続Man in the Browser in Androidの可能性

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

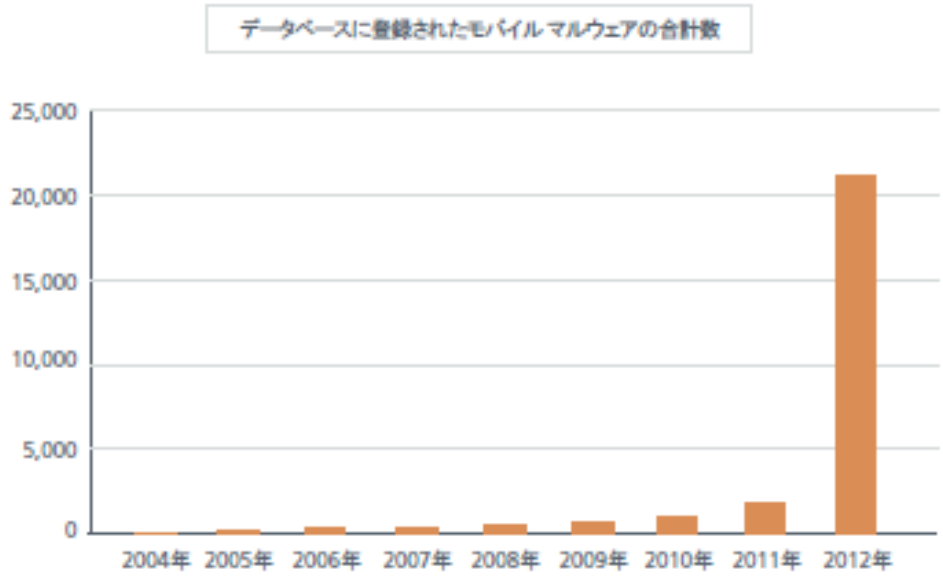
Ver 2.00.01

はじめに

- この資料は、2013年2月28日に行われた「Security Days」 ナイトセッションでの発表資料を一部修正し作成したものです。
- 2012年12月ののマンスリーリサーチ「Man in the Browser in Androidの可能性」をまだお読みでない方は、そちらをご覧になってからお読みいただくと理解しやすいと思います。

背景：Androidの普及とMan in the Browser

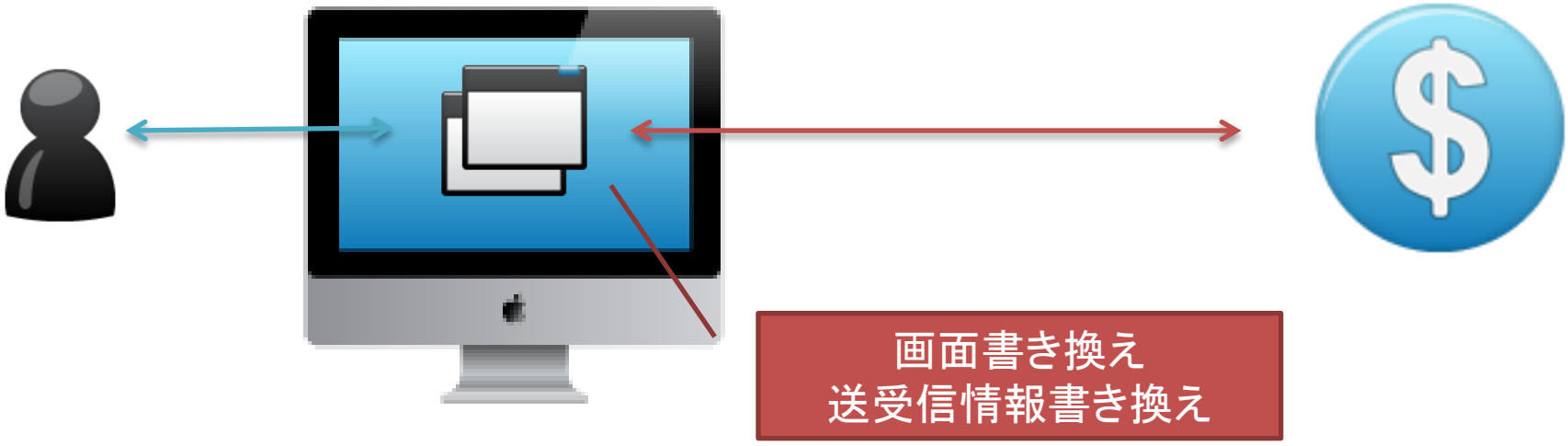
- Androidマルウェアの増加
- 従来からのWindows PCマルウェアの攻撃手法の高度化
  - オンラインバンクを狙ったMan in the Browser (MITB)



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）

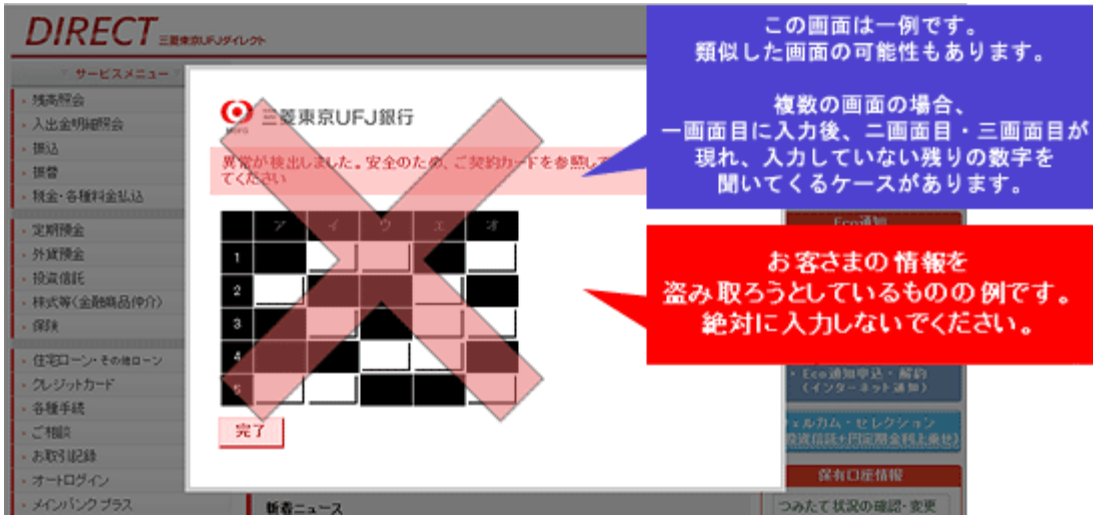
- Operation High Roller※
  - 2012年 US、ヨーロッパを中心に行われたMITB攻撃
  - 2ヵ月間で最大で20億ユーロの被害が発生（およそ2,000億円）

※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

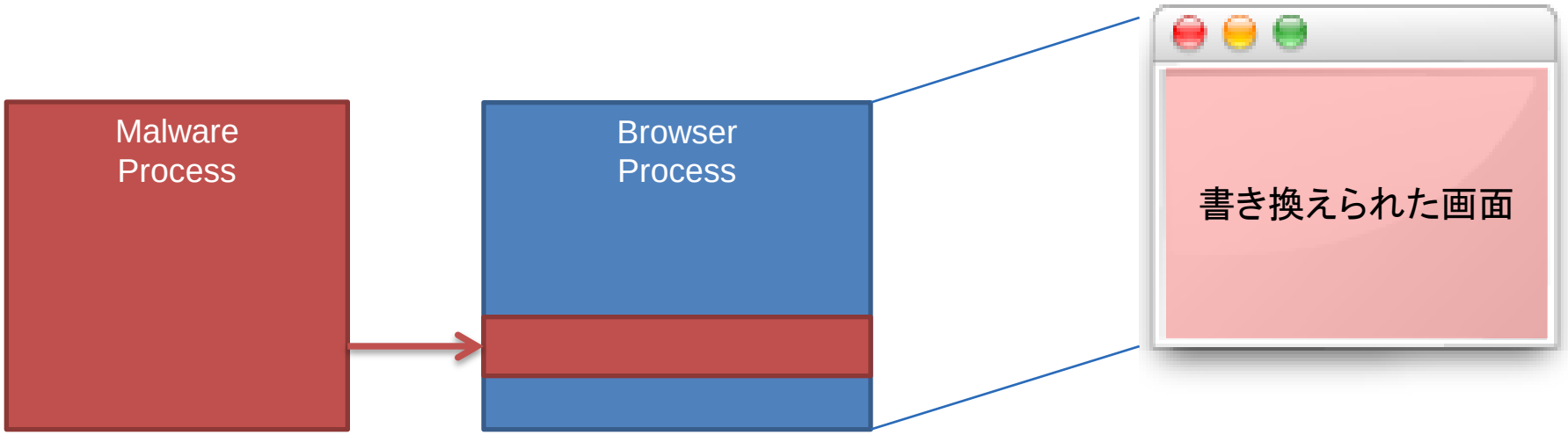
スマートフォン上で、MITB攻撃が成立するのか
対策方法があるのか

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

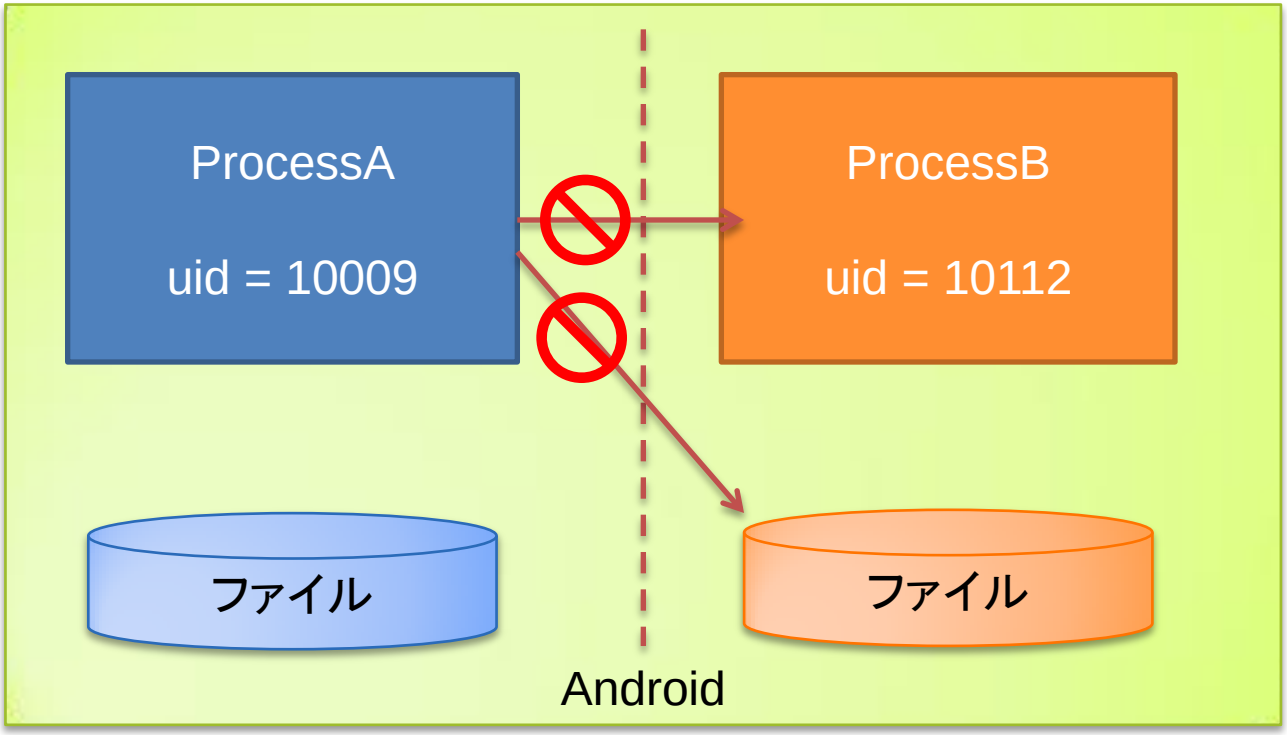
- 典型的な手法
  - メールなどでマルウェアをユーザーに配布、実行させる
  - IEなどのブラウザプロセスのメモリを書き換え
  - 特定のURLへの接続を見張る



コードインジェクション

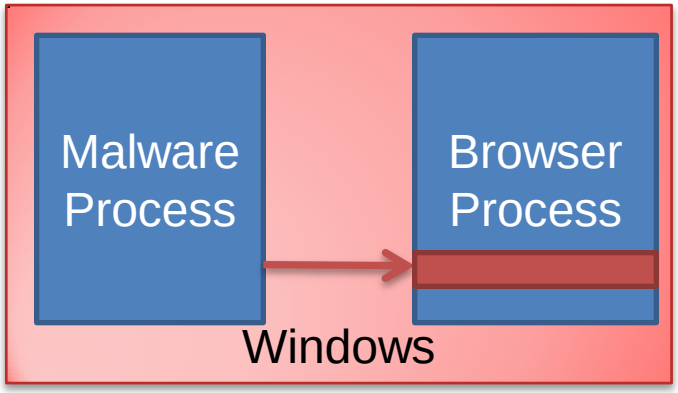
Androidのセキュリティアーキテクチャ

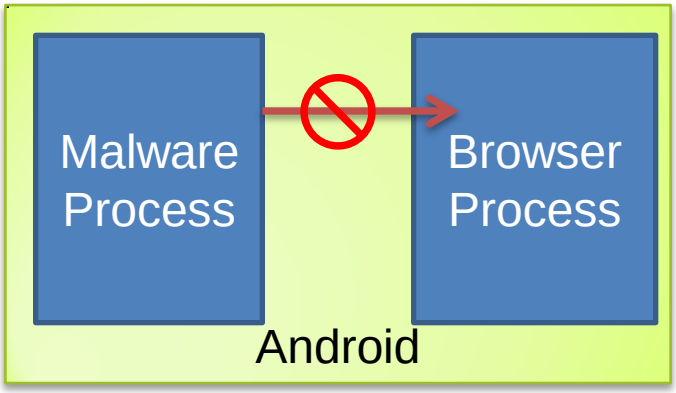
すべてのアプリケーションが原則別ユーザーで動作
メモリ、ファイルにアクセスできない



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

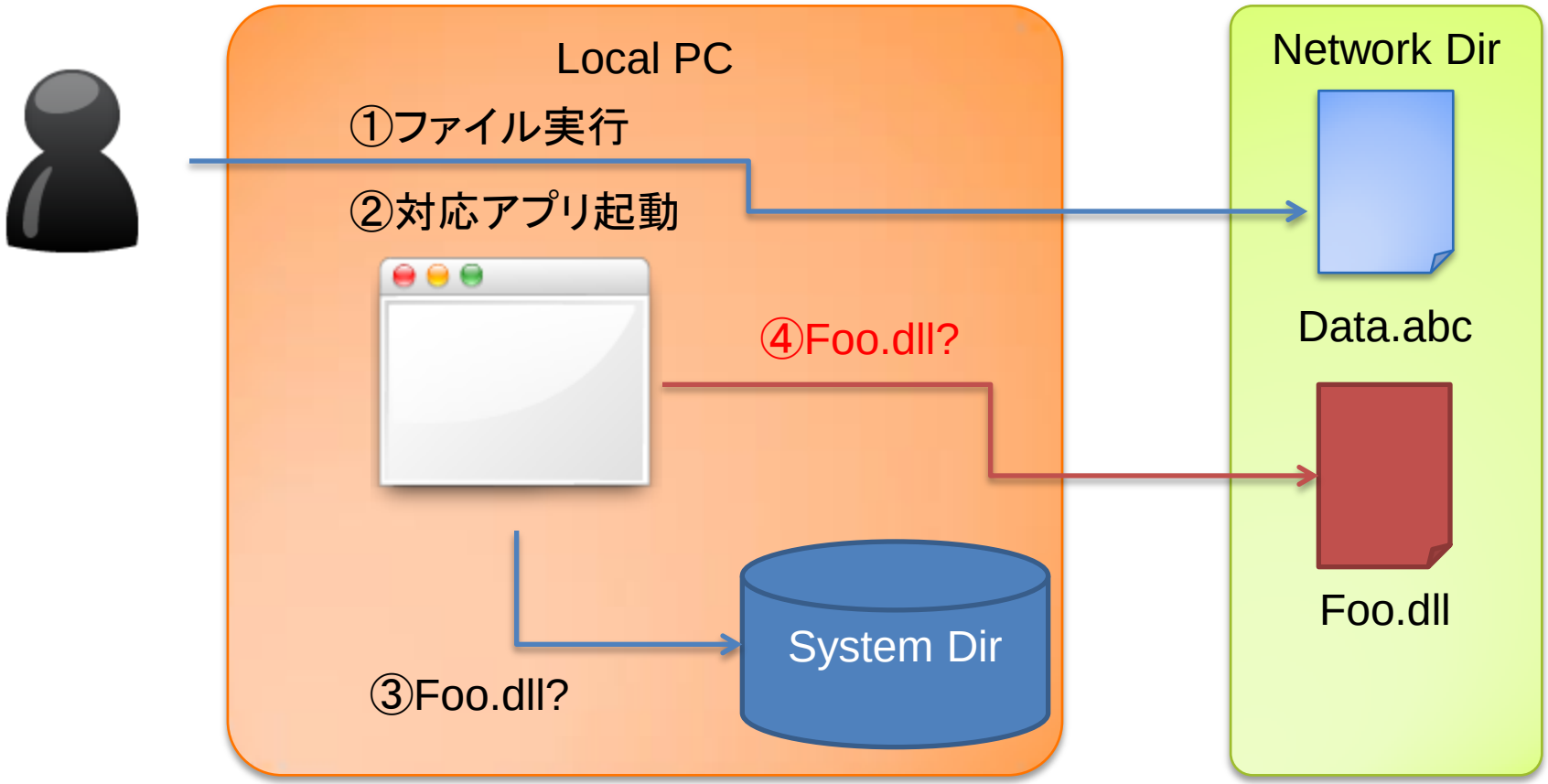
- AndroidにおいてMITBを成功させるには？
  - 相手プロセスのメモリへの侵入方法を考える
- Androidにおいて、ブラウザに介入できるとすれば、以下の4つの可能性が考えられる
  - root化端末への侵入
  - Androidシステム、アプリの脆弱性
  - Class Loading Hijacking脆弱性
  - Browser Extension

- 今回は、下の二つについて詳しく見てみる

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

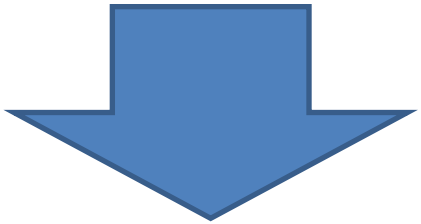
Windows DLL Hijacking



ネットワーク上におかれたファイルを開いたとき、
アプリケーションがFoo.dllを読み込もうとするが、
ローカルに見つからないと、ネットワーク上のものを読んでしまう。

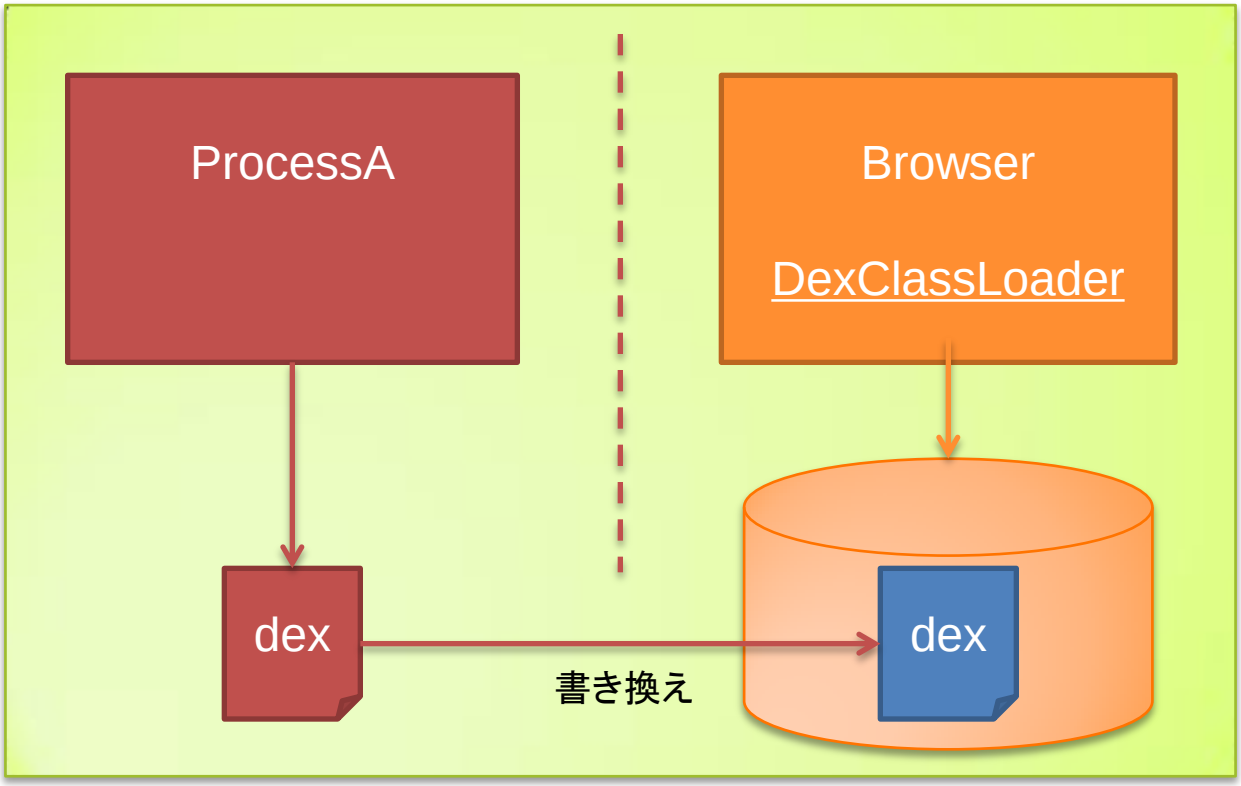
DexClassLoader

- 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



ブラウザが読み込むdexファイルを書き換えられるとすれば、ブラウザに侵入可能

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

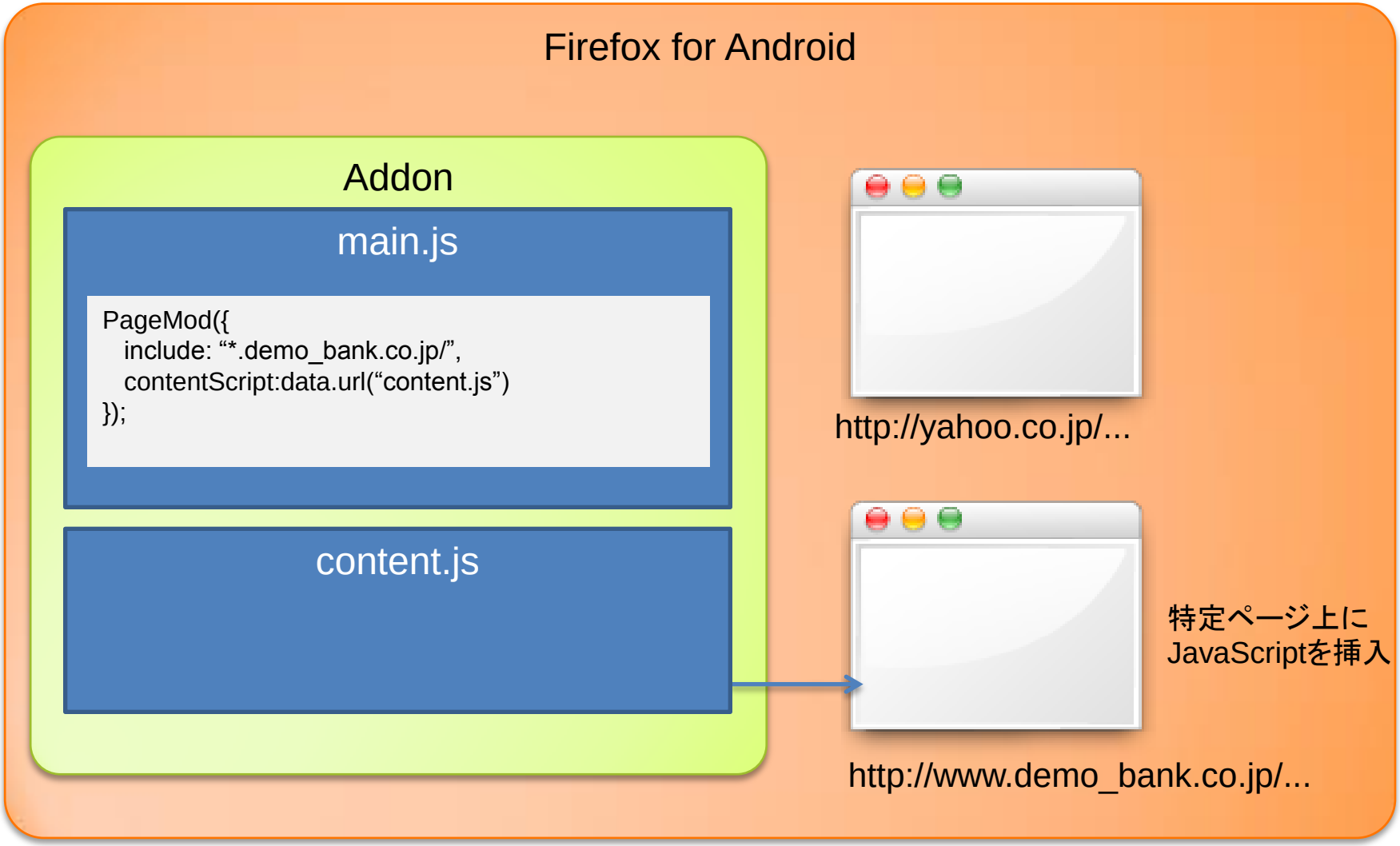
- Android版Firefoxはブラウザアドオンをサポート
  - 悪意あるアドオンを導入
  - 画面等を書き換えられる可能性がある
- アドオンが安全かどうか判定する明確な方法はない
- AMO(addons.mozilla.org)には、審査を通ったもののみが登録されている
- 対策としては、ウェブページなどで促されるままにAMO以外からアドオンを導入しないこと

First PoC of Firefox for Mobile MITB Addon

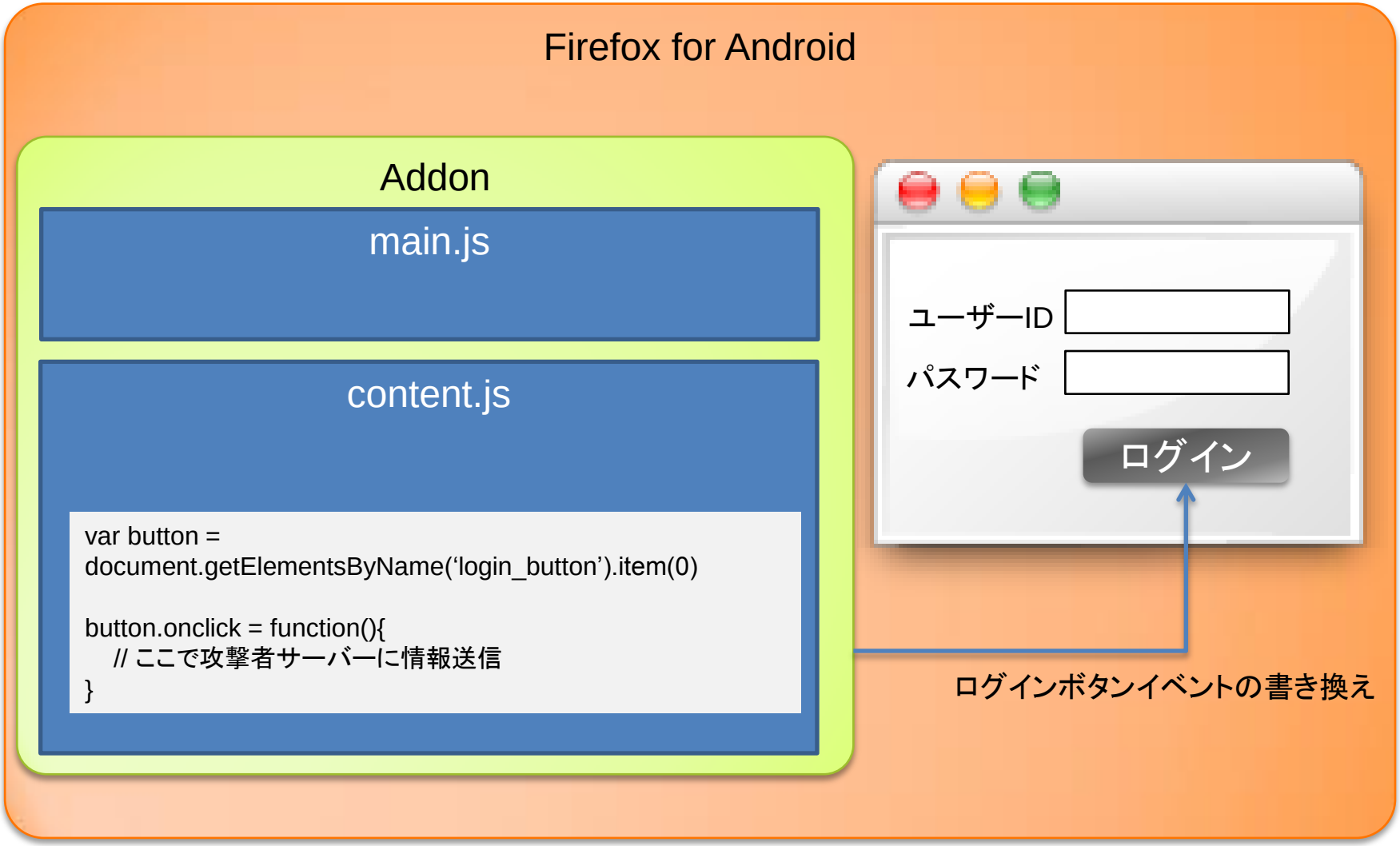
- ユーザーが悪意あるアドオンをFirefoxにインストールしてしまった場合何ができるのか？
  - 画面の書き換え
  - 情報の読み取り（パスワードなど）
  - 送信データの書き換え

- 実際にどのような動きになるのか

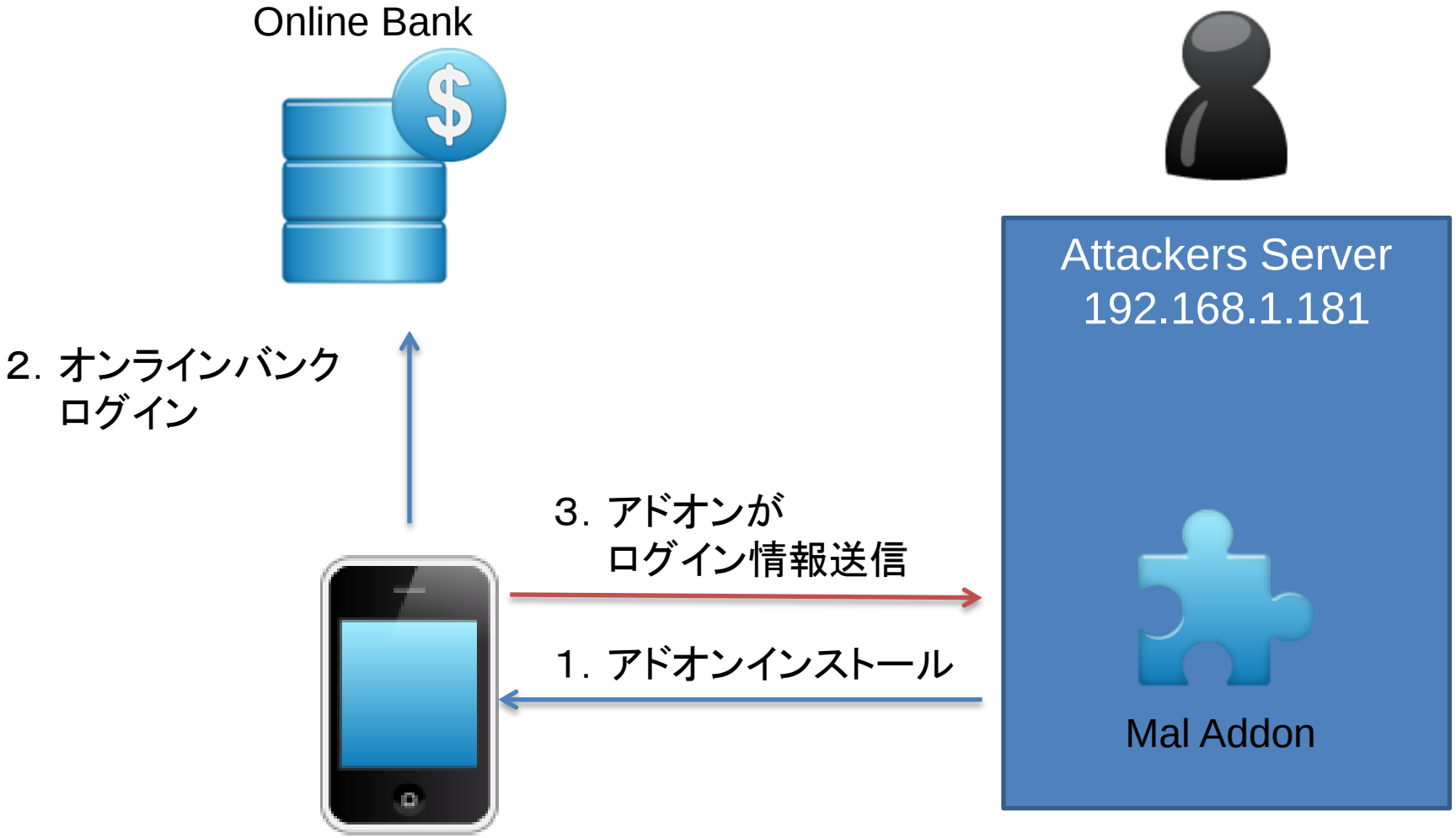
Addonの構造1


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
Attackers Server
192.168.1.181
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
Mal Addon

最初に、悪意あるアドオンをインストール
デモ用に作成した銀行サイトにログインすると、攻撃者側に
ユーザーIDとパスワードが送信されていることがわかる。

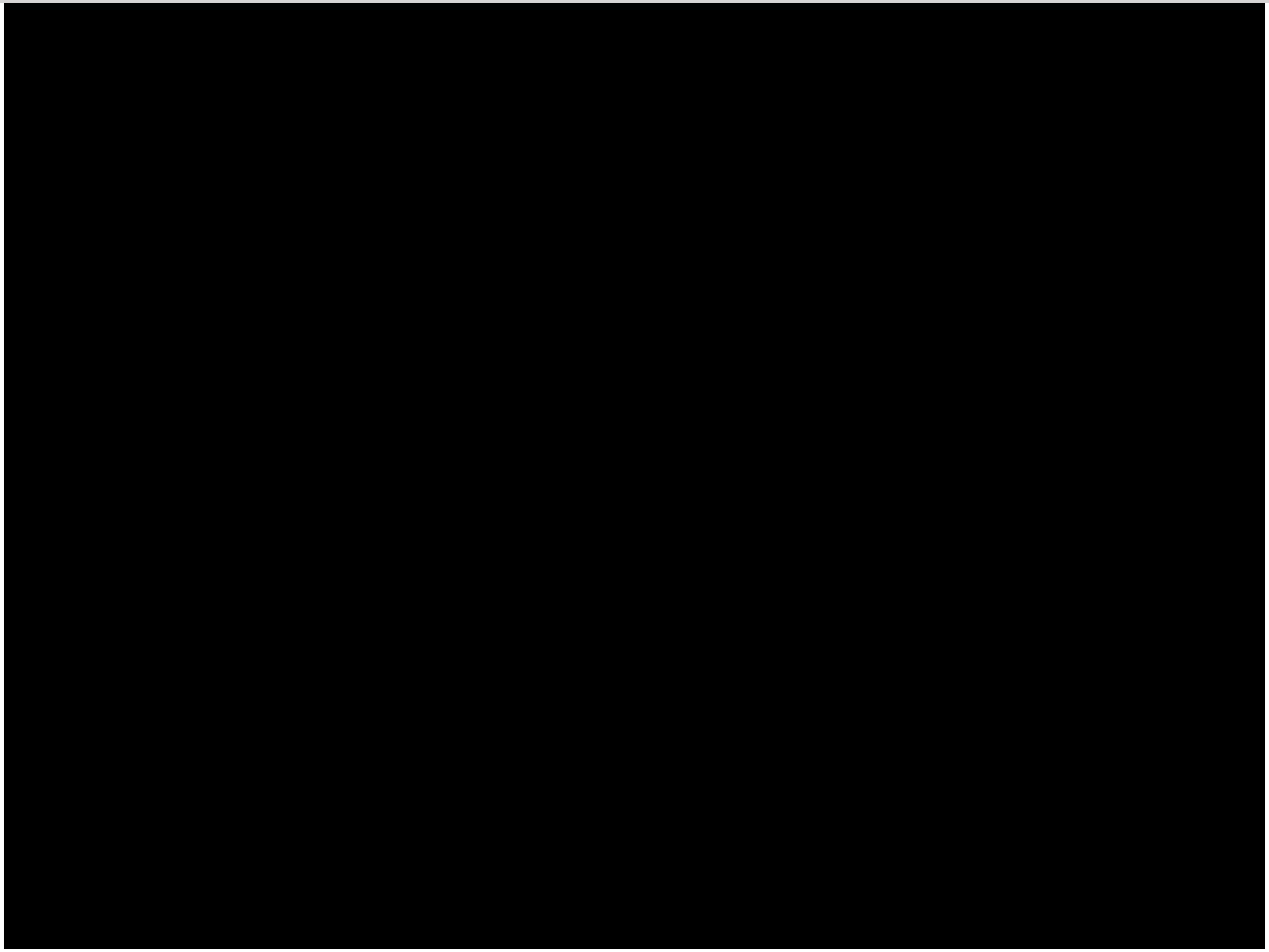
Firefox Addon Install without User Awareness

- 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（すでに報告済み）
- 悪意あるアプリ（特殊なパーミッションは必要なし）をインストール、実行すると、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

（方法の詳細は現在は公開不可）

→その結果どのようなことが起きるかは、先ほどお見せした通り。



今回は、APKファイルをインストール後、起動する。
その後Firefoxで、とある操作を行うが、現在は公開できないため暗転
この操作は、特別なものではなくユーザーが通常利用で行う操作
このタイミングで、悪意あるアドオンがインストールされる

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？

  → 脆弱性を突かないとMITBが成功しないという点ではAndroidのほうが安全と言える
- 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！