2W-03

AndroidにおけるWebViewのWebアクセス観測機構の提案

今村 祐太† 上川 先之‡ 工藤 直樹‡ 佐藤 将也‡ 山内 利宏‡
†岡山大学工学部 ‡岡山大学大学院自然科学研究科

1 はじめに

Android端末の普及に伴い，Android端末を標的とするマルウェアが増加している．Android端末にマルウェアを感染させる攻撃手法として，攻撃者は悪性なWebコンテンツを利用する場合がある．また，Androidアプリケーション（以降，アプリ）にWebコンテンツへのアクセスを提供するWebViewが多くのアプリで利用されている．WebViewの利用により，アプリの開発者は，アプリ内にWebコンテンツを表示できる．しかし，Webブラウザへの切り替えが行われないため，利用者はWebアクセスが行われていることに気付きにくい．また，WebViewを対象としたWebアクセス観測機構が存在しないため，WebViewを利用したWebアクセスが安全か否か検証できない．

そこで，本稿では，AndroidにおけるWebViewのWebアクセス観測機構を提案し，その基本機構について述べる．提案機構は，WebViewの改変により，WebViewを利用したWebアクセスを観測できる．

2 Androidへの Webを経由した攻撃

Android端末は，悪性なWebコンテンツへのWebアクセスによりマルウェアに感染する場合がある[1]．ここで，悪性なWebコンテンツを利用した攻撃として，Drive-by Download攻撃（以降，DbD攻撃）が存在する．DbD攻撃とは，攻撃サイトへアクセスした利用者のアプリケーションの脆弱性を悪用し，攻撃コードの実行により，利用者の意図に関わらずマルウェアをダウンロードさせる攻撃手法である．文献[1]では，AndroidにおけるDbD攻撃を利用するマルウェアとして，Jifake，GGTracker，Spitmo，およびZitMoが挙げられている．

3 WebView

3.1 概要

WebViewとは，Webブラウザに切り替えることなく，アプリ内にWebコンテンツを表示するAndroidのコンポーネントである．WebViewの利用により，アプリの開発者は，利便性の高いアプリを容易に開発できる．このため，WebViewは多くのアプリで利用されている．例えば，Google Playの10種類の各カテゴリにおいて，ダウンロード数上位20個のアプリのうち，86%においてWebViewが利用されている[2]．

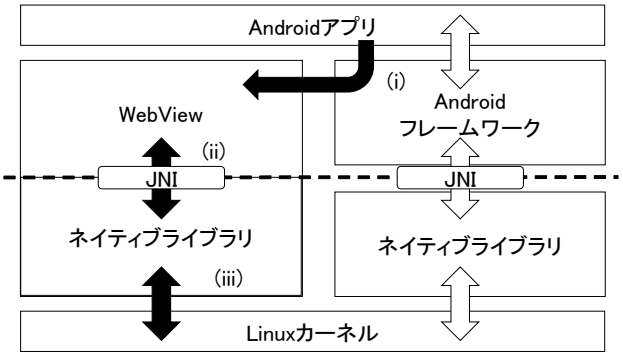

図1 WebViewを利用したWebアクセスの処理流れ

3.2 不正なWebアクセスの可能性

WebViewはWebブラウザに切り替えることなくWebコンテンツを表示するため，利用者はWebアクセスが行われていることに気付きにくい．また，WebViewにおいても悪性なWebコンテンツによる攻撃の可能性がある．そこで，WebViewの利用による意図しないWebアクセスが安全か否か検証するために，通信内容を把握することが必要である．

4 WebViewのWebアクセス観測機構

4.1 基本機構

3.2節で述べた要求に対処するため，WebViewを利用したWebアクセスの通信内容を把握する手段として，WebViewのWebアクセス観測機構（以降，提案機構）を提案する．提案機構はWebViewを利用したWebアクセスを観測し，通信内容を把握することを目的とする．

図1にWebViewを利用したWebアクセスの処理流れを示す．WebViewを利用するアプリがWebアクセスを行う場合，図1に示した(i)，(ii)，(iii)の順序で処理を行う．また，WebViewを利用したWebアクセスは，システムコールを利用する．このため，WebViewのシステムコール処理後にWebアクセスを観測する機能を追加することで提案機構を実現する．提案機構の利点として，WebViewの置き換えのみで，WebViewを利用したWebアクセスを観測可能であることが挙げられる．

Proposal of Web Access Monitoring Mechanism for WebView on Android
Yuta Imamura†, Hiroyuki Uekawa‡, Naoki Kudo‡, Masaya Sato‡, Toshihiro Yamauchi‡
†Faculty of Engineering, Okayama University
‡Graduate School of Natural Science and Technology, Okayama University

4.2 課題

提案機構を実現するには，以下の 2 つの課題に対処する必要がある．

(課題 1) 取得する情報の検討
WebView を利用した Web アクセスが安全か否か検証するために，提案機構が取得すべき情報について検討する．

(課題 2) 提案機構の追加箇所
(課題 1) で検討した情報を漏れなく観測可能な提案機構の追加箇所について検討する．

4.3 取得する情報

4.2 節で述べた (課題 1) に対処するため，以下の情報を取得する．

(1) HTTP リクエストと HTTP レスポンス
WebView においても，Web アクセスを行う際，HTTP を利用する．このため，WebView を利用した Web アクセスの通信内容を把握するために，HTTP リクエストと HTTP レスポンスの情報を取得する．

(2) 端末の環境情報
DbD 攻撃は，OS やブラウザ，プラグインの種類とバージョンなどの端末の環境情報を判別し，条件を満たす場合のみ，攻撃サイトに利用者を誘導する．Android 端末に脆弱性が存在する場合，攻撃者がその脆弱性を悪用し攻撃を行う可能性がある．そこで，端末の環境情報を取得する．

(3) 通信先の IP アドレスの情報
DbD 攻撃に用いられるドメインや IP アドレスが有効である期間は短い．このため，(1) で取得した情報を基に解析を行った場合，悪性な Web サイト（以降，攻撃サイト）を特定することは困難である．このため，通信先の IP アドレスの情報を取得する．

(4) ソケット接続の際のコネクションエラーの情報
攻撃サイトが存在する期間は短いため，攻撃サイトへ Web アクセスする場合，コネクションエラーが生じる可能性がある．そこで，ソケット接続の際のコネクションエラーの情報も取得する．これにより，攻撃サイトへ誘導する改ざんされた Web コンテンツを特定できる可能性を高くできる．

4.4 提案機構の追加箇所の検討

4.3 節で述べた情報を取得するためには，WebView に提案機構を追加する必要がある．ここで，図 1 を用いて，WebView を利用した Web アクセスの処理流れについて述べる．

(i) WebView の API の呼び出し
WebView の API は Android フレームワークに存在しており，その実装は Android フレームワークから分離したコンポーネントとして実現している．このため，WebView を利用する Android アプリは，Android フレームワークを経由し，WebView における Java メソッドを呼び出す．ここで，アプリは表示する Web コンテンツの URL を WebView に渡す．

(ii) WebView における C++側のメソッドの呼び出し
WebView における Java メソッドは Java Native Interface (JNI) を利用し，WebView における C++（ネイティブ）側のメソッドを呼び出す．ここで，WebView における C++側のメソッドが Web アクセスを行うための HTTP リクエストを作成する．

(iii) システムコールの利用
WebView における C++側のメソッドがシステムコールを発行し，Web アクセスを行う．

上記のように，WebView は，WebView の C++側のメソッドが HTTP リクエストを作成し，システムコールを発行することで Web アクセスを行う．このため，WebView の C++側のメソッドによるシステムコール発行後に 4.3 節で述べた情報を取得する．これにより，WebView を利用した全ての Web アクセスを観測できる．

5 おわりに

本稿では，WebView の Web アクセス観測機構の基本機構，課題，および対処について述べた．残された課題として，提案機構の実装と評価がある．

謝辞 本研究成果は，国立研究開発法人情報通信研究機構 (NICT) の委託研究「Web 媒介型攻撃対策技術の実用化に向けた研究開発」により得られたものです．

参考文献

[1] Zhou, Y. and Jiang, X.: Dissecting Android Malware: Characterization and Evolution, Proceedings of 2012 IEEE Symposium on Security and Privacy (SP 2012), pp. 95–109 (2012).

[2] Luo, T., Hao, H., Du, W., Wang, Y., and Yin, H.: Attacks on WebView in the Android System, Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC 2011), pp. 343–352 (2011).