

Androidセキュリティ勉強会
～WebViewの脆弱性編～

2012年10月6日

goroh_kun

Agenda

- セキュアコーディングTF紹介
- WebViewの脆弱性
 - 概要
 - AndroidのWebViewどこがまずいの？
 - JavaScript編
 - fileスキーマ編

Goroh

- 本名とは関係ありません
- 主にtwitterで活動中 (@goroh_kun)
 - 画像は良くいく店のマスコット
- Android周りでは端末のroot化手法を公開する人として認識されてるかも
- 最近 @sutegoma2のメンバーになりました
- 来週からマレーシアに行ってきます
 - Hack In the Box マレーシア
- 最近セキュリティの会社に就職しました



- Androidセキュリティ部所属
- JSSEC(日本スマートフォンセキュリティ協会)脆弱性WGリーダー
- 日本で販売されているAndroid端末のほぼすべてのroot取得方法・NANDアンロック方法を解析・公開している
- シャープさんの端末で言うと
 - Au IS01,IS03,IS05,IS11SH,IS12SH,IS13SH,IS14SH, ISW16SH
 - Docomo SH-10B, SH-03C, SH-12C, SH-01D, SH-09D
 - Softbank 003SH, 005SH, 007SH, 009SH, 106SH
- 富士通東芝さんだと
 - F-07C,F-01D, F-05D, ISW13F, F-10D
- パナソニックさんだと(tomoyo Linux)
 - P-04D, P-05D, P-06D

WebViewの脆弱性

まずは概要から

Androidセキュリティ勉強会

- JVNのデータベースに乗る脆弱性の数々
- JPCERTに報告して、対象企業が脆弱性を修正すると公開されます

JVN#25731073 **複数のクックパッド製 Android アプリケーションにおける WebView クラスに関する脆弱性**

JVN#00000601 **TwitRocker2 (Android 版) における WebView クラスに関する脆弱性**

JVN#90751882 **Dolphin Browser における WebView クラスに関する脆弱性**

JVN#46088915 **Yahoo!ブラウザ における WebView クラスに関する脆弱性**

JVN#88643450 **Sleipnir Mobile for Android における WebView クラスに関する脆弱性**

JVN#99192898 **複数の GREE 製 Android アプリにおける WebView クラスに関する脆弱性**

JVN#77393797 **サイボウズLive for Android における WebView クラスに関する脆弱性**

JVN#59652356 **サイボウズ KUNAI for Android における WebView クラスに関する脆弱性**

JVN#23568423 **サイボウズ KUNAI for Android において任意の Java のメソッドが実行される脆弱性**

JVN#03015214 **KUNAI Browser for Remote Service β における WebView クラスに関する脆弱性**

JVN#86318665 **Android 版 jigbrowser+ における WebView クラスに関する脆弱性**

- 実は氷山の一角なのです。
- 切りがないのでわざわざ報告しないのです。
- 大本の作りを何とかするしかないのです。
- JPCERTに報告しても、そもそも修正が終わらないとデータベースに載らないのです。

今日話すこと

- ほとんどは既にネットワークで公開され、気を付けましょうと言われていることなのです。
- この記事は必読
 - 熊谷 裕志 (JPCERTコーディネーションセンター)
 - スマートフォンアプリへのブラウザ機能の実装に潜む危険、WebViewクラスの問題について
 - <http://codezine.jp/article/detail/6618>
- 今日話すことはほぼ、ここに関することです。上の記事では気を付けましょうね～ぐらいのレベルで書いてますが、。実はこれ、このままWebView使うのどうよ？ぐらいの話でもあります。

- 何がそもそも問題なのか？
- よくJVNで書かれる脆弱性情報
 - ユーザが、不正な他の Android アプリケーションを使用した場合、当該製品のデータ領域にある情報が漏えいする可能性があります。
 - 任意のメソッドを実行される恐れがあります。
- これじゃわからないですよね・・・。
- よっぽど迂闊な人でないと、こういうことにならないんじゃないの？
 - そんなことはありません。誰でもこの脆弱性を作りこむ可能性があります。
 - むしろWebViewを使ってサーバーと連解するアプリを作った場合、作りこむ可能性が高いです。

- 今日覚えてほしいキーワード
 - WebViewキャッシュ
 - addJavaScriptInterface
 - fileスキーマ

WebViewキャッシュ

- ブラウザでブラウズするときにフォームのデータなどを保存するデータベース
- 一般の人は、パスワードを保存しますか？で特に考えなしに「はい」を押してしまよね。
- 聞かれないでも勝手に保存するものもある。(Cookie、アクセス履歴)

WebViewキャッシュのAndroidでの問題点

- WebViewキャッシュの保存場所が一意に決まってしまうため、攻撃がしやすい点
 - /data/data/パッケージ名/databases/webview.db
 - デモします(皆さんも見てみてくださいね~)
- データベース自体はsqliteというデータベースファイルであり、バイナリデータなのだが、パスワードやID部分は特に暗号化されておらず、WebView上で目視ができる。
- WebViewキャッシュ自体は他アプリからの読出しに関してはパーミッションで保護されているが、WebView自身からの攻撃は保護できない

fileスキーマ

- WebViewにURLを指定するときに[file://が利用できます](#)。
- 端末ローカルのファイルが見れます。
- ファイルのパーミッション的に読めるものは基本いけます
- WebViewとして表示できなかったとしても、JavaScriptを利用して表示させずにデータをブラウザ内にロードできます。(それをサーバーにあげることも..)

WebViewキャッシュの問題点

- ということが起きるのか
 - 端末を借りたら・・・ブラウザから簡単にWebViewキャッシュが目しできます。目grepでパスワード&ID入手！
 - 標準ブラウザの問題点と組み合わせることでサーバーにWebViewキャッシュをアップロード
 - 家電量販店に並んでいるブラウザでID&パスワード記録させちゃ駄目ですよ。
- IPAさんの言う内部のデータ、というのは実はアカウント情報も含んでいた、ということなんですね・・・。

デモします

- 動画撮影はやめたほうがよさそう
- ブラウザから以下を入力すればOKですよ
- Android3.x以下の人
 - <file:///data/data/com.android.browser/databases/webview.db>
- Android4.0(ICS)以上の人
 - <file:///data/data/com.google.android.browser/databases/webview.db>
- ブラウザを愛している人ほど驚きの結果に・・・

- Android標準ブラウザの問題
- 実はPCの世界でもfile://で内容が見えるなんて当たり前の世界
- Androidに限って何が問題？
 - WebViewキャッシュのパスが決まっちゃってること？
- 他にもあるんです

- Android標準ブラウザの問題
- ファイルのダウンロードの自動開始問題
 - リンクを踏んで瞬間にブラウザからファイルをダウンロードさせることができます。
 - ダウンロード時にユーザーに確認はなし(最近のPC向けブラウザなら確認ありますよね)
- ダウンロードされるファイルの場所が大体決まってる
 - /sdcard/Download/のした。
 - ファイル名はそのまま
 - 攻撃者はファイル名がかぶらないようにランダムなファイル名をダウンロード時に生成してくる
- ダウンロードの終わるタイミングを推測して、ダウンロード済みのファイルに自動遷移させることが可能
 - JavaScriptにより、document.locationを変更するだけ

- Android標準ブラウザの問題
- ダウンロード済みのファイルでもJavaScriptが有効
- その際、ユーザーさんへの確認はなし(最近のPC向けブラウザなら確認ありますよね)
- <file:///sdcard/Download/hogehoge.html>からい、JavaScriptを実行した場合、端末内のfileスキームでアクセスできるファイルは全部読める
- つまり、、WebViewも・・・。
- 読み取ったファイルはPOSTなり、GETなりでサーバーにアップロード可能
- アンドロイドのこういった仕様の積み重ねで、ワンクリックでWebViewキャッシュをアップロードするまでのコンボ攻撃が完成している

- Android標準ブラウザの問題
- ダウンロード済みのファイルでもJavaScriptが有効
- その際、ユーザーさんへの確認はなし(最近のPC向けブラウザなら確認ありますよね)
- <file:///sdcard/Download/hogehoge.html>からい、JavaScriptを実行した場合、端末内のfileスキームでアクセスできるファイルは全部読める
- つまり、、WebViewも・・・。
- 読み取ったファイルはPOSTなり、GETなりでサーバーにアップロード可能
- アンドロイドのこういった仕様の積み重ねで、ワンクリックでユーザーに許諾なしに、WebViewキャッシュをアップロードするまでのコンボ攻撃が完成してしまう

- 対策ある？
- fileスキームは本当に必要？いらないなら使えないように
- fileスキームを使う場合は必ずパスをチェックしよう
 - 相対パスを利用した抜け穴を作らないためにも利用するAPIには注意
 - 詳細はAndroidセキュアコーディングガイドライン等を見てください
 - 要望しだいでは別途機会を設けます
- JavaScriptも必要ないなら有効にしないこと
- WebViewキャッシュは・・・
 - こまめに削除するとか・・・
 - いらないサイトにアクセスしないとか・・・
 - 標準ブラウザを使わないとか・・・

- addJavaScriptInterfaceの話
- これは何か？
- こんな使い方してますよね
 - スマートフォンアプリへのブラウザ機能の実装に潜む危険、WebViewクラスの問題について
 - <http://codezine.jp/article/detail/6618>
- リフレクションが使えてしまう問題
- なんだ、開発者が気をつければすむ話じゃないの？
- 実はかなり気を付けて使っていても問題があります
- 信用の置けるサイトを開く以外で利用したら何が起こるかわからない代物なのです

Androidセキュリティ勉強会

- 大事なデータをJavascriptInterfaceに使うオブジェクトに持たせなければよいのではないの？
- サイトの情報では、Contextを持たせると脆弱性につながります、と解説
- Contextがとられなければ大丈夫？
 - 現状Contextがリフレクションで取れてしまいます。
 - 解説します。
- リフレクションが使えるということは、クラスのstaticなメンバーは誰からも参照できるんです。
- ここ見てください。JniUtil.javaのソースコード
 - http://tools.oesf.biz/android-4.0.4_r1.0/xref/frameworks/base/core/java/android/webkit/JniUtil.java
- あ、、sContext。でも初期化されてなければ良いんじゃない？
- WebViewの初期化時に必ずセットされる素敵仕様
 - http://tools.oesf.biz/android-4.0.4_r1.0/xref/frameworks/base/core/java/android/webkit/WebView.java#1084

- つまり、setJavascriptInterfaceを使うオブジェクトは全部Contextが取得可能ってこと？
 - その通りです。悪意のあるJavaScript経由で
 - Intent発行できます
 - アプリ一覧取得できます
 - Androidのさまざまなサービスにアクセスできる！



- でもそれだけじゃないんだ。
- もっと恐ろしいことに。
- java.lang.Runtimeが使える！
- Runtimeが取れたらどうなるの？
 - execメソッド: 任意のプログラムをWebViewを実装したアプリのユーザー権限で実行できます
 - loadメソッド: 任意のJNIライブラリをWebViewを実装したアプリのユーザー権限で実行できます。アプリそのものに寄生・改造することも可能。
 - これって、ウィルスじゃ..

- とりあえずデモします
- Javaのソースコードに注目
- こんな、、アンチウィルスアプリでなんともならん
- Googleのマーケットで不正監視するアプリ(Bouncer)でも何とかなるのか？

- 対策

- addJavascriptInterfaceは本当に使わないでください。
- 安心できるサイトしかアクセスしないから大丈夫??
- いやいや、偽装APや偽装DNSを使えば..
- https使えば大丈夫?
 - そうかもね。ちゃんと証明書の確認はしないと駄目だけどね
- 本来はどうするべき??
 - Googleさんに早くなおしてもらえるように、理想的な動きはどうあるべきか話し合っって提案していきましょう

- 最後に宣伝させてください
- イエラエセキュリティというところでアプリ診断担当しています。
- <http://www.ierae.co.jp/>
- WebViewを使ったアプリ優先で、キャンペーン価格で診断しています。
- 無料アプリの場合も相談可能
- お気軽にお問い合わせください