Electronのセキュリティその後

Yosuke HASEGAWA @hasegawayosuke

自己紹介

はせがわようすけ @hasegawayosuke
セキュリティのほうからやってきました。

- OWASP Kansai Local Chapter Leader
- OWASP Japan board member
- (株)セキュアスカイ・テクノロジー 技術顧問
- http://utf-8.jp/ author of aaencode / jjencode


提供
Shibuya.XSS
UTF-8.jp
エレクトロンで安全なアプリを
作るための挑戦が今はじまる!!

宣伝　(*'ω'*)

宣伝: gihyo.jpで連載始めました

- JavaScriptセキュリティの基礎知識
  - http://gihyo.jp/dev/serial/01/javascript-security

前回までのあらすじ

前回までのあらすじ

- DOM-based XSSがあると
  - node機能を完全にオフにできない
    - webviewタグやwindow.openで再活性化
  - 攻撃者によってnode機能が利用可能
  - 任意コードの実行が可能
    - ファイル読み書き、プロセス生成、通信etc

前回までのあらすじ

- 詳しい話はこのあたりを見てください
  - http://utf-8.jp/public/2016/0307/electron.pdf

その後の改善

その後の改善

- 親のnode-integrationを継承するようになった

```
mainWindow = new BrowserWindow( {
    width: 600,
    height: 400,
    webPreferences: { nodeIntegration : false }
} );
```

```
<webview nodeintegration src="data:text/html,
    // 以前はnode機能を強制的に有効にできた
    <script>
        require('child_process').exec('calc.exe', ()=>{})
    </script>
"></webview>
```

```
window.open('http://example.jp/', '', 'nodeIntegration=1');
// 以前はnode機能を強制的に有効にできた
```

その後の改善

- 親のnode-integrationを継承するようになった
  - Disable node on child window when disabled on parent
    https://github.com/electron/electron/pull/4897

- 現状、これでnode機能が再活性化されることはなさそう。(*'ω'*)

node機能を使いたい場合

- レンダラ内からもnode機能を使いたい場合
  - IPC経由でメイン側に丸投げ
  - preloadする

```
mainWindow = new BrowserWindow( {
    webPreferences: { preload : "./preload.js" }
} );
```

```
//preload.js
global.execCalc = function(){
    require('child_process').exec('calc.exe', ()=>{})
};
```

```
<webview src=data:text/html,
    <script>
        execCalc(); // 呼び出し可能
    </script>
"></webview>
```

preload機能 [2016-07-06]

- ドキュメントによればpreload機能は前ページのコードで動作するはずだが、2016-07-06現在のv1.2.5ではpreload機能が動作していない。
- そのうち直るかもなので気長に待つ(；´Д`)

- https://github.com/electron/electron/blob/master/docs/api/browser-window.md#new-browserwindowoptions

その他の懸念

shell.openExternal

- 外部ブラウザを開くために使われる

```
const {shell} = require('electron');
shell.openExternal('http://example.com/');
```

- shell.openExterenalは外部プロセスを起動可

```
shell.openExternal('c:¥¥windows¥¥notepad.exe');
```

  - shell.openExternalへは/^https?:¥/¥//のみ渡す

サードパーティの広告

- webview内に3rd partyの広告が入る場合
  - 広告JSもwebview内で自由にコード実行
  - webviewでnodeが有効な場合はnode機能も使える
  - nodeが無効な場合でも全画面書き換え、偽ログイン画面の表示など
  - 悪意ある広告、配信サーバの汚染など…
- iframe sandboxでtopへの干渉を防ぐ

```
<iframe sandbox="allow-scripts" src="3rdparty-ads">
```

  - JS埋め込み型の広告だと対策できない…

まとめ

- Electronのセキュリティ
  - レンダラ内でnodeが無効な場合はかなり安全
  - node機能を使いたければIPC or preload
- shell.openExternal
  - 外部リンクがhttp(s)か確認する
- 悪意ある攻撃への懸念
  - 広告などを含める場合は<iframe sandbox>でtopへの干渉を防止
  - JSの広告は諦めるしかない

提供

Shibuya.XSS
UTF-8.jp

Question?

hasegawa@utf-8.jp

@hasegawayosuke

http://utf-8.jp/