5ZA-01

Cordovaハイブリッドアプリケーションにおけるプラグインに着目したリパッケージング攻撃の防御手法

秋本　裕史†　　岸　知二‡

早稲田大学大学院 創造理工学研究科 経営システム工学専攻† ‡

1. はじめに

HTML5ハイブリッドアプリケーション(以下，ハイブリッドアプリ)は，Web 技術で開発の大半を行うモバイルアプリケーションである．利点として開発コストが低いことが挙げられる反面，脆弱性の問題が指摘されている．通常のモバイルアプリケーションとは異なり，コンパイル時にソースコードが変換されずにアプリ内に保存されるため，アプリ内のソースコードの改ざんが容易である．特にフレームワーク Apache Cordova(以下，Cordova)を利用して開発したハイブリッドアプリ(以下，Cordova アプリ)の場合，プラグインを不正利用するコードを注入することにより，端末に保存されている連絡先などの情報の漏洩が懸念される．また，Android アプリの場合は，サードパーティマーケットでの再配布が容易なため，悪意のあるコードを注入した上でアプリの再配布を行う，リパッケージング攻撃に対して脆弱である．本稿では，データフロー解析を利用し，Cordova アプリにおいて，プラグインの不正利用による機密情報の漏洩を目的とした，リパッケージング攻撃の防御手法を提案する．対象 OS は，Android とする．

2. 関連研究

Android におけるハイブリッドアプリのセキュリティに関連した研究の例として，以下を挙げる．[1]の研究では，ハイブリッドアプリに Content-Security-Policy を自動適用し，外部からの悪意のある JavaScript コードの注入を防ぐ手法を提案している．[2]の研究では，Cordova アプリにおけるプラグインのアクセス制御を，ユーザー側が行うことのできる仕組みを提案している．アプリの使用中，プラグインの使用の許可をユーザーに尋ねることにより，ユーザー側で怪しいと感じたプラグインの使用の拒否が可能となる．リパッケージング攻撃によってプラグインの不正利用を行うコード注入されても，防ぐことが可能である．

3. 研究アプローチ

リパッケージング攻撃は，内部のファイルの書き換えを行うため，[1]の手法の適用によって防ぐことは難しい．また，[2]の手法では，ユーザーの判断によって，プラグインの使用を決める．そのため，本来使用すべきプラグインの拒否や，機密情報の漏洩の可能性のあるプラグインを許可してしまう場合がある．これらの問題について，プラグインの使用によって機密情報の漏洩の可能性があるもののみを判断し，ユーザーにその情報を提示することで，改善できると考えた．そこで，本研究では，データフロー解析を利用してその判断を行う．

データフロー解析とは，変数の値がどのような経路をたどって伝播するかなどの情報を収集する技法である．この技法により，プラグインが取得した機密情報の経路を辿り，最終的に外部への漏洩の可能性を判断する．機密情報については，Android のパーミッションの保護レベルが Dangerous に設定されているプラグイン(以下，機密情報プラグイン)が取得するデータとする．

ただし，機密情報の外部への送信について，不正に注入されたコードによるものと，アプリ本来の仕様で行われるものの判断が難しい．そのため，今回は，機密情報が外部に送信されるものについては，全て，危険性があると判断する．

4. 提案手法

4.1. 提案手法概要

Android における Cordova アプリに，データフロー解析を適用し、機密情報の漏洩の可能性を判断する手法を提案する．提案手法の流れを(1)～(4)に示し，全体像を図 1 に示す．

(1) apk ファイルの解凍
(2) 使用プラグイン情報の抽出
(3) 機密情報プラグイン情報の抽出
(4) データフロー解析の実行

「A repackaging attack defense method focusing on plugins in Cordova hybrid applications」
† Hirofumi Akimoto・Waseda University
‡ Tomoji Kishi・Waseda University

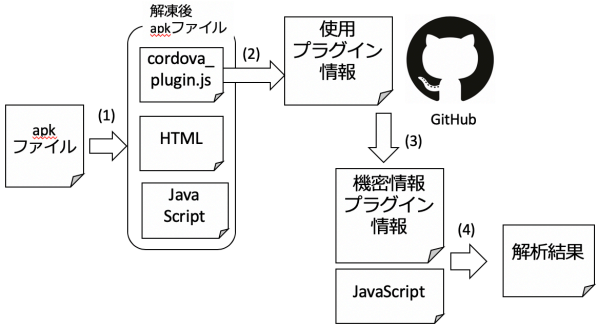

図 1. 提案手法の全体像

4.2. (1) apk ファイルの解凍

Android の Cordova アプリは，拡張子が apk のファイル形式(以下，apk ファイル)で配布される．apk ファイルは，ZIP 形式で圧縮されているため，解凍が可能である．解凍により，Cordova アプリで使用されている html ファイルや，JavaScript ファイル等を取り出すことが可能である．

4.3. (2)使用プラグイン情報の抽出

apk ファイル含まれる cordova_plugin.js より，Cordova アプリ内で使用されているプラグイン(以下，使用プラグイン)の情報を抽出する．

4.4. (3)機密情報プラグイン情報の抽出

(2)の手順で抽出した使用プラグイン情報の中から，機密情報プラグインの情報を抽出する．

プラグインは Java ファイルでの実装であり，コンパイル時に dex ファイルに変換されるため，コンパイル後のプラグインから情報を抽出することは難しい．本手法では，GitHub 上に存在するコンパイル前のプラグインより情報を抽出し，どの使用プラグインが機密情報プラグインであるかを判別する．その後，機密情報プラグインに関して，(2)で得た使用プラグイン情報の中から，JavaScript ソースコード中でプラグインを使用する際のオブジェクト名を抽出する．

もし，機密情報プラグインが存在しない場合は，情報漏洩の可能性がないとし，解析を終了する．

4.5. (4)データフロー解析の実行

(3)で抽出した機密情報プラグインの情報を基に，JavaScript ソースコードに対してデータフロー解析を実行する．まず，パーサーを利用して，JavaScript ソースコードを Abstract Syntax Tree(以下，AST)に変換する．AST 内で，(3)で得たオブジェクト名が使用されている箇所を探す．その後，AST よりオブジェクト，もしくはそのオブジェクトより生成されるデータの伝播をたどる．その伝播によって，機密情報の漏洩の可能性の有無を解析結果として出力する．情報漏洩が行われるかの判断については，ソースコード中に，XMLHttpRequest 等の特定のサーバーにプログラム中のデータの送信を行う可能性のあるメソッドやタグが，機密情報に対して使用されているか否かとする．

5. 評価実験

5.1. 実験概要，目的

4 で提案した手法を用いて，Google play で配布されている Cordova アプリを対象とした機密情報の漏洩の可能性の判断を行った．下記 3 パターンの Cordova アプリを用意し，本提案手法によって実際に機密情報の漏洩の可能性の判断が可能かであるか，機密情報漏洩のないものについて，誤った判断が行われないかについて確かめる．

① 機密情報プラグインが使われていないもの
② 機密情報プラグインが使われているが，外部への送信メソッドやタグがないもの
③ 機密情報プラグインが使われていて，かつ自身で機密情報を外部に送信するようにコードを書き換えたもの

5.2. 実験結果

①，②については，機密情報の漏洩の可能性は判断されず，Cordova アプリに危険性はないとの結果が出力された．③については，機密情報漏洩の可能性が判断された．この結果から，プラグインを利用した機密情報の漏洩の可能性の判断において，本手法は妥当であると考えられる．

6. 結論と今後の課題

本研究では，Cordova アプリにおける，プラグインの不正利用による機密情報の漏洩を目的とした，リパッケージング攻撃の防御手法を提案した．提案手法を用いることで，リパッケージング攻撃が行われた Cordova アプリおける情報漏洩の可能性の判断が可能となった．

今後の展望としては，機密情報の外部への送信がアプリの本来の仕様である場合の判断が挙げられる．また，ハイブリッドアプリでは JavaScript ライブラリの jQuery を使用できるため，こういったライブラリの使用による記述への対応も課題として挙げられる．

参考文献

[1] T. Takeuchi, K. Mouri, S. Saito: Mocha: Automatically Applying Content Security Policy to HTML Hybrid Application on Android Device, 2017 Fifth International Symposium on Computing and Networking (CANDAR), pp.503-509, 2017.

[2] N. Kudo, T. Yamauchi, T. H. Austin: Access control for plugins in Cordova-based hybrid applications, Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, no. 2, pp. 1063-1069, 2017.