今から始めるHTML5セキュリティ

一般社団法人JPCERTコーディネーションセンター
早期警戒グループ 情報セキュリティアナリスト
重森 友行

JPCERT/CCとは

一般社団法人JPCERTコーディネーションセンター

（JPCERT/CC (ジェーピーサート・コーディネーションセンター)）

Japan Computer Emergency Response Team Coordination Center

- https://www.jpcert.or.jp/
- サービス対象: 日本国内のインターネット利用者やセキュリティ管理担当者、ソフトウエア製品開発者等（主に、情報セキュリティ担当者）
- コンピュータセキュリティインシデントへの対応、国内外にセンサをおいたインターネット定点観測、ソフトウエアや情報システム・制御システム機器等の脆弱性への対応などを通じ、セキュリティ向上を推進
- インシデント対応をはじめとする、国際連携が必要なオペレーションや情報連携に関する、我が国の窓口となる　CSIRT

※各国に同様の窓口となる　CSIRTが存在する

(例えば、米国のUS-CERT 、中国のCNCERT, 韓国のKrCERT/CC、など)

- 経済産業省からの委託事業として、コンピュータセキュリティ早期警戒体制構築運用事業を実施

- JPCERT/CCをご存知ですか? -

JPCERT/CCの活動

インシデント予防　インシデントの予測と捕捉　発生したインシデントへの対応

脆弱性情報ハンドリング

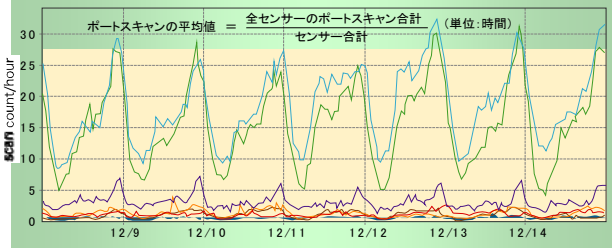
- 未公開の脆弱性関連情報を製品開発者へ提供し、対応依頼
- 関係機関と連携し、国際的に情報公開日を調整
- セキュアなコーディング手法の普及
- 制御システムに関する脆弱性関連情報の適切な流通

JVN Japan Vulnerability Notes

情報収集・分析・発信

定点観測（TSUBAME）

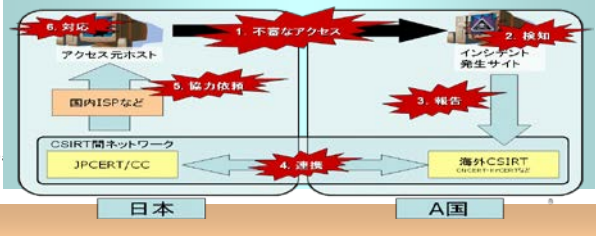
- ネットワークトラフィック情報の収集分析
- セキュリティ上の脅威情報の収集、分析、必要とする組織への提供



インシデントハンドリング

（インシデント対応 *調整*支援）

- マルウエアの接続先等の攻撃関連サイト等の閉鎖等による被害最小化
- 攻撃手法の分析支援による被害可能性の確認、拡散抑止
- 再発防止に向けた関係各関の情報交換及び情報共有



早期警戒情報

重要インフラ、重要情報インフラ事業者等の特定組織向け情報発信

CSIRT構築支援

海外のNational-CSIRTや企業内のセキュリティ対応組織の構築・運用支援

アーティファクト分析

マルウエア（不正プログラム）等の攻撃手法の分析、解析

国際連携

各種業務を円滑に行うための海外関係機関との連携

本日の内容

- HTML5の概要
  - HTML5とは
  - HTML5とセキュリティ
- 「HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書」の紹介
- 報告書の内容の紹介
  - JavaScript API
    - XMLHttpRequest
    - OfflineWebApplication
  - HTML新要素・属性
  - セキュリティ関連機能

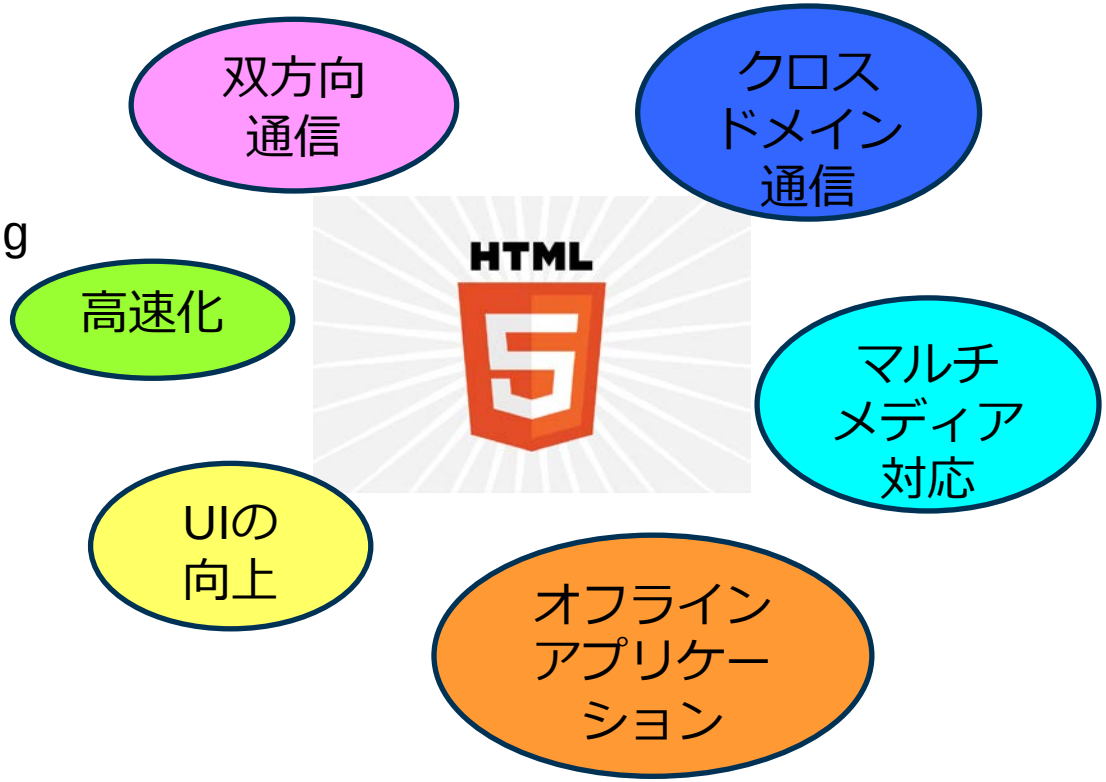
HTML5とは

- 従来のHTMLに代わる次世代のHTML
- ブラウザでのデータ格納、クライアントとサーバ間での双方向通信、位置情報の取得など、従来のHTMLよりも柔軟かつ利便性の高いWeb サイトの構築が可能となる
- 日本を含むアジア太平洋地域においても急速に普及が進みつつある
- 最近のブラウザの多くは、HTML5に対応している（一部実装されていない機能もある）

今回はHTML5だけでなく、HTML5を使う上で使用することの多い関連する機能なども併せて紹介

HTML5とセキュリティ

- HTML5は開発者にとって非常に便利
- 様々な機能により表現の幅が大きく広がる
- 関連キーワード
  - XMLHttpRequest
  - Cross Document Messaging
  - Offline Web Application
  - Web Storage
  - WebSocket
  - Web Workers
  - 新規追加HTML属性・要素



HTML5とセキュリティ

- HTML5は開発者にとって非常に便利

⇒ 攻撃者にとっても非常に便利

- 様々な機能により表現の幅が大きく広がる

⇒ 攻撃の幅も大きく広がる

- 関連キーワード
  - XMLHttpRequest
  - Cross Document Messaging
  - Offline Web Application
  - Web Storage
  - WebSocket
  - Web Workers
  - 新規追加HTML属性・要素

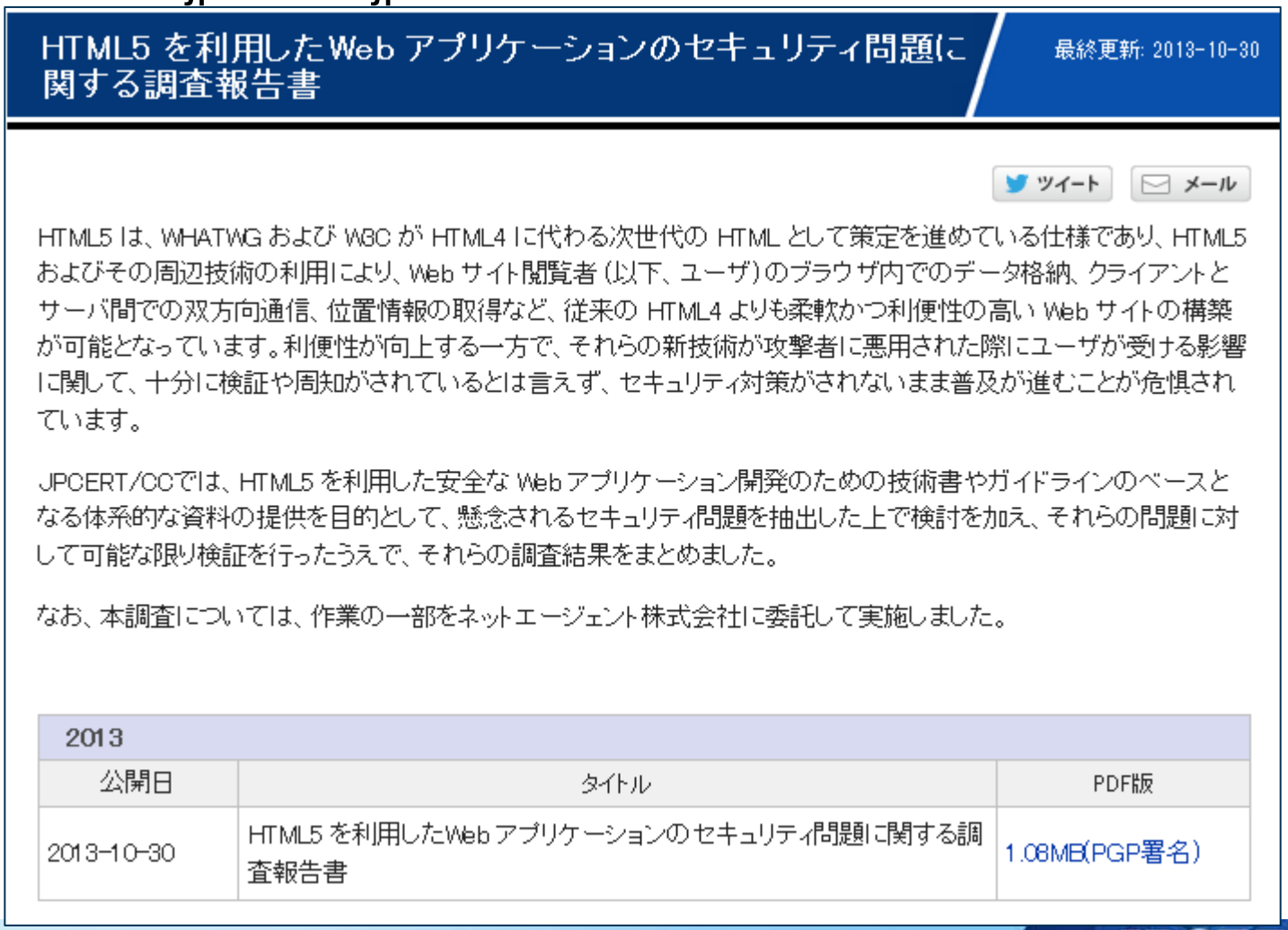
HTML5とセキュリティ

- 従来のHTMLでは影響のなかったケースが、ブラウザのHTML5対応により、脆弱性となってしまうケースが存在する

- 利便性が向上する一方で、それらの新技術が攻撃者に悪用された際にユーザが受ける影響に関して、十分に検証や周知がされているとは言えない

HTML5 を利用したWeb アプリケーションの セキュリティ問題に関する調査報告書

- 2013年10月30日に公開
- https://www.jpcert.or.jp/research/html5.html



HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書

最終更新: 2013-10-30

ツイート　メール

HTML5 は、WHATWG および W3C が HTML4 に代わる次世代の HTML として策定を進めている仕様であり、HTML5 およびその周辺技術の利用により、Web サイト閲覧者（以下、ユーザ）のブラウザ内でのデータ格納、クライアントとサーバ間での双方向通信、位置情報の取得など、従来の HTML4 よりも柔軟かつ利便性の高い Web サイトの構築が可能となっています。利便性が向上する一方で、それらの新技術が攻撃者に悪用された際にユーザが受ける影響に関して、十分に検証や周知がされているとは言えず、セキュリティ対策がされないまま普及が進むことが危惧されています。

JPCERT/CCでは、HTML5 を利用した安全な Web アプリケーション開発のための技術書やガイドラインのベースとなる体系的な資料の提供を目的として、懸念されるセキュリティ問題を抽出した上で検討を加え、それらの問題に対して可能な限り検証を行ったうえで、それらの調査結果をまとめました。

なお、本調査については、作業の一部をネットエージェント株式会社に委託して実施しました。

2013

公開日	タイトル	PDF版
2013-10-30	HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書	1.08MB(PGP署名)

報告書で紹介している内容(機能別)

- JavaScript API
  - Cross Document Messaging
  - Web Storage
  - WebSocket
  - **Offline Web Application**
  - Web Workers
  - **XMLHttpRequest**
- HTML新要素・属性
  - a
  - **button**
  - **iframe**
  - meta
  - **video**
  - **audio**
  - canvas
  - **input**
  - source
- セキュリティ機能
  - X-XSS-Protection
  - X-Content-Type-Options
  - X-Frame-Optoins
  - Content-Security-Policy
  - Content-Disposition
  - Strict-Transport-Security

※本資料では、赤文字で記載している項目について説明する

報告書の使い方

- 技術書・ガイドラインのベース資料
- 仲間内の勉強会資料
- セミナの参考資料

などにどうぞ

引用・転載にあたっては以下を参照してください。

JPCERT/CC ご利用にあたってのお願い
https://www.jpcert.or.jp/guide.html

記載例）
引用元: JPCERTコーディネーションセンター
「HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書」
https://www.jpcert.or.jp/research/HTML5-20131030.pdf

報告書の内容を紹介

JavaScript API
HTML新要素・属性
セキュリティ関連機能

報告書の内容を紹介

JavaScript API

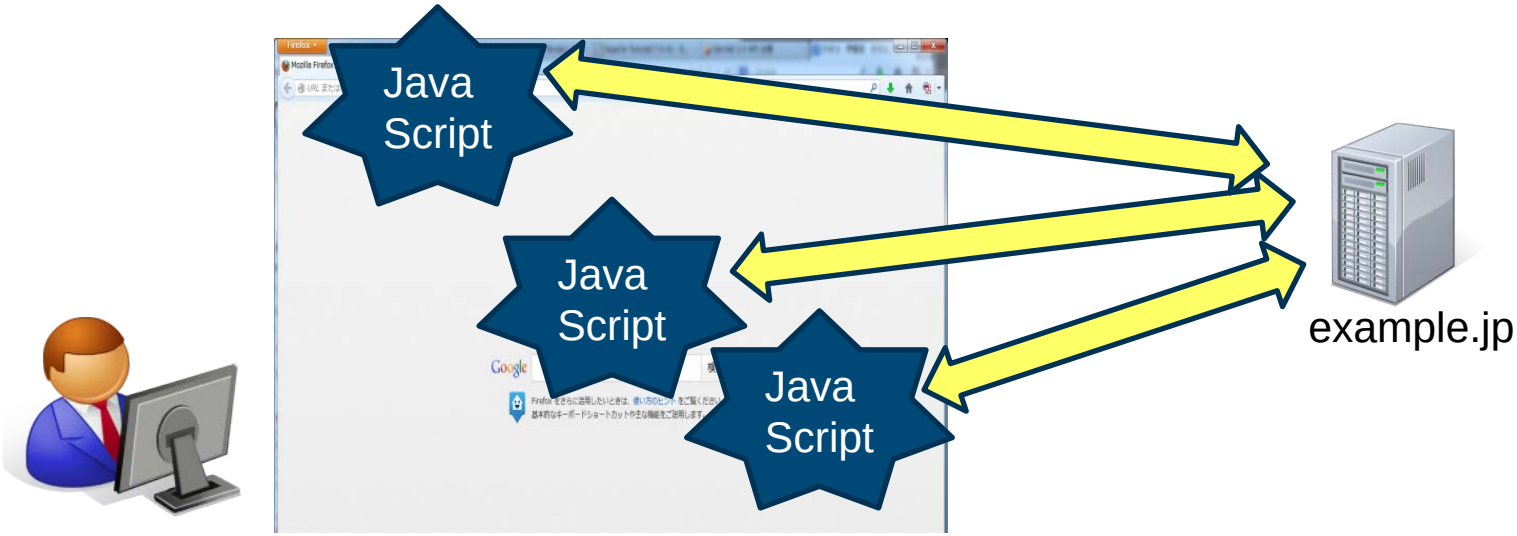
HTML新要素・属性

セキュリティ関連機能

JavaScript API
XMLHttpRequest

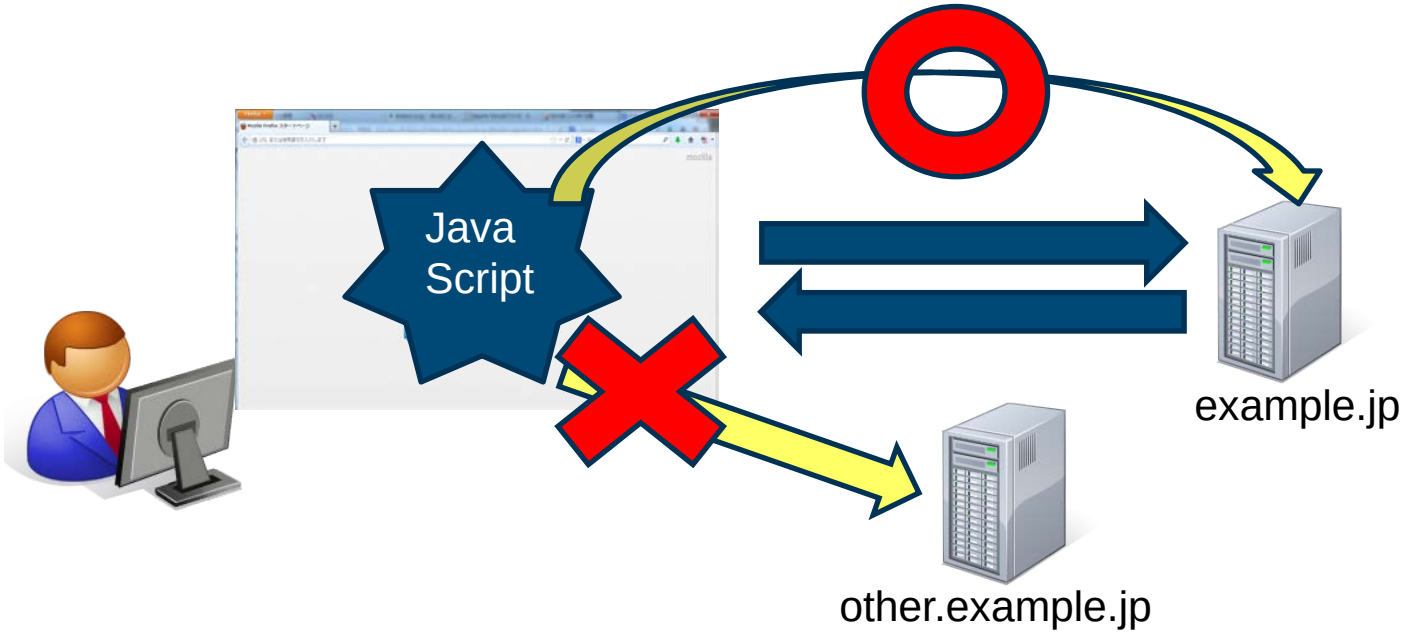
XMLHttpRequest(XHR)概要

- XMLHttpRequest(XHR)とは
  - JavaScriptでHTTP通信を行うためのAPI
  - 非同期通信によりインタラクティブな表現が可能
  - AJAXの普及に伴い使用される機会が増加
  - HTML5以前からある機能だが、HTML5で新しくなった



従来のXHR

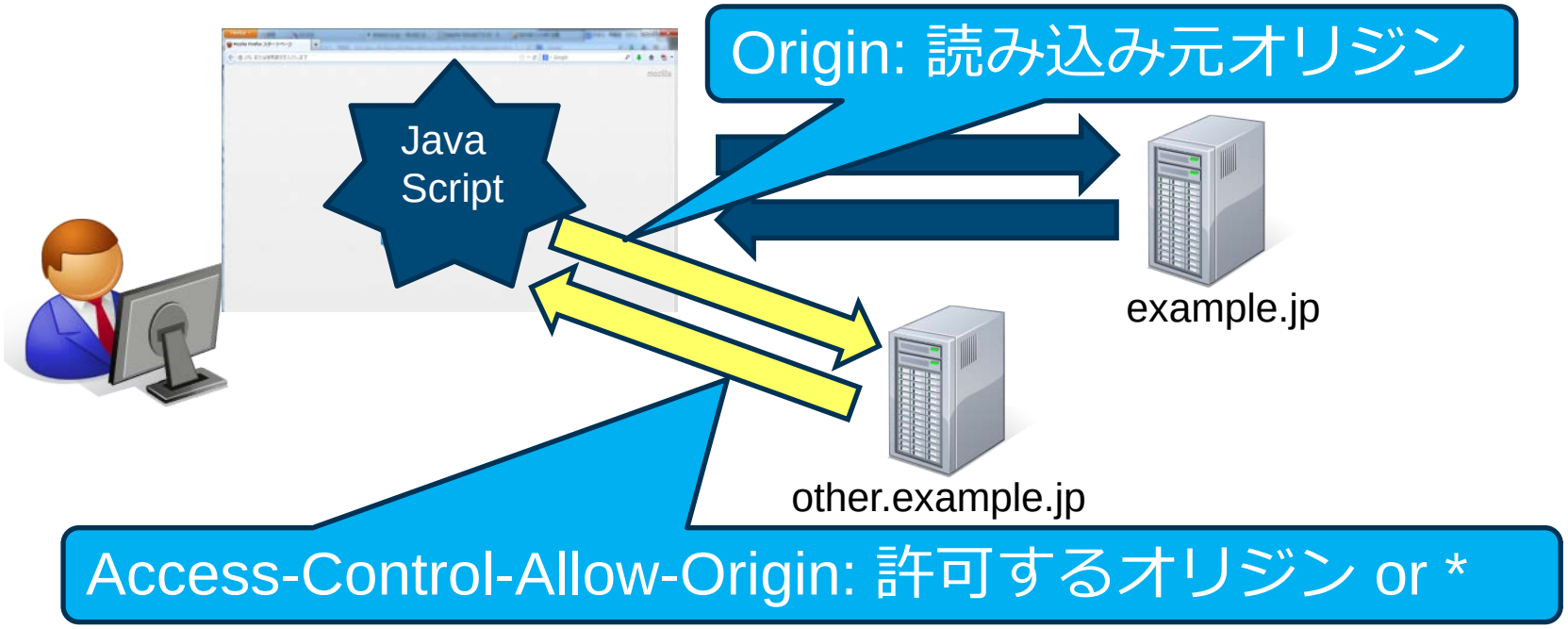
- JavaScriptを読み込んだオリジン(※)のみと通信が可能
- 別オリジンへの通信(クロスオリジン)では、リクエスト自体が発行されない



※オリジン： ホスト・ポート・スキームの組み合わせ

HTML5のXHR（XMLHttpRequest Level 2）

- クロスオリジンに対応
- クロスオリジンで通信するためには、サーバの合意が必要
- クライアントはサーバからのレスポンスを見てアクセスの可否を判断するため、サーバ側で許可していない場合でも、リクエストは送られる



XHRを使用するJavaScriptコード

- コード例
  - —http://example.jp/から読み込んだJavaScriptで、http://other.example.jp/と通信を行い、結果を表示する

```
 1:  var url = “http://other.example.jp/”;
 2:  var xhr = new XMLHttpRequest();
 3:  xhr.open( "GET", url, true );
 4:  xhr.onreadystatechange = function(){
 5:      if( xhr.readyState == 4 && xhr.status == 200 ){
 6:          var resp = xhr.responseText;
 7:          …
 8:      }
 9:  };
10:  xhr.send( null );
```

XHRでのHTTPリクエスト・レスポンス例

リクエスト例）

```
GET / HTTP/1.1
Host: other.example.jp
Origin: http://example.jp
User-Agent: Mozilla/5.0(Windows NT 6.0; rv:18.0)
Connection: keep-alive
```

レスポンス例）

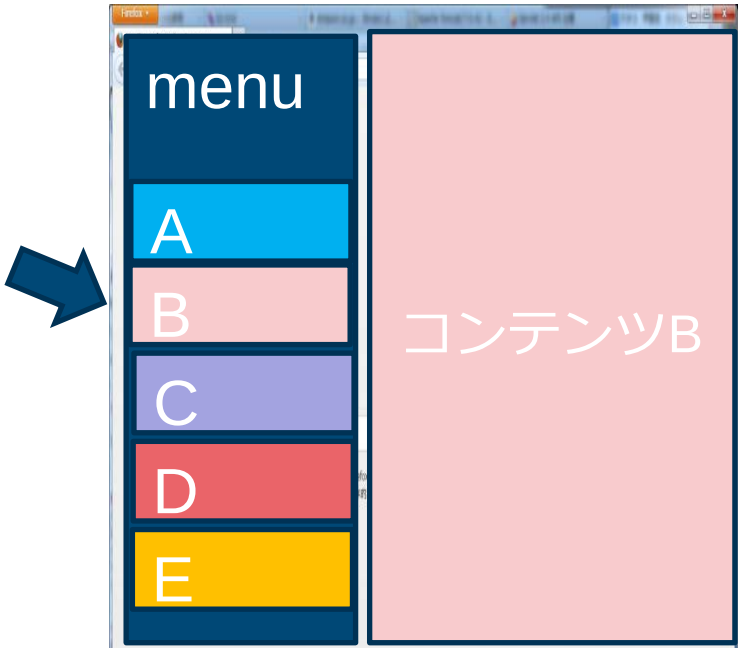
```
HTTP/1.1 200 OK
Date: Tue, 1 Jan 2013 09:00:00 GMT
Content-Length: 1512
Content-Type: text/plain; charset=utf-8
Access-Control-Allow-Origin: http://example.jp
…
```

XHRが脆弱性の原因となるケース

ケース１：XHRを使用した既存のWebサイト

- フレームを使わずXHRで動的にコンテンツを書き換えているケース
  - http://example.jp/#/fooのようなURLでアクセスし、#以降を通信先URLとして使用

http://example.jp/#/b.html



正常な場合の動作

1. http://example.jp/にアクセス
2. 画面左側のメニューからBへのリンクをクリック（http://example.jp/#/b.html）
3. JavaScriptで#以降をURLとして扱い、XHRでコンテンツBの内容を取得
4. XHRのレスポンスを右側のコンテンツとして表示

ケース１：XHRを使用した既存のWebサイト

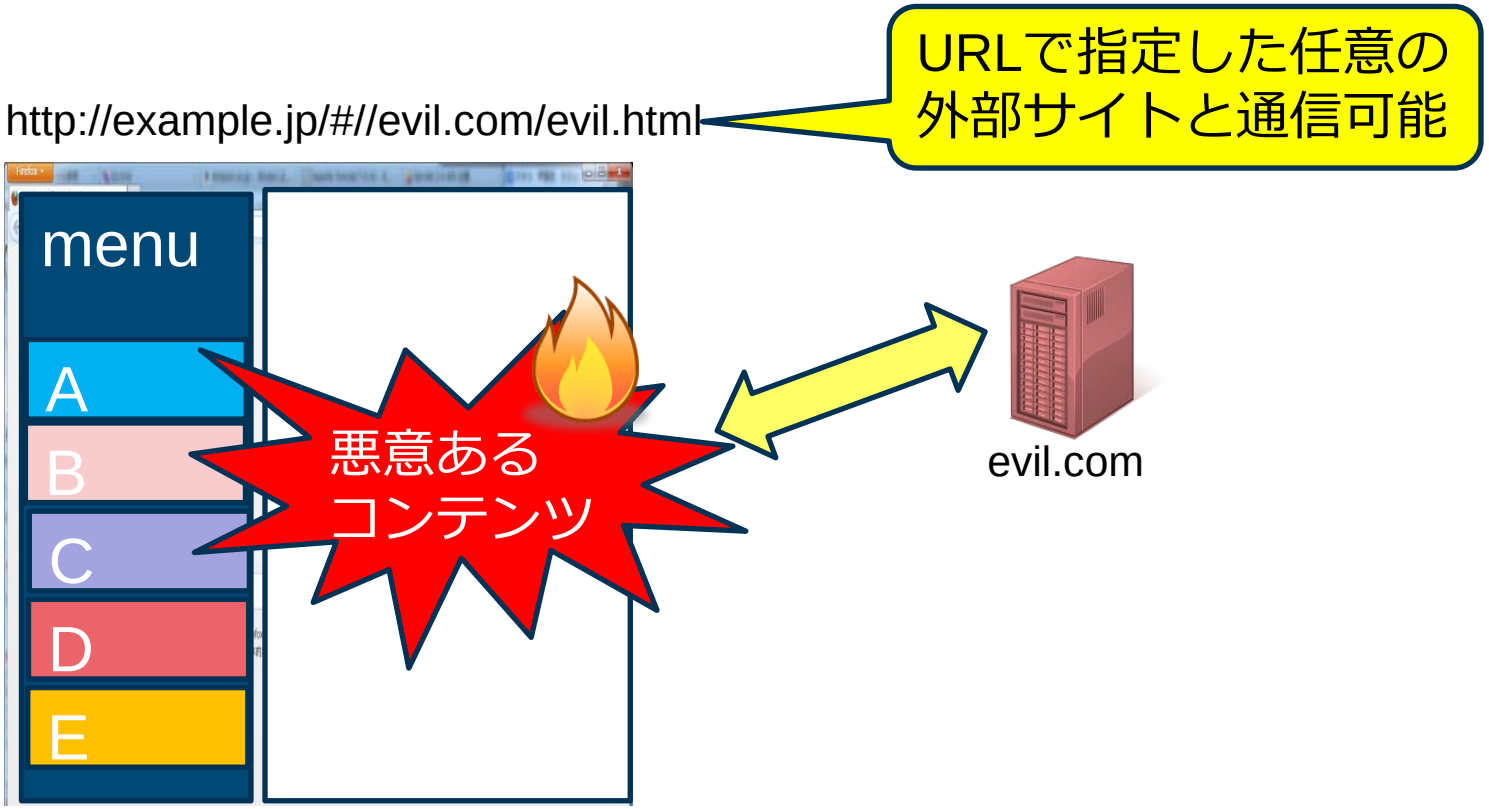
- フレームを使わずXHRで動的にコンテンツを書き換えているケース
  - —http://example.jp/#/fooのようなURLでアクセスし、#以降を通信先URLとして使用

具体的なコード例

```
1:  var url = location.hash.substring(1);
2:  var xhr = new XMLHttpRequest();
3:  xhr.open( "GET", url, true );
4:  xhr.onreadystatechange = function(){
5:      if( xhr.readyState == 4 && xhr.status == 200 ){
6:          div.innerHTML = xhr.responseText;
7:      }
8:  };
9:  xhr.send( null );
```

ケース１：問題

- HTML5未対応ブラウザではリクエスト自体が行われないが、ブラウザがHTML5対応したことにより、外部の任意のサーバと通信可能
  - XSS攻撃に使用される
  - 自サイト内に意図しないコンテンツが表示される



ケース１：対策

- HTML5未対応ブラウザではリクエスト自体が行われないが、ブラウザがHTML5対応したことにより、外部の任意のサーバと通信可能
  - XSS攻撃に使用される
  - 自サイト内に意図しないコンテンツが表示される

任意のサイトをURLで指定しても、
固定した通信先としか通信できない

対策

- 通信先をホワイトリストとして指定しておく事で悪意のある通信先へのアクセスを行わない様にする

```
1:  var pages = [ "/", "/foo", "/bar", "/baz" ];
2:  var index = location.hash.substring(1) | 0;
3:  var url = pages[ index ] || '/';
4:  var xhr = new XMLHttpRequest();
5:  xhr.open( "GET", url, true );
6:  ....
```

ケース１：補足

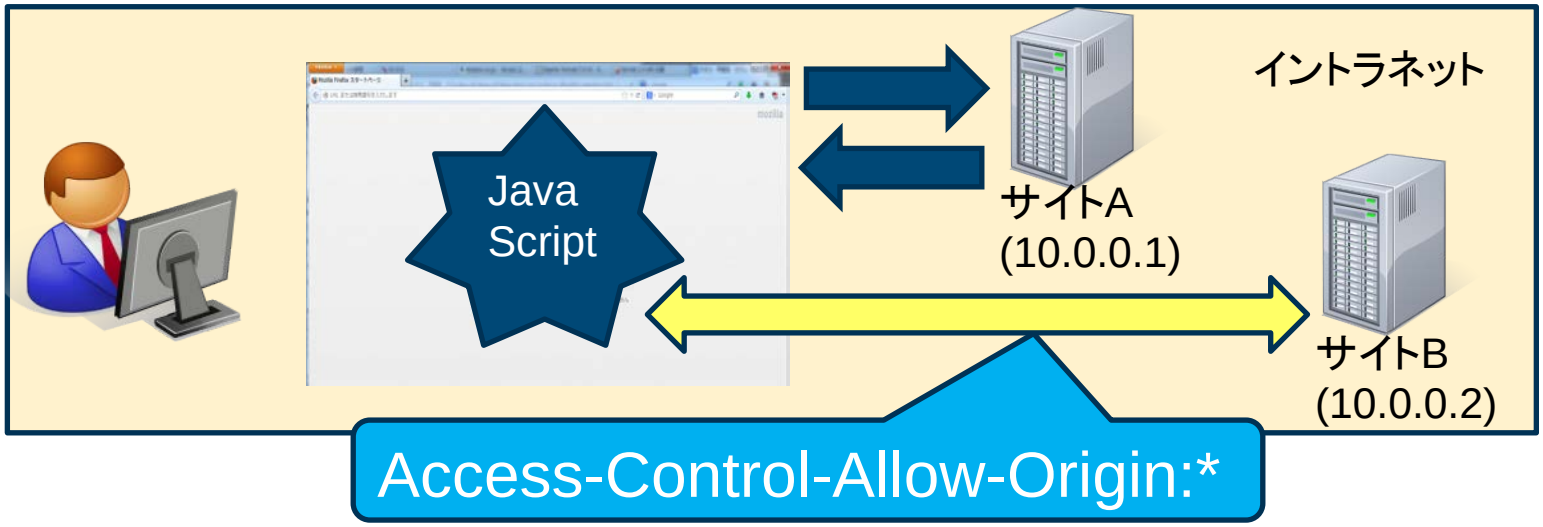
- 脆弱となる可能性のある例

```
1:  var url = location.hash.substring(1);
2:  if( url.indexOf( "http://example2.jp/" ) == 0 ||
3:      url.indexOf( "http://example3.jp/" ) == 0 ){
4:
5:      // example2.jp または example3.jp のときのみ通信
6:      var xhr = new XMLHttpRequest();
7:      xhr.open( "GET", url, true );
8:      ....
9:  }
```

- サイト内にオープンリダイレクタが存在すると、任意のサイトとの通信が可能
- ハッシュを利用した場合、サーバ側にデータが残らないため、どこに接続されたかの把握が困難

ケース２：HTML5対応のイントラ内Webサイト

- イントラネット内のシステム(サイトA、サイトB)
  - サイトAのJSからサイトBに対してクロスオリジン通信を行っている
  - 外部には非公開の情報を取り扱っている
  - イントラネットの外部からのアクセスはできないため、Cookieでの制限は行っていない
  - サイトBでは、レスポンスヘッダにAccess-Controll-Allow-Origin:*を設定



ケース２：攻撃コード例

- 以下のコードを実行するインターネットWebサイト（http://evil.com/）にアクセスするとどうなるか？

```
1:  for(i = 1; i <= 255; i++){
2:      var url = “http://10.0.0.” + i + “/”;
3:      var xhr = new XMLHttpRequest();
4:      xhr.open( "GET", url, true );
5:      xhr.onreadystatechange = function(){
6:          if( xhr.readyState == 4 && xhr.status == 200 ){
7:              // xhr.responseText を evil.com に送信
8:          }
9:      };
10:      xhr.send( null );
11:  }
```

ケース２： HTTPリクエスト・レスポンス例

リクエスト例）

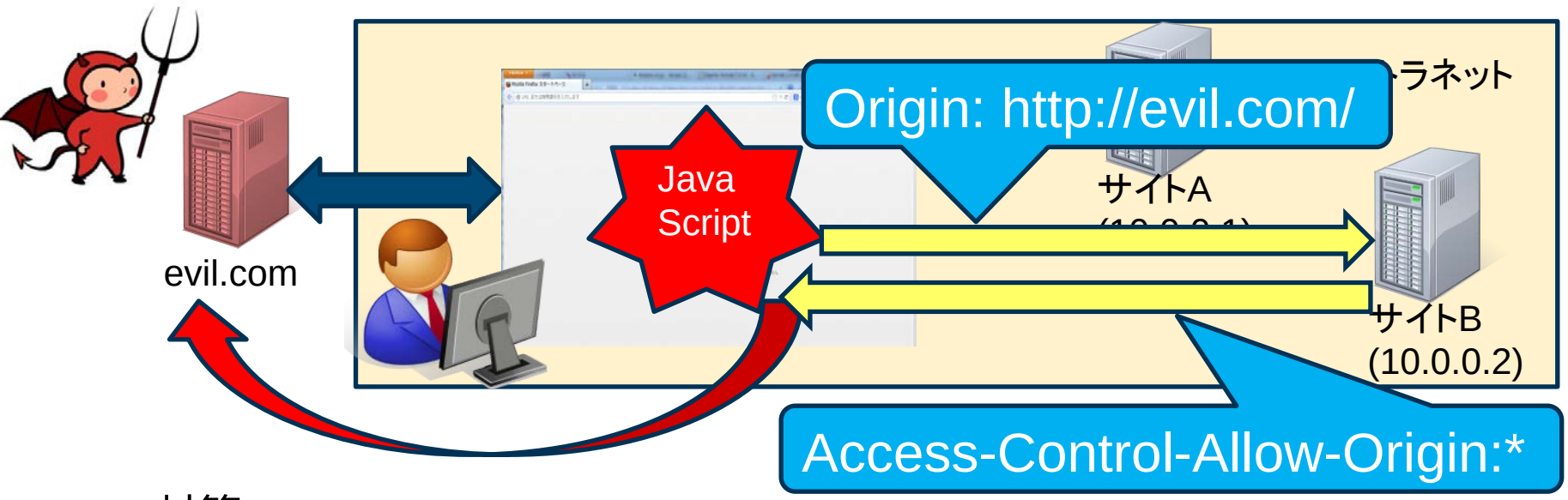
```
GET / HTTP/1.1
Host: 10.0.0.2
Origin: http://evil.com
User-Agent: Mozilla/5.0(Windows NT 6.0; rv:18.0)
Connection: keep-alive
```

レスポンス例）

```
HTTP/1.1 200 OK
Date: Tue, 1 Jan 2013 09:00:00 GMT
Content-Length: 1512
Content-Type: text/plain; charset=utf-8
Access-Control-Allow-Origin: *
…
```

ケース２：問題と対策

- 問題
  - 外部の攻撃者サイトに置かれたJavaScriptのXHRから、イントラネット内部の情報にアクセスされ情報漏えいに繋がる可能性がある



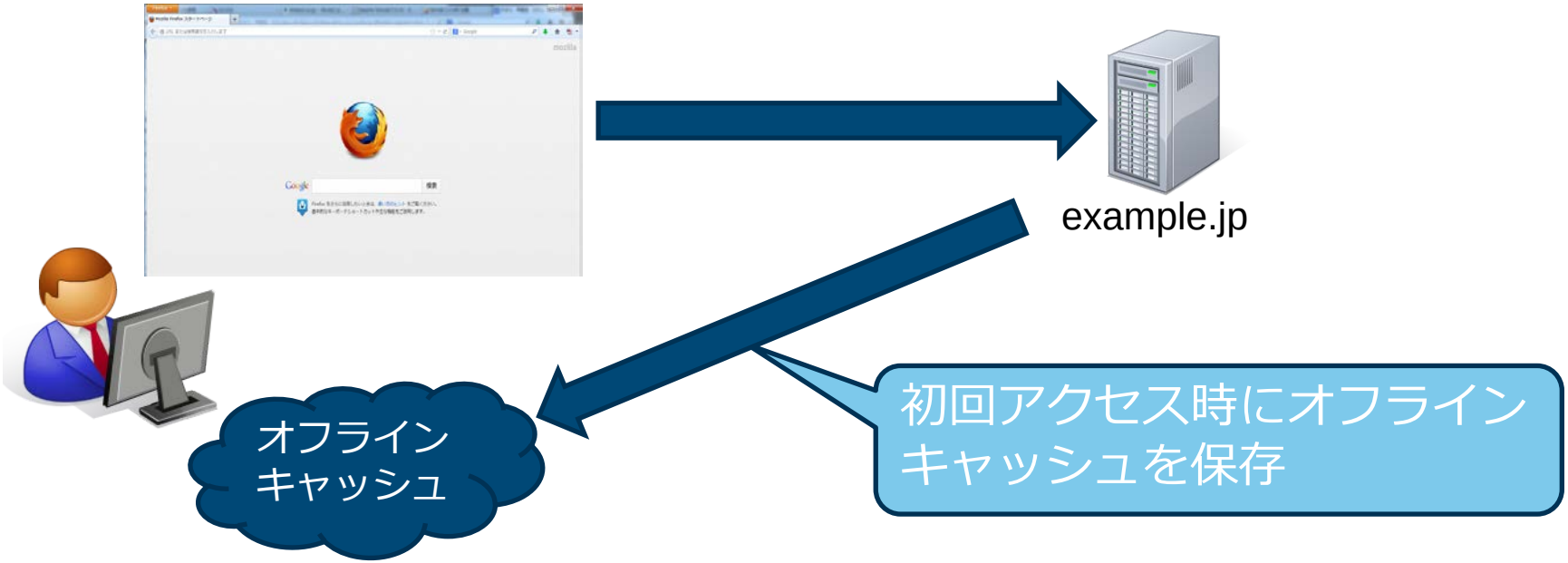
- 対策
  - 閲覧を制限したいコンテンツの場合
    - Cookieでアクセスを制限する（Cookieを使う場合、Access-Control-Allow-Originに、*は指定できず、オリジンを指定する必要がある）

※Access-Control-Allow-Originは、アクセス制御の目的で使用できないことに注意

JavaScript API
Offline Web Application

Offline Web Application概要

- 機能概要
  - 指定したリソースをオフラインキャッシュとしてローカルデバイス上に保存しておくことにより、ネットワークに接続されていない状況でも、Webアプリケーションが利用できる

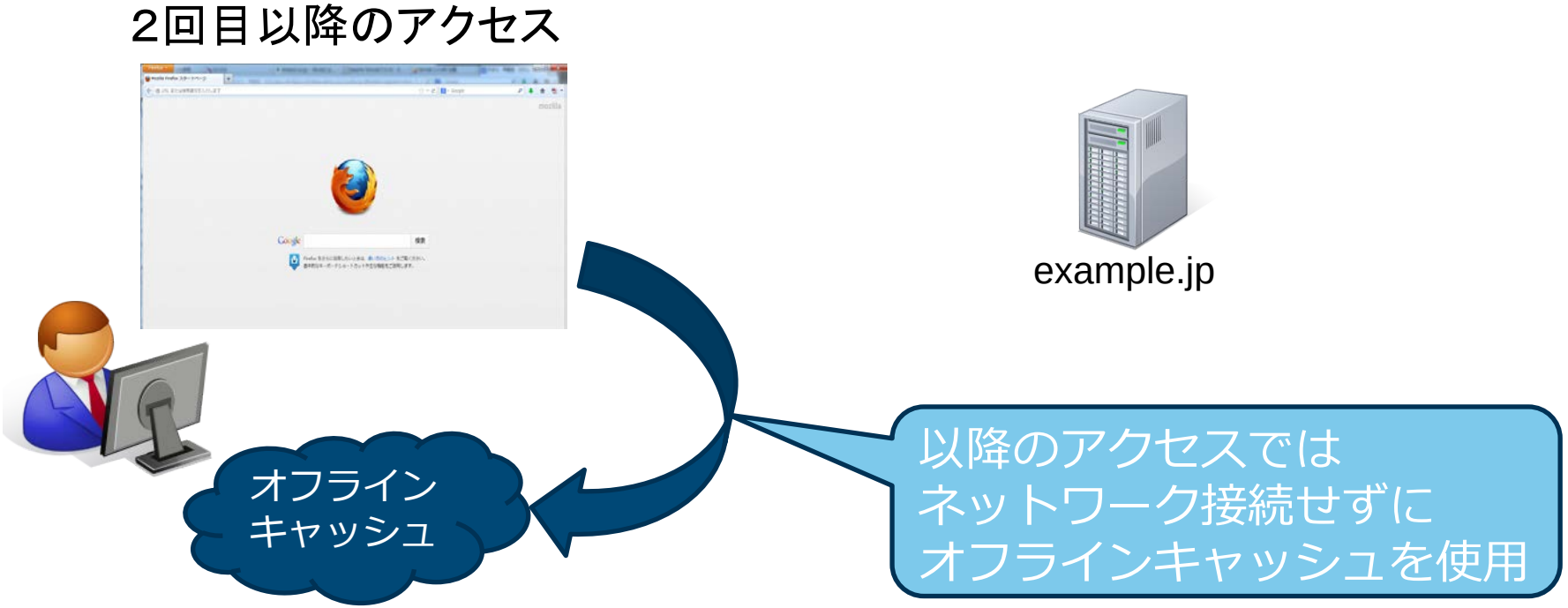


Offline Web Application概要

- 機能概要
  - 指定したリソースをオフラインキャッシュとしてローカルデバイス上に保存しておくことにより、ネットワークに接続されていない状況でも、Webアプリケーションが利用できる

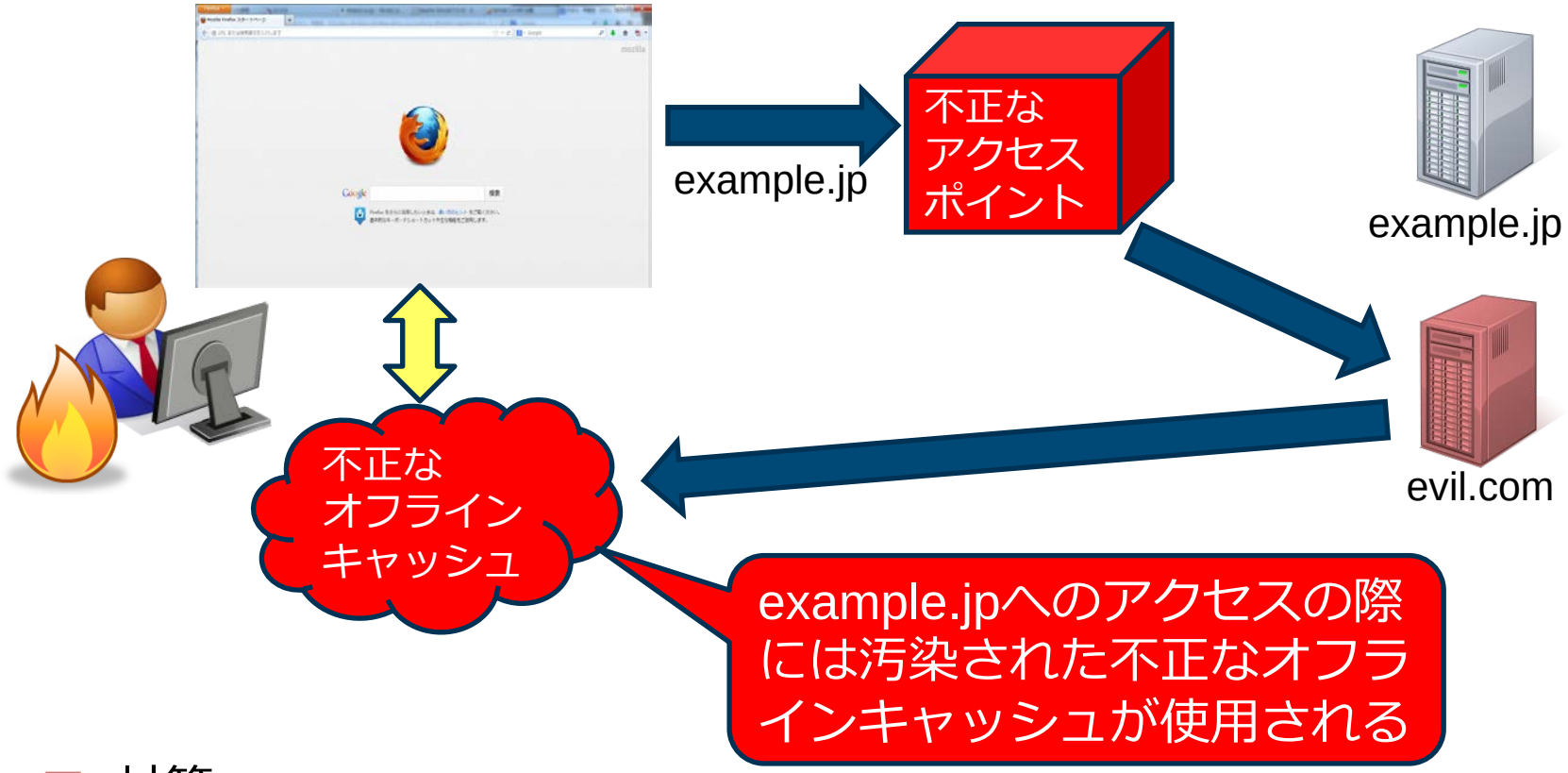
Offline Web Application概要

- 機能概要
  - 指定したリソースをオフラインキャッシュとしてローカルデバイス上に保存しておくことにより、ネットワークに接続されていない状況でも、Webアプリケーションが利用できる



Offline Web Application

- 問題
  - 中間者攻撃により汚染されたリソースを保存した場合、そのネットワークを離れた後も、汚染されたリソースを使い続ける

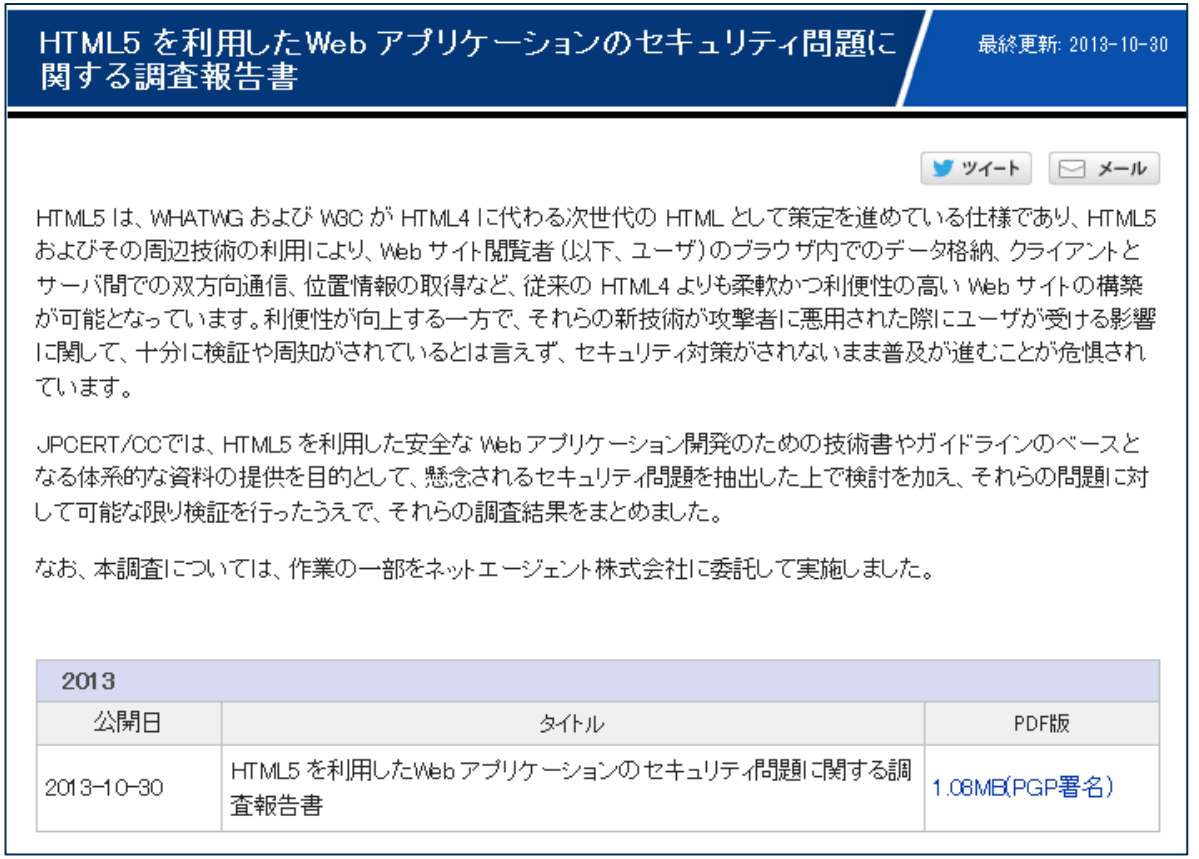


- 対策
  - 当該リソースを持つサイト全体に対してHTTPSを利用する

その他のJavaScript API

- 報告書では、今回紹介した以外にも様々な機能・問題を紹介しています
  - Cross Document Messaging
  - Web Storage
  - WebSocket
  - Web Workers
  - XHRによるCSRF

  など



HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書

最終更新: 2013-10-30

ツイート　メール

HTML5 は、WHATWG および W3C が HTML4 に代わる次世代の HTML として策定を進めている仕様であり、HTML5 およびその周辺技術の利用により、Web サイト閲覧者（以下、ユーザ）のブラウザ内でのデータ格納、クライアントとサーバ間での双方向通信、位置情報の取得など、従来の HTML4 よりも柔軟かつ利便性の高い Web サイトの構築が可能となっています。利便性が向上する一方で、それらの新技術が攻撃者に悪用された際にユーザが受ける影響に関して、十分に検証や周知がされているとは言えず、セキュリティ対策がされないまま普及が進むことが危惧されています。

JPCERT/CCでは、HTML5 を利用した安全な Web アプリケーション開発のための技術書やガイドラインのベースとなる体系的な資料の提供を目的として、懸念されるセキュリティ問題を抽出した上で検討を加え、それらの問題に対して可能な限り検証を行ったうえで、それらの調査結果をまとめました。

なお、本調査については、作業の一部をネットエージェント株式会社に委託して実施しました。

2013

公開日	タイトル	PDF版
2013-10-30	HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書	1.08MB(PGP署名)

報告書の内容を紹介

JavaScript API

HTML新要素・属性

セキュリティ関連機能

HTML新要素・属性

HTML新要素・属性

- <**video**>, <**audio**>
  - —<img>と同様にHTML内に直接再生可能な動画・音声を埋め込む
- <button **formaction**="xxx">
  - —ボタンが押された際のフォームの挙動を指定する
- <input type="**email**">
  - —メール形式のデータのみを受け付ける
- <input type="text" **pattern**="^[0-9a-fA-F]+$">
  - —パターンにマッチするデータのみを受け付ける
- <iframe **sandbox**>
  - —scriptの実行や、フォームの送信、トップレベルWindowへの干渉などを制限する

HTML新要素・属性

- <video>, <audio>
  - —<img>と同様にHTML内に直接再生可能な動画・音声を埋め込む
- <button formaction="xxx">
  - —ボタンが押された際のフォームの挙動を指定する
- <input type="email">
  - —メール形式のデータのみを受け付ける
- <input type="text" pattern="^[0-9a-fA-F]+$">
  - —パターンにマッチするデータのみを受け付ける
- <iframe sandbox>
  - —scriptの実行や、フォームの送信、トップレベルWindowへの干渉などを制限する

HTML新要素・属性：<video>,<audio>

- Internet Explorer 9 および10 では、<video>,<audio>要素の onerror イベントが動作するため、XSS 攻撃が行われる可能性がある。

```
<!-- video 要素によるXSS 攻撃の例 -->
<video onerror="javascript:alert(1)">
    <source src="#"></source>
</video>
```

```
<!-- audio 要素によるXSS 攻撃の例 -->
<audio onerror="javascript:alert(1)">
    <source src="#"></source>
</audio>
```

HTML新要素・属性

- <**video**>, <**audio**>
  - —<img>と同様にHTML内に直接再生可能な動画・音声を埋め込む
- <button **formaction**="xxx">
  - —ボタンが押された際のフォームの挙動を指定する
- <input type="**email**">
  - —メール形式のデータのみを受け付ける
- <input type="text" **pattern**="^[0-9a-fA-F]+$">
  - —パターンにマッチするデータのみを受け付ける
- <iframe **sandbox**>
  - —scriptの実行や、フォームの送信、トップレベルWindowへの干渉などを制限する

HTML新要素・属性：<button formaction>

- 生成する HTML へのイベントハンドラの挿入を禁ずるために、on で始まる属性を検出する対策をとる場合があったが、このような対策では不十分

```
<form>
    <button formaction="javascript:alert(1)">text</button>
</form>
```

- 従来は onmouseover, onclickといったユーザの操作が必要な属性が攻撃に使われたが、autofocus 属性と組み合わせることで、ユーザの操作なしに攻撃可能になった

```
<!-- ユーザの操作が不要な攻撃の例 -->
<button autofocus onfocus="alert(1)">
```

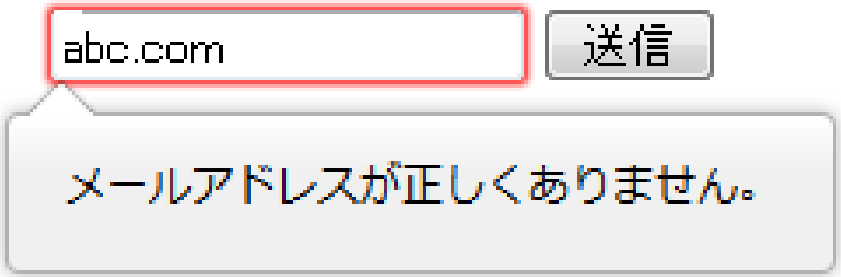
HTML新要素・属性

- <**video**>, <**audio**>
  - —<img>と同様にHTML内に直接再生可能な動画・音声を埋め込む
- <button **formaction**="xxx">
  - —ボタンが押された際のフォームの挙動を指定する
- <input type="**email**">
  - —メール形式のデータのみを受け付ける
- <input type="text" **pattern**="^[0-9a-fA-F]+$">
  - —パターンにマッチするデータのみを受け付ける
- <iframe **sandbox**>
  - —scriptの実行や、フォームの送信、トップレベルWindowへの干渉などを制限する

HTML新要素・属性：<input type="email"> <input pattern="xxx">

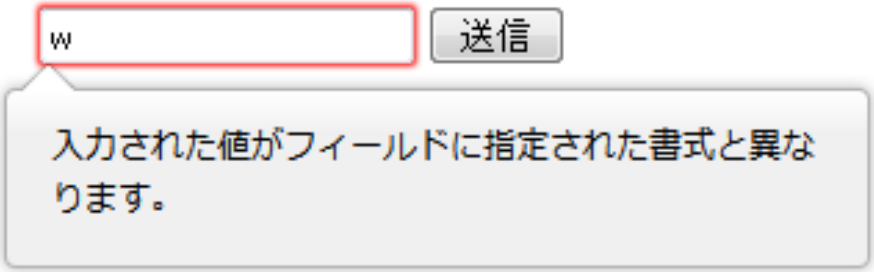
- メールアドレス形式の入力データのみを受け付ける

```
<form>
    <input type="email">
    <input type="submit">
</form>
```



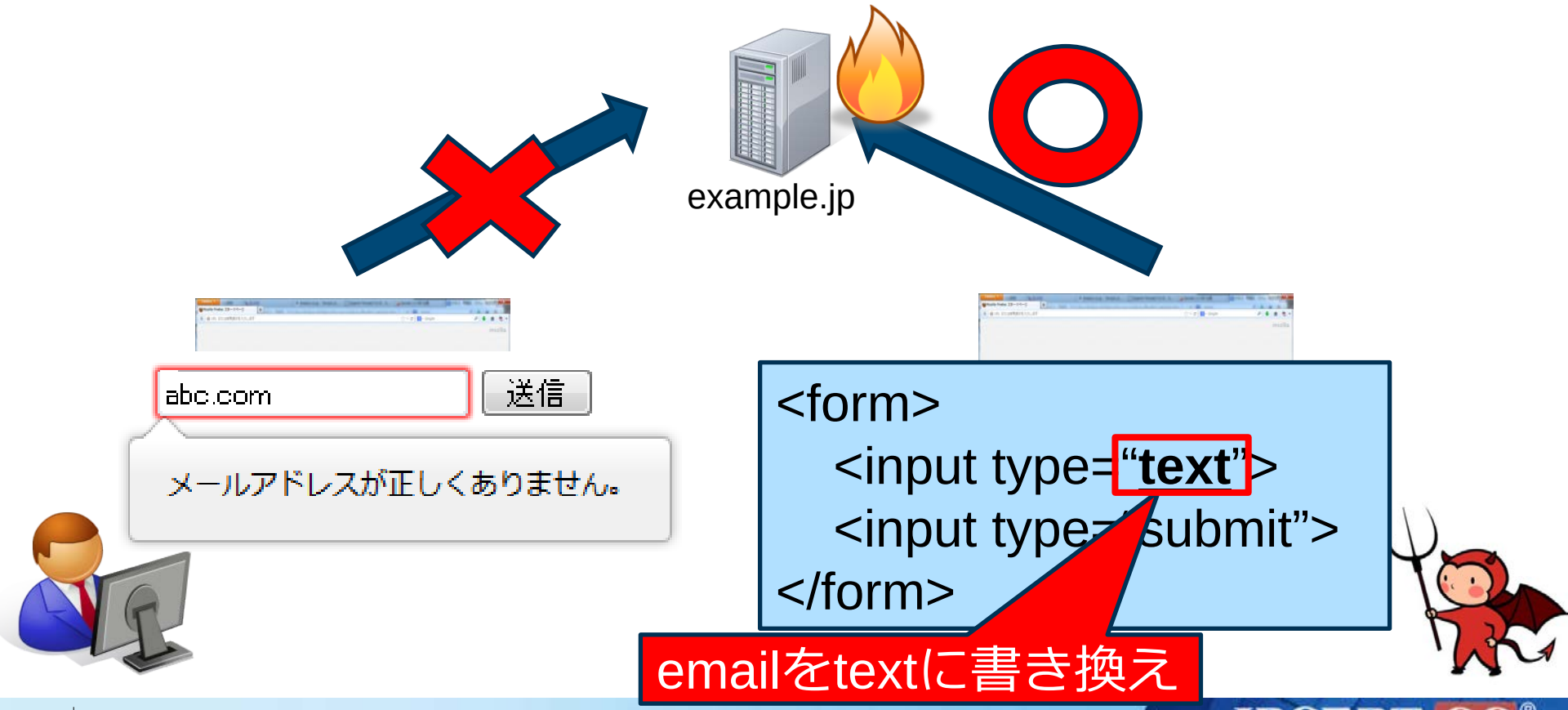
- patternにマッチする入力データのみを受け付ける
  （下の例では、16進数表現のみを受け付ける）

```
<form>
    <input type="text"
      pattern="^[0-9a-fA-F]+$">
    <input type="submit">
</form>
```



HTML新要素・属性：<input type="email"> <input pattern="xxx">

- 攻撃者はブラウザ内でHTML を書き換え、<input>要素による入力制限を回避することが可能であるため、本機能を入力値に対するセキュリティ対策として使用してはいけない。

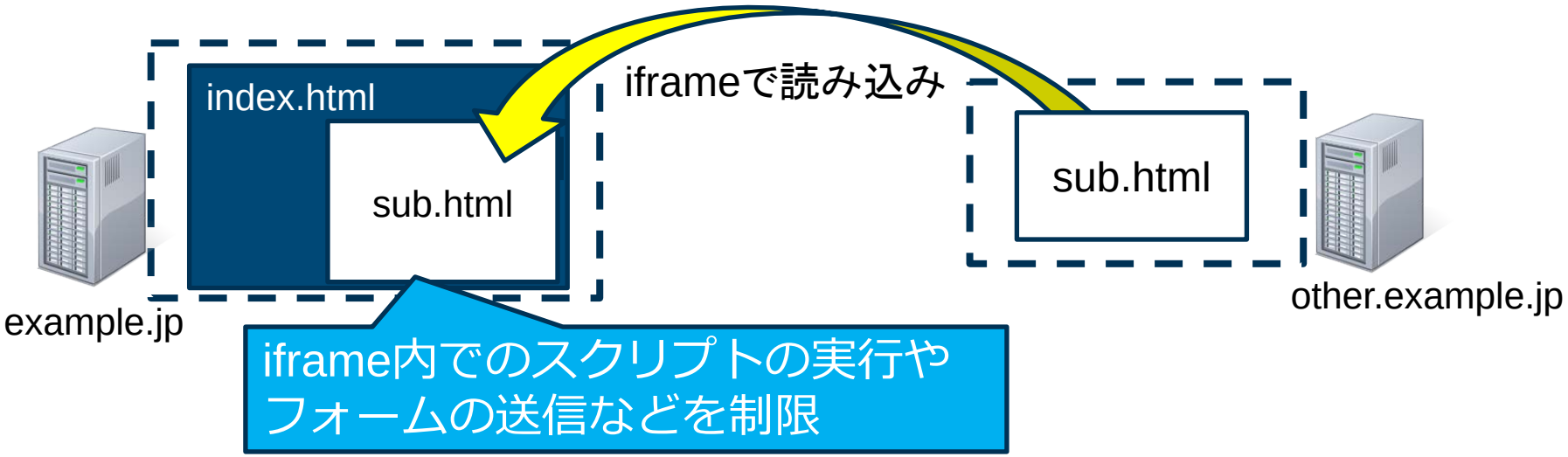


HTML新要素・属性

- <**video**>, <**audio**>
  - —<img>と同様にHTML内に直接再生可能な動画・音声を埋め込む
- <button **formaction**="xxx">
  - —ボタンが押された際のフォームの挙動を指定する
- <input type="**email**">
  - —メール形式のデータのみを受け付ける
- <input type="text" **pattern**="^[0-9a-fA-F]+$">
  - —パターンにマッチするデータのみを受け付ける
- <iframe **sandbox**>
  - —scriptの実行や、フォームの送信、トップレベルウィンドウへの干渉などを制限する

HTML新要素・属性：<iframe sandbox>

```
<iframe sandbox
src="http://other.example.jp/sub.html"></iframe>
```

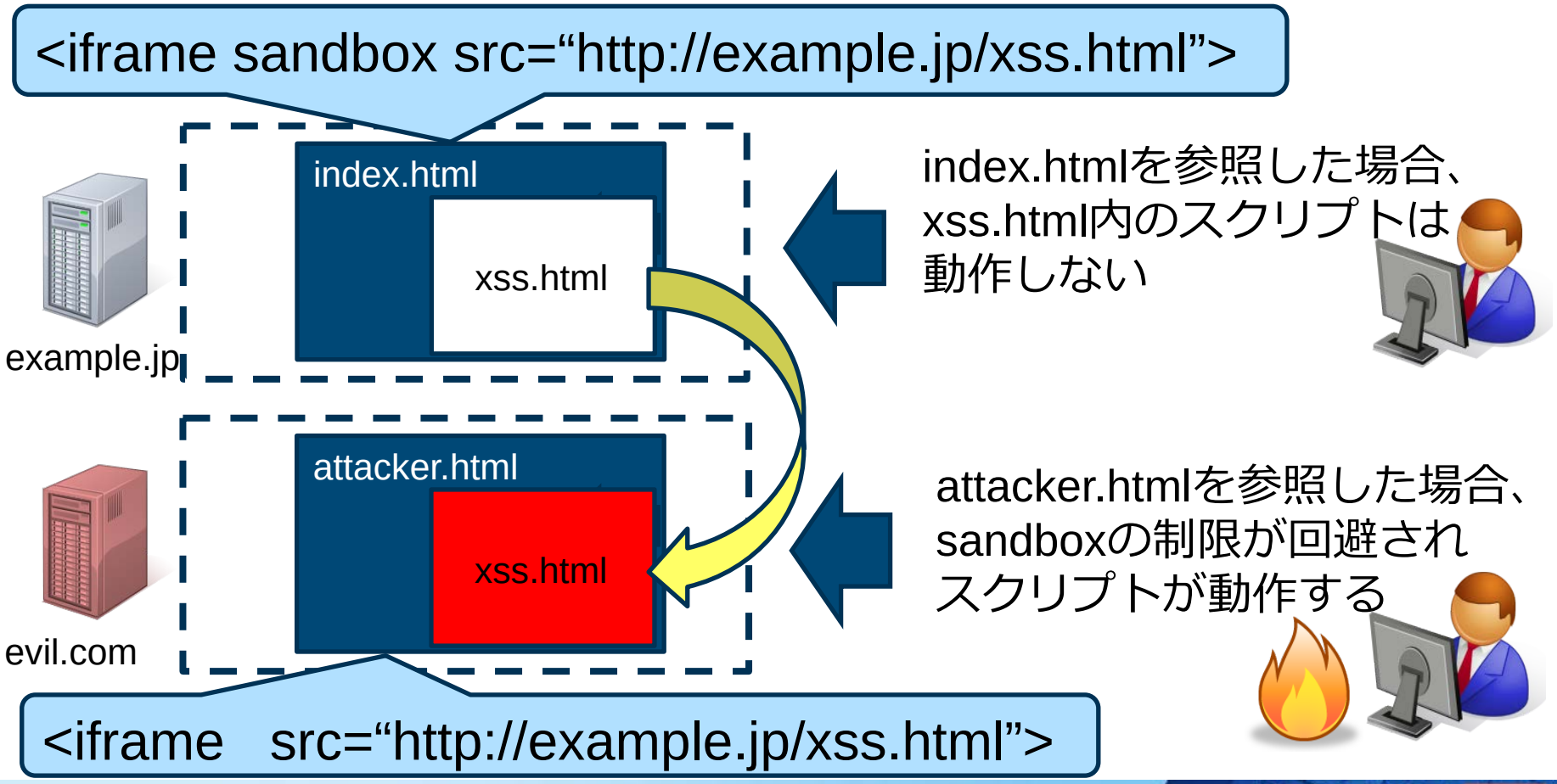


sandbox属性の値に以下を指定することで制限を解除することも可能

sandboxに指定可能な値	説明
allow-scripts	スクリプトの実行を許可する
allow-forms	フォームの送信を許可する
allow-top-navigation	トップレベルウィンドウの操作を許可する

HTML新要素・属性：<iframe sandbox>

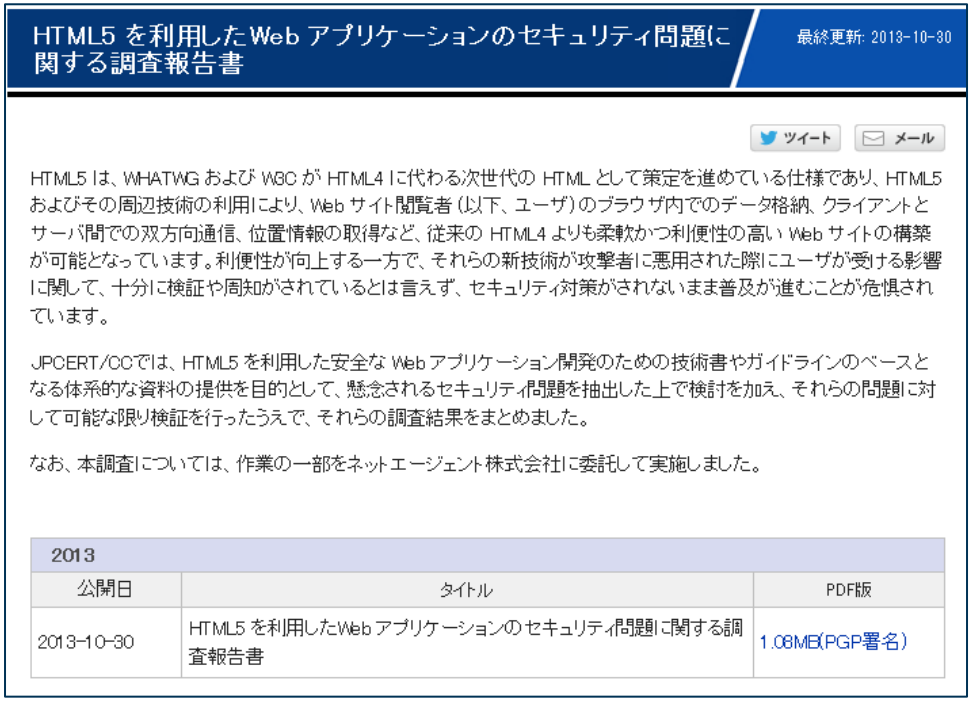
- 自身のページを<iframe sandbox>で読み込むようにしても、攻撃者は直接ページを参照させることが可能なため、XSS攻撃などを防ぐ対策にはならない



その他のHTML新要素・属性

- 報告書では、今回紹介した以外にも様々な機能・問題を紹介しています
  - —<a>
  - —<canvas>
  - —<meta>
  - —<source>
  - —JSONハイジャック

  など



HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書

最終更新: 2013-10-30

ツイート　メール

HTML5 は、WHATWG および W3C が HTML4 に代わる次世代の HTML として策定を進めている仕様であり、HTML5 およびその周辺技術の利用により、Web サイト閲覧者（以下、ユーザ）のブラウザ内でのデータ格納、クライアントとサーバ間での双方向通信、位置情報の取得など、従来の HTML4 よりも柔軟かつ利便性の高い Web サイトの構築が可能となっています。利便性が向上する一方で、それらの新技術が攻撃者に悪用された際にユーザが受ける影響に関して、十分に検証や周知がされているとは言えず、セキュリティ対策がされないまま普及が進むことが危惧されています。

JPCERT/CCでは、HTML5 を利用した安全な Web アプリケーション開発のための技術書やガイドラインのベースとなる体系的な資料の提供を目的として、懸念されるセキュリティ問題を抽出した上で検討を加え、それらの問題に対して可能な限り検証を行ったうえで、それらの調査結果をまとめました。

なお、本調査については、作業の一部をネットエージェント株式会社に委託して実施しました。

2013

公開日	タイトル	PDF版
2013-10-30	HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書	1.08MB(PGP署名)

報告書の内容を紹介

JavaScript API
HTML新要素・属性
セキュリティ関連機能

セキュリティ関連機能

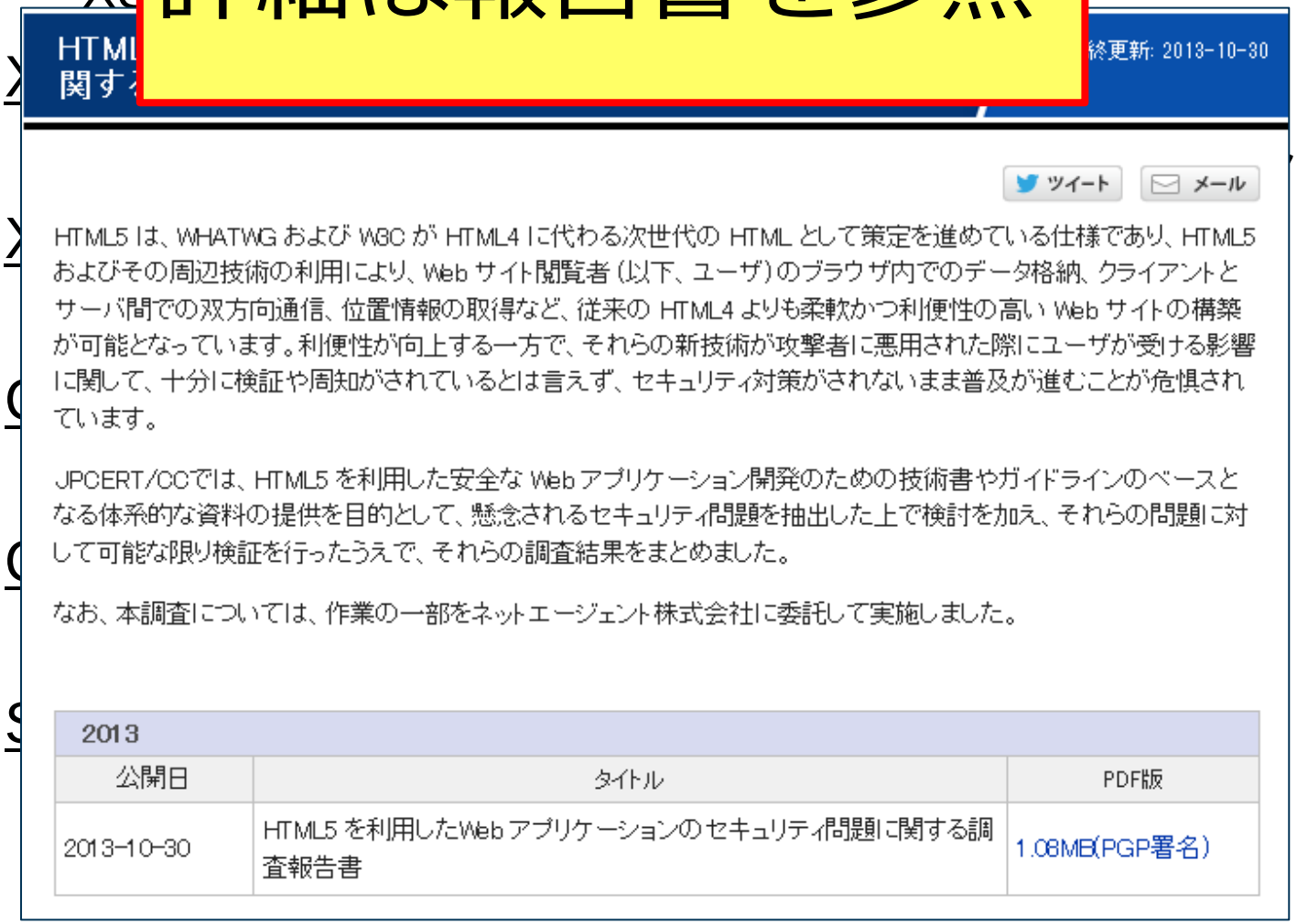
セキュリティ関連機能

- X-XSS-Protection
  - —XSS攻撃からの保護
- X-Content-Type-Options
  - —Content-Typeヘッダに従ったコンテンツの取り扱い
- X-Frame-Options
  - —フレームへの埋め込みを制限
- Content-Security-Policy
  - —コンテンツの読み込み元を制限
- Content-Disposition
  - —ファイルのダウンロードダイアログの制御
- Strict-Transport-Security
  - —HTTPSの強制

セキュリティ関連機能

- X-XS
  - —XS
- X
- X
- C
- C
- S

詳細は報告書を参照



HTML
関す

終更新: 2013-10-30

ツイート　メール

HTML5 は、WHATWG および W3C が HTML4 に代わる次世代の HTML として策定を進めている仕様であり、HTML5 およびその周辺技術の利用により、Web サイト閲覧者（以下、ユーザ）のブラウザ内でのデータ格納、クライアントとサーバ間での双方向通信、位置情報の取得など、従来の HTML4 よりも柔軟かつ利便性の高い Web サイトの構築が可能となっています。利便性が向上する一方で、それらの新技術が攻撃者に悪用された際にユーザが受ける影響に関して、十分に検証や周知がされているとは言えず、セキュリティ対策がされないまま普及が進むことが危惧されています。

JPCERT/CCでは、HTML5 を利用した安全な Web アプリケーション開発のための技術書やガイドラインのベースとなる体系的な資料の提供を目的として、懸念されるセキュリティ問題を抽出した上で検討を加え、それらの問題に対して可能な限り検証を行ったうえで、それらの調査結果をまとめました。

なお、本調査については、作業の一部をネットエージェント株式会社に委託して実施しました。

2013		
公開日	タイトル	PDF版
2013-10-30	HTML5 を利用したWeb アプリケーションのセキュリティ問題に関する調査報告書	1.08MB(PGP署名)

まとめ

まとめ

- HTML5を用いることで、従来よりも柔軟かつ利便性の高いWebサイトの構築が可能になる一方で、開発時にはセキュリティ対策が必要となる機能も多数ある

- 「HTML5を利用したWebアプリケーションのセキュリティ問題に関する調査報告書」をセキュアなWebアプリケーションの開発に役立ててください

- お問い合わせは以下まで
  - ―ご意見お待ちしています

JPCERTコーディネーションセンター
Email：ww-info@jpcert.or.jp
Tel：03-3518-4600
Web: https://www.jpcert.or.jp/

ご静聴ありがとうございました