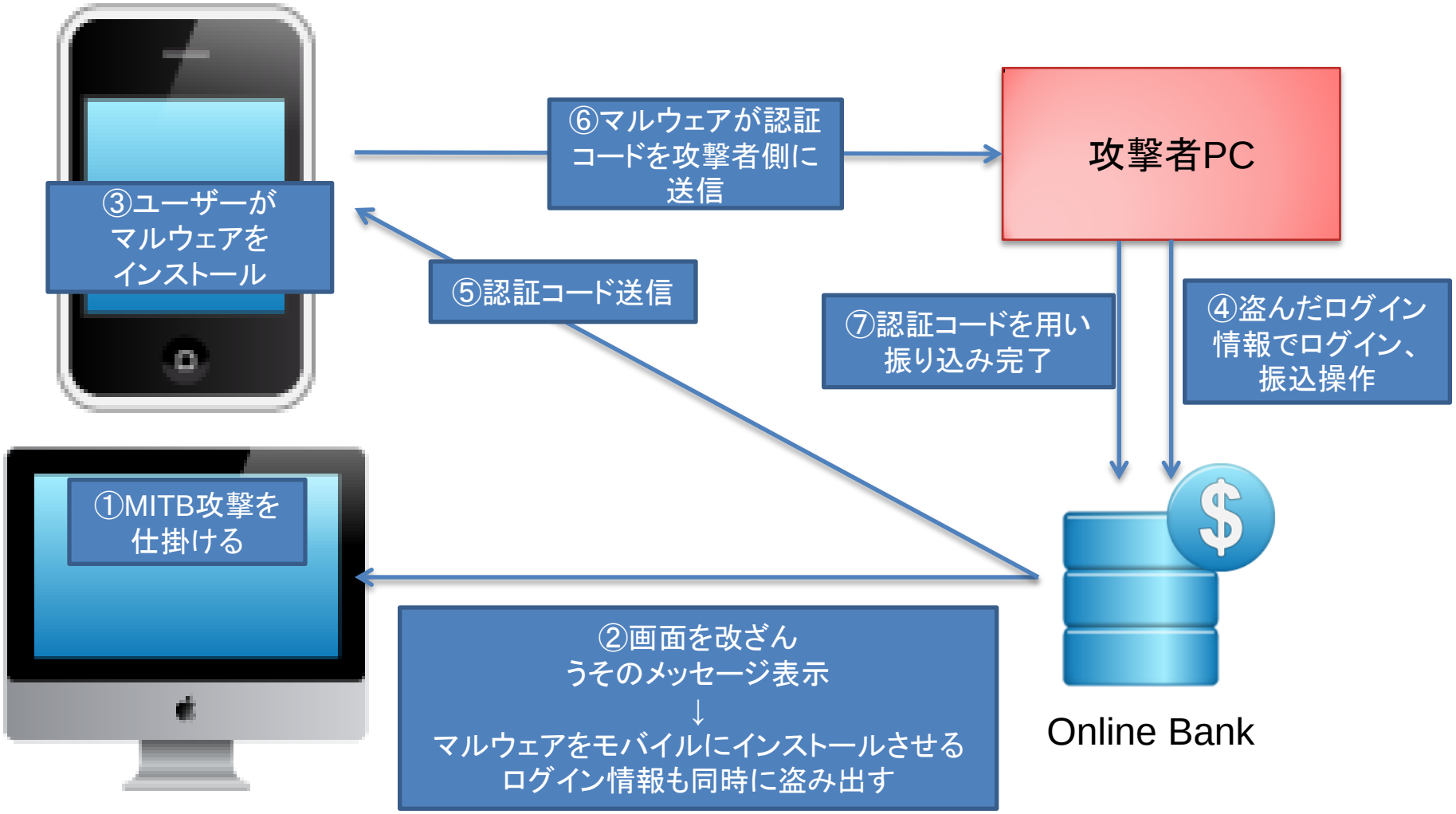
スマートフォン上で、MITB攻撃が成立するのか
対策方法があるのか

Man in the Mobile (MITMO)とMITB

- MITBと似た用語としてMITMOがある
  - 別の概念
- モバイル端末にアプリをインストールさせることで、**SMSメッセージを利用した認証を回避**
- MITBと組み合わせて攻撃に利用される
- MITMO ≠ MITB

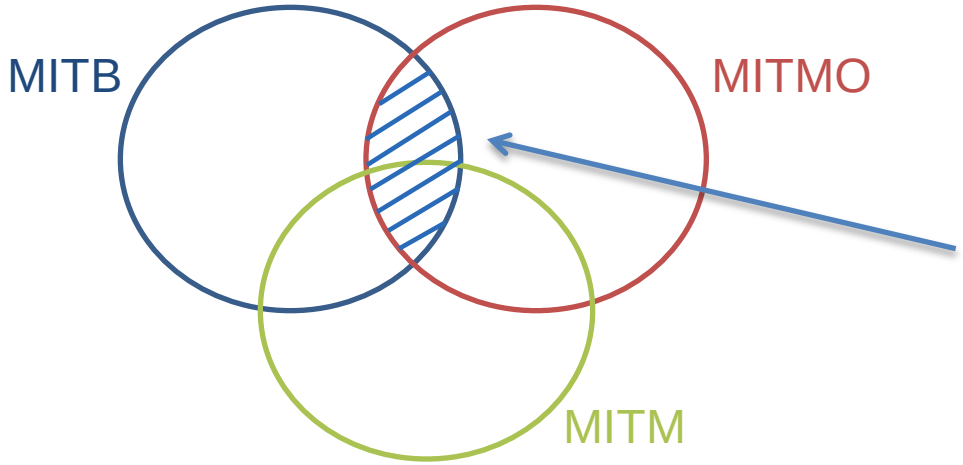
- 典型的なシナリオは・・・

Man in the Mobile (MITMO)の流れ



MITB, MITMO, MITM(Man in the Middle)の関係

- MITBやMITMOは攻撃コードやデータが存在する場所に着目した分類
  - ブラウザ内：MITB
  - モバイル端末内：MITMO
- MITMは攻撃の形態の一つ。
  - MITBやMITMOと組み合わせて利用される
  - それ以外のものも存在する
- それぞれ独立した概念



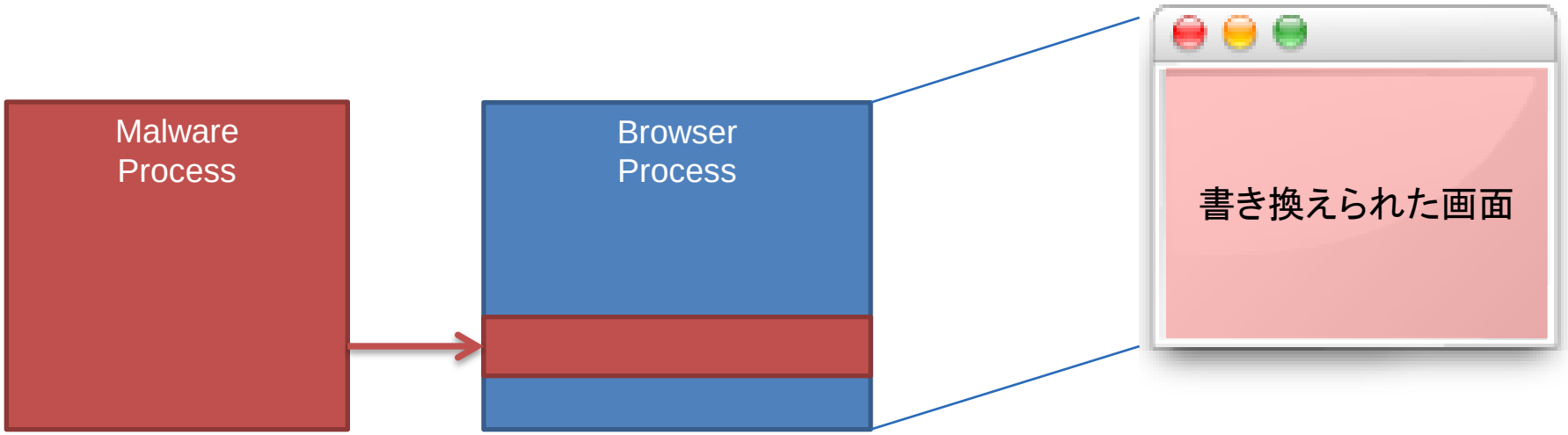
今回は、この領域の可能性について考える

MITB in Android

- Android端末のブラウザに侵入
  - 画面書き換え
  - 送受信情報の書き換え
- 現状では現実の脅威の報告例はない

MITB in Windows

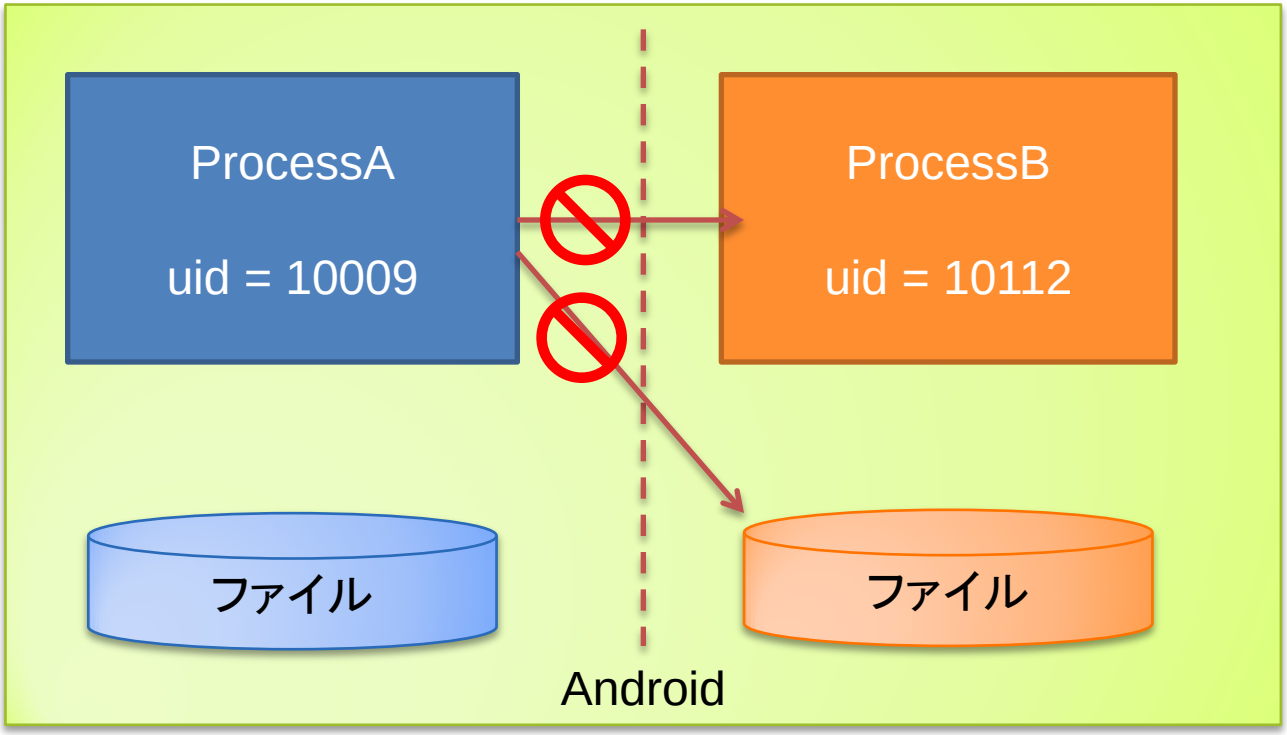
- 典型的な手法
  - メールなどでマルウェアをユーザーに配布、実行させる
  - IEなどのブラウザプロセスのメモリを書き換え
  - 特定のURLへの接続を見張る



コードインジェクション

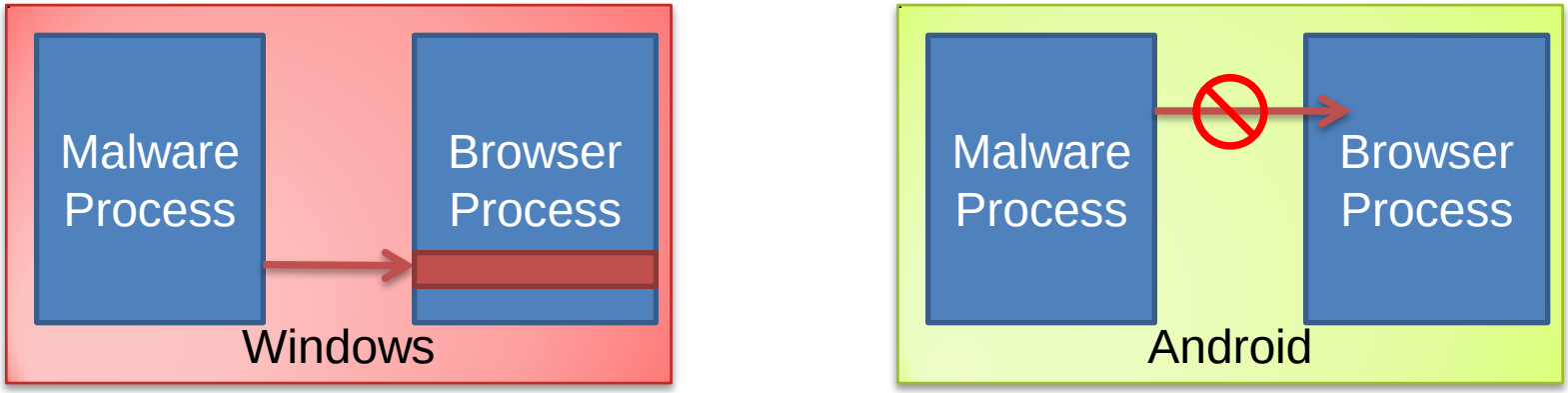
Androidのセキュリティアーキテクチャ

すべてのアプリケーションが原則別ユーザーで動作
メモリ、ファイルにアクセスできない



AndroidとPC(Windows)との大きな違い

- Windowsで起きるMITBがそのままAndroidで起きるか？
  - Windowsでは同じユーザーで動作させている他のプロセスのメモリを変更可能
  - MITBの基本的な手法として利用
- Androidでは各プロセス（アプリ）が別ユーザーとして動いており、他のプロセスにアクセスできないように設計されている
- Androidマルウェアをインストールしてしまってもブラウザそのものへの影響は**原則ない**



Androidではマルウェアが直接ブラウザに介入することができない

Man in the Browser in Androidの可能性

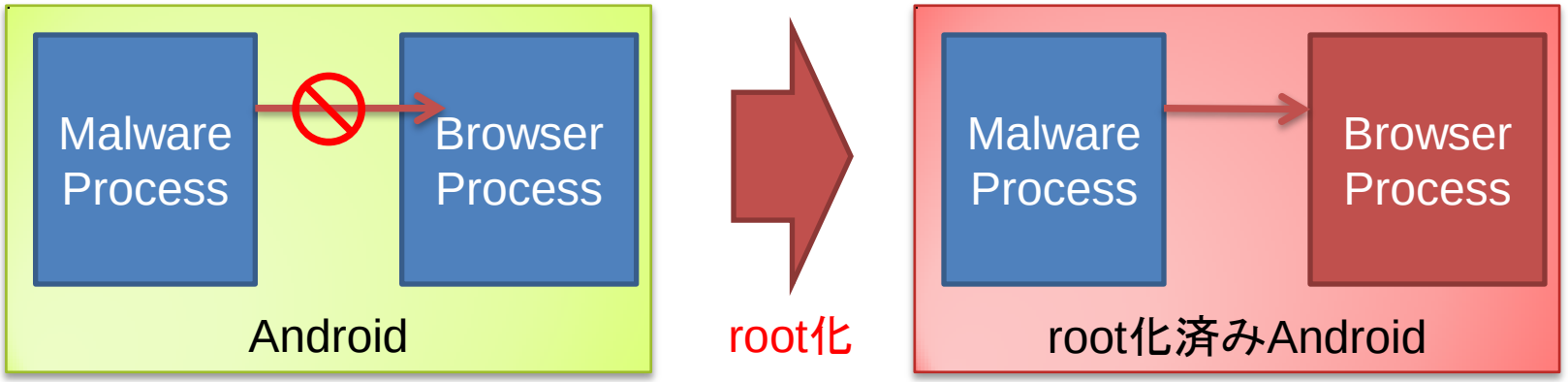
- AndroidにおいてMITBを成功させるには？
  - 相手プロセスのメモリへの侵入方法を考える
- Androidにおいて、ブラウザに介入できるとすれば、以下の4つの可能性が考えられる
  - root化端末への侵入
  - Androidシステム、アプリの脆弱性
  - Class Loading Hijacking脆弱性
  - Browser Extension

root化端末への侵入

- root化端末では、他のプロセスへの介入が可能となる
- root権限を持つプロセスからはメモリの書き換えやファイルの置き換えなどが自由に可能
  → 無防備な状態
- プロセスのメモリの直接の書き換えのほか、アプリケーションの置き換えや、Dalvik-Cacheの置き換えなど、さまざまな手法でMITBを実現できる

- 対策
  - root化をした状態でAndroidを利用しない

root化端末への侵入



root化によりマルウェアがブラウザに介入できるようになる

LSMの効果

- root化対策
  - LSM（Linux Secure Modules)によって、root権限でできることを制限
  - ただし、LSMが有効でも、他のプロセスへの介入が可能になる例も(*1)
  - root化した端末ではMITBに対するリスクは相対的にはかなり高い

*1 http://www.fourteenforty.jp/assets/files/research/research_papers/yet-another-android-rootkit.pdf

Androidシステム自体の脆弱性

- Androidシステム自体に脆弱性がある場合
  - root化
  - ブラウザプロセスの乗っ取り

  の可能性
- ブラウザが読み込むライブラリに脆弱性
  - 任意のコードが実行可能であった場合、MITBは可能となる
  - ASLRやDEPなどの実装 → このタイプの攻撃は困難に

Version	-2.2	2.3-,3.0-	4.0-	4.1-
DEP(スタック)	×	○	○	○
DEP(その他)	×	○	○	○
ASLR(スタック)	○	○	○	○
ASLR(ヒープ)	×	×	△	○
ALSR(モジュール)	×	×	△	○

AndroidのDEP, ASLRへの対応状況
http://www.fourteenforty.jp/assets/files/research/research_papers/InternetWeek2011_s10-02.pdf
より一部修正、追記して転載

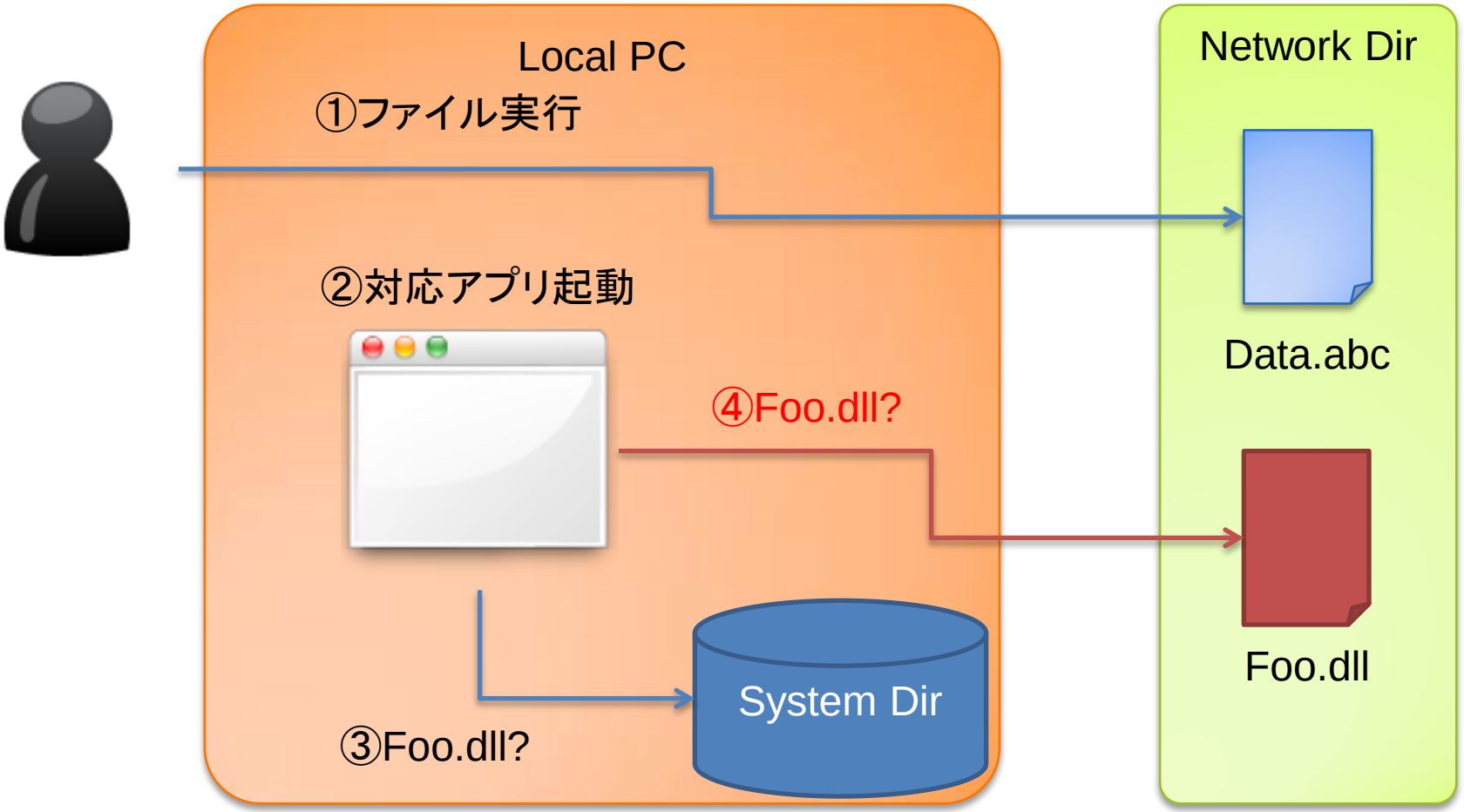
Androidシステム自体の脆弱性

- 対策
  - アップデートを適切に行う

Class Loading Hijacking脆弱性の利用

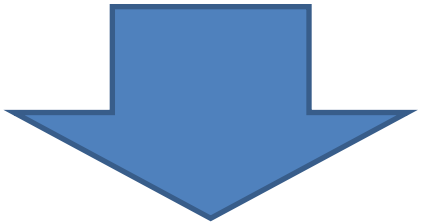
- Class Loadingという外部のDEXコードを読み込む機能
  - ファイルなどをコードとして取り込める
- WindowsのDLL Hijackingと同様の脆弱性の可能性
- 2011年シマンテックより発表された
  - http://www.symantec.com/connect/blogs/android-class-loading-hijacking

Windows DLL Hijacking



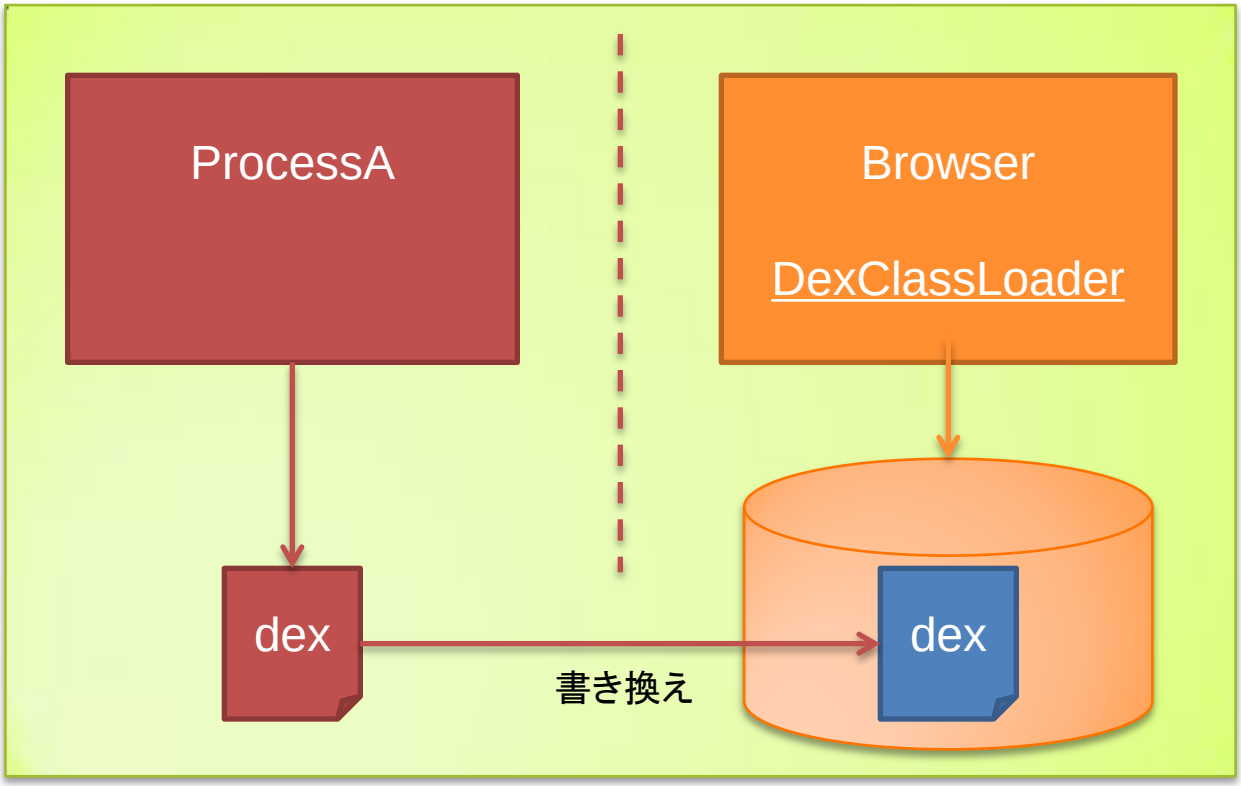
DexClassLoader

- classes.dex (複数のコードのアーカイブ)を動的に読み込む
- 引数
  - dexPath : 読み込むdexファイルのパス
  - optimizedDirectory : 最適化適用後のodexファイルの出力パス



いずれかのパスが他アプリから書き換え可能な場合脆弱

Class Loading Hijacking脆弱性の利用



Class Loading Hijacking脆弱性の利用

- Android APIのドキュメントにも注意書き
  - 主要ブラウザでこのような問題を作りこんでしまう可能性は低い
  - 標準ブラウザ、Chrome, Firefoxを調べた限りは現状脆弱性はない

- 対策
  - システム（アプリ）のアップデート

Browser Extension

- Android版Firefoxはブラウザアドオンをサポート
  - 悪意あるアドオンを導入
  - 画面等を書き換えられる可能性がある
- アドオンが安全かどうか判定する明確な方法はない
- AMO(addons.mozilla.org)には、審査を通ったもののみが登録されている
- 対策としては、ウェブページなどで促されるままにAMO以外からアドオンを導入しないこと

MITB in Androidの可能性と対策

ユーザーが行える対策をまとめると、以下のようになる

可能性	対策
root化端末への侵入	root化を故意に行わない root化した端末を利用しない
Androidシステムの脆弱性	システムのアップデート
Class Loading Hijacking	ブラウザのアップデート
Browser Extension(Firefox)	AMO以外からのアドオンのインストールを控える

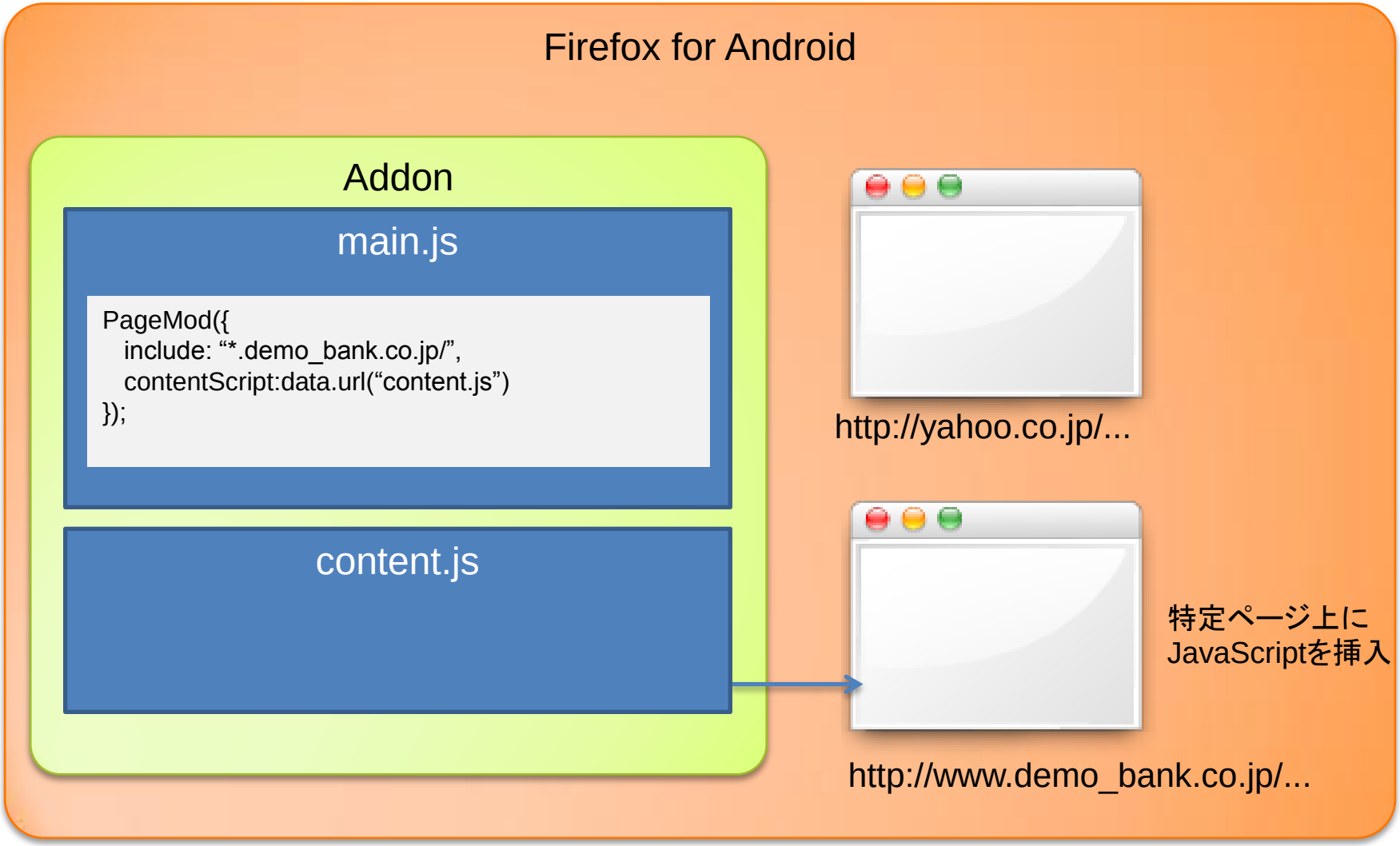
(参考)Browser以外での脅威

- ブラウザ以外の銀行専用アプリへの攻撃
  - MITBではない
  - ただし、こちらも攻撃される可能性あり
  - Class Loading Hijackingなどの脆弱性への攻撃
- 専用アプリに似せた偽アプリを利用した攻撃の可能性

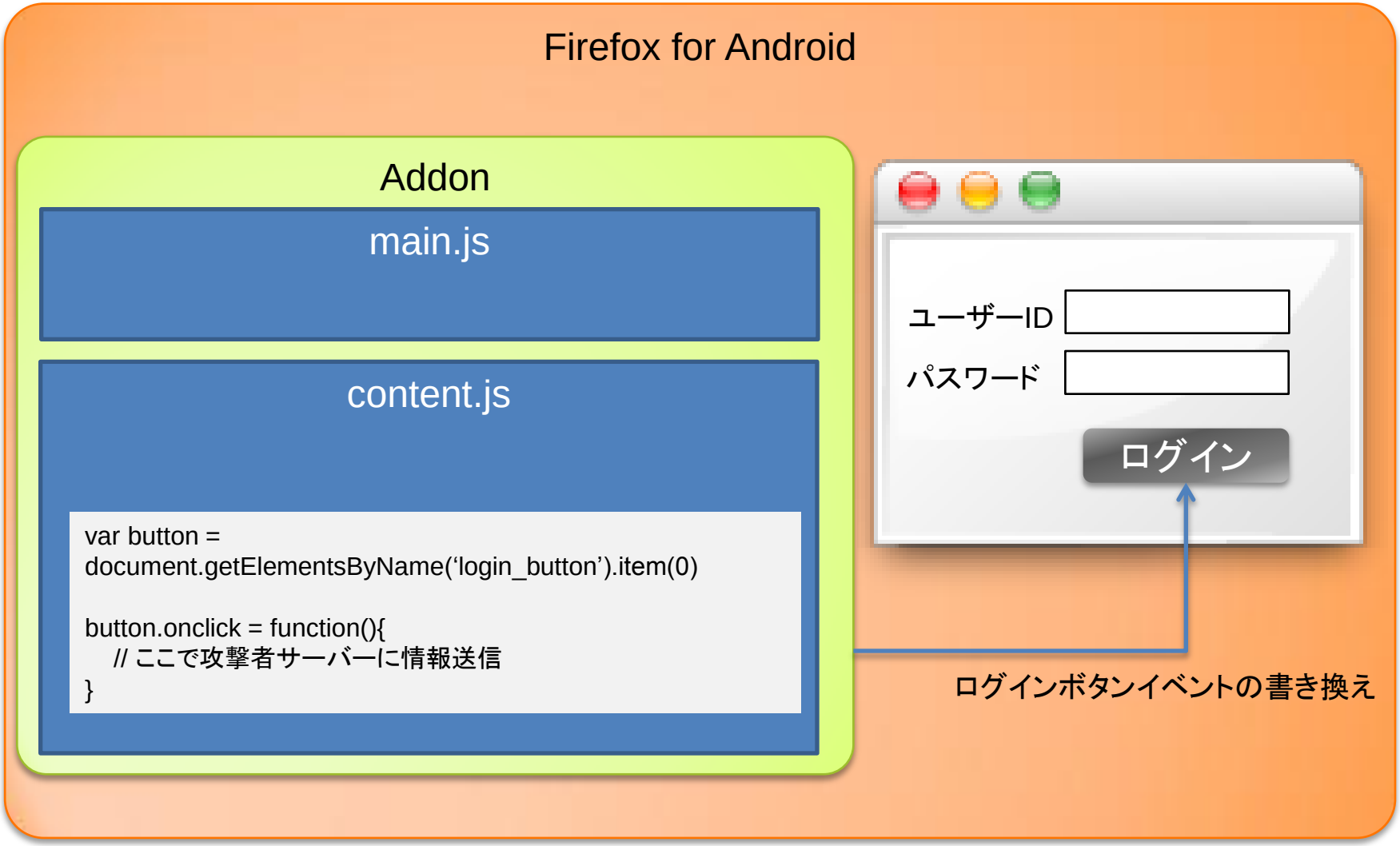
First PoC of Firefox MITB Addon

- ユーザーが悪意あるアドオンをFirefoxにインストールしてしまった場合何ができるのか？
  - 画面の書き換え
  - 情報の読み取り（パスワードなど）
  - 送信データの書き換え

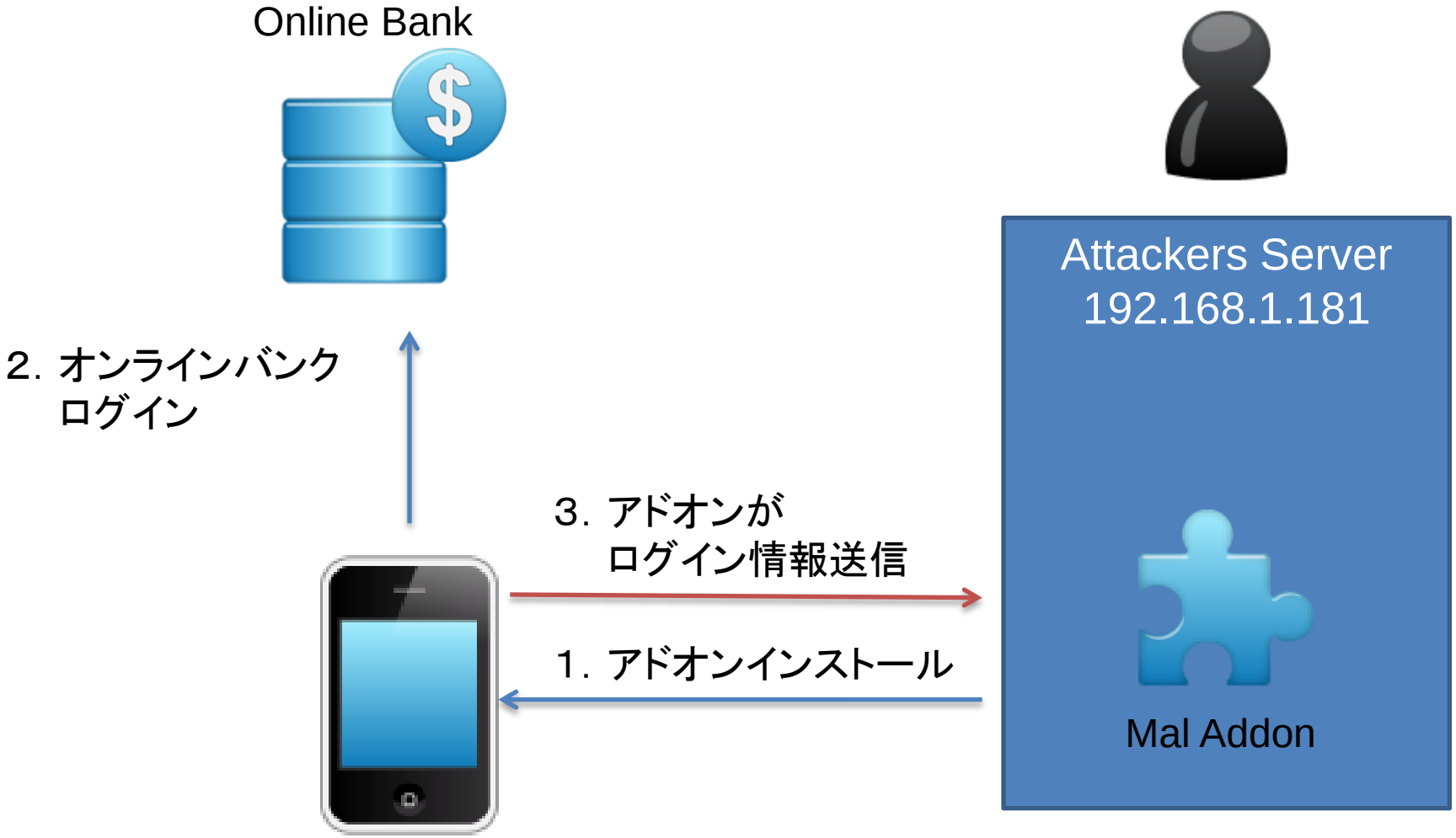
Addonの構造1

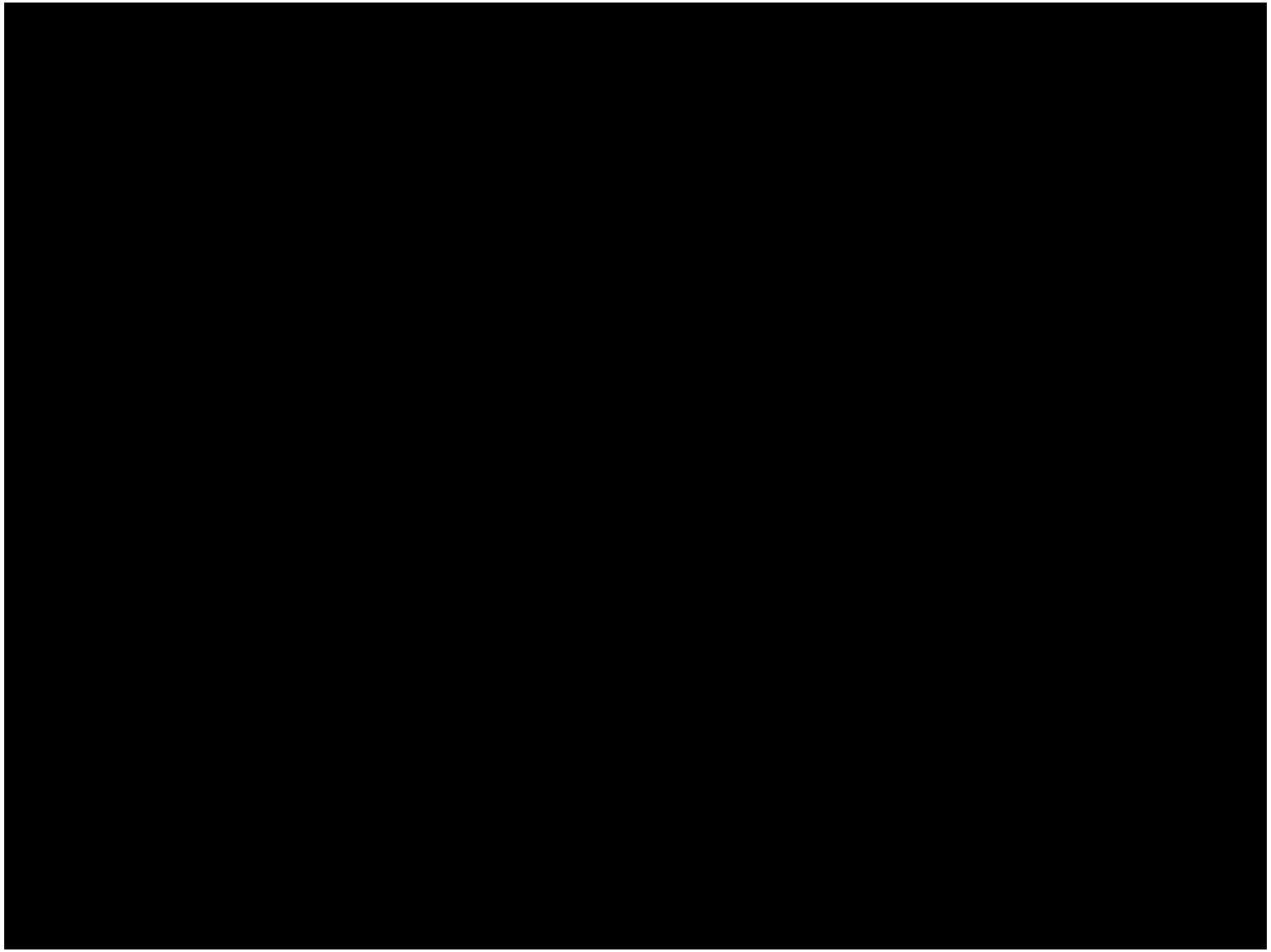

Firefox for Android
Addon
main.js
PageMod({
include: “*.demo_bank.co.jp/”,
contentScript:data.url(“content.js”)
});
content.js
http://yahoo.co.jp/...
http://www.demo_bank.co.jp/...
特定ページ上に
JavaScriptを挿入

Addonの構造2


Firefox for Android
Addon
main.js
content.js
var button =
document.getElementsByName('login_button').item(0)
button.onclick = function(){
// ここで攻撃者サーバーに情報送信
}
ユーザーID
パスワード
ログイン
ログインボタンイベントの書き換え

デモ


Online Bank
2. オンラインバンク
ログイン
3. アドオンが
ログイン情報送信
1. アドオンインストール
Attackers Server
192.168.1.181
Mal Addon



Firefox Addon Install without User Awareness

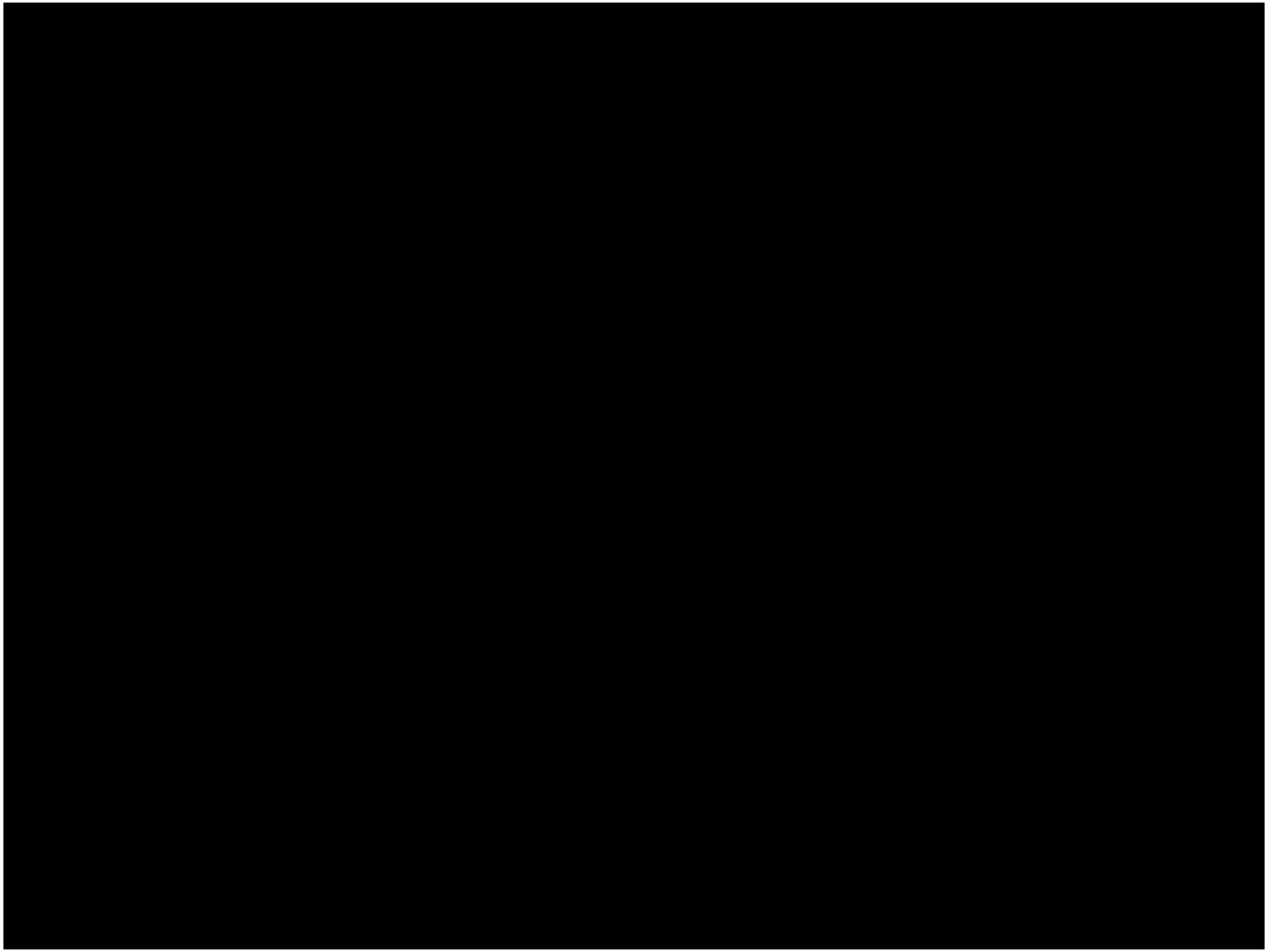
- ユーザーに許可なく、悪意あるAddonをFirefoxにインストールことができるか？
- できる場合、root化なし、ユーザーによる悪意あるアドオンのインストールなしで、MITBが成立する

- Firefoxの脆弱性を発見(CVE-2013-0798)
- 悪意あるアプリをインストール、実行してしまうと、ユーザーに気づかれることなくアドオンをインストール可能

Firefox Addonを利用した脆弱性攻撃シナリオ

- ユーザーが悪意ある（無害なアプリを装う）アプリをインストール、起動
- ユーザーがFirefox for Androidを利用して無害なAddonをインストール
  - ユーザーが無害なAddonを選択、「インストール」をタップ
  - Firefox for Androidが確認画面を表示
  - この際、バックグラウンドで動作している悪意あるアプリはでテンポラリディレクトリにダウンロードされたアドオンのファイルを変更
  - ユーザーが確認画面で「インストール」を選択しインストール
  - Firefox for Androidは悪意あるAddonをインストール

  （Firefox for Androidがテンポラリファイルを置き換え可能なパーミッションで保存していることが脆弱）
- ユーザーはいつも通りFirefox for Androidを用いてオンラインバンクなどを利用
- Webページの書き換え、パスワードの搾取が行われる



危険度の比較(攻撃者視点)

	Mal-Addon のインストール	Androidマルウェアによる Addonインストール
審査	△ (AMOによる審査)	△ (Playストアによる審査)
インストール可能性	× (Addonはインストール数が少ない)	○ (Playストア、またはその他の場所からインストールされる可能性が高い)
脆弱性	○ (脆弱性を探す必要なし)	× (脆弱性を突く必要あり)

まとめ

- Androidのブラウザの MITB可能性
  - 有り
  - Windowsに比べるとハードル高
- root化によるリスク大
- root化されてない場合
  - システムの脆弱性
  - ブラウザの脆弱性
  - ブラウザプラグイン
- Androidのセキュリティモデルを適切に運用できることが重要
- Androidならではの対策の難しさも（サードパーティアプリには限界がある）
- MITBに関して言えば、マルウェアを実行してしまうだけで、MITBが可能となってしまう Windows PCを利用するよりも、Androidを利用したほうが安全？
- MITB以外の、フィッシングや、偽アプリにはWindows PC同様注意が必要

怪しいアプリはインストールしない！