ぼくたちの愛したIE8

はせがわようすけ
@hasegawayosuke

自己紹介

はせがわようすけ @hasegawayosuke

- XSSのほうから来ました
- http://utf-8.jp/
  - author of jjencode, aaencode


いよいよお別れの時が迫る
提供
Shibuya.XSS
UTF-8.jp
みんな大好きなあのブラウザと

さよなら
Internet Explorer 8

Internet Explorer 8

- タイムライン
  - 2005-06年　Ajax、Web 2.0
  - 2006年10月　Internet Explorer 7公開
  - 2008年1月　HTML5 最初のドラフト公開
  - 2008年8月　ECMAScript 4廃案
  - 2008年9月　Google Chrome公開
  - 2009年3月　Internet Explorer 8 公開
    2010年2月にはブラウザトップシェア
  - 2011年3月　Internet Explorer 9 公開
  - 2012年8月　Internet Explorer 10 公開
  - 2013年10月　Internet Explorer 11 公開
  - 2015年7月　Microsoft Edge 公開
  - 2016年1月13日　IE8サポート終了

Internet Explorer 8

- 2009年3月リリース
- 時代はAjax、Web2.0、マッシュアップブーム
  - evalでのJSONパース
  - JSONPを用いたクロスオリジンのデータ交換
  - window.nameを用いたiframe間通信
- HTML5、ES5の制定が始まったばかり
- MSが標準化路線へ舵を切る直前

Internet Explorer 8

- IE8、実はすごいブラウザだった
  - MSらしくない挑戦的な機能が山盛り
  - 特に、セキュアにするための機能もたくさん
  - 独自実装だったものから標準化された機能もたくさん
- 今日はそんなIE8のすごいセキュリティの機能を振り返ります
  - あんまりJS関係なくてすみません＞＜

IE8のここがすごい その1

XSSフィルター

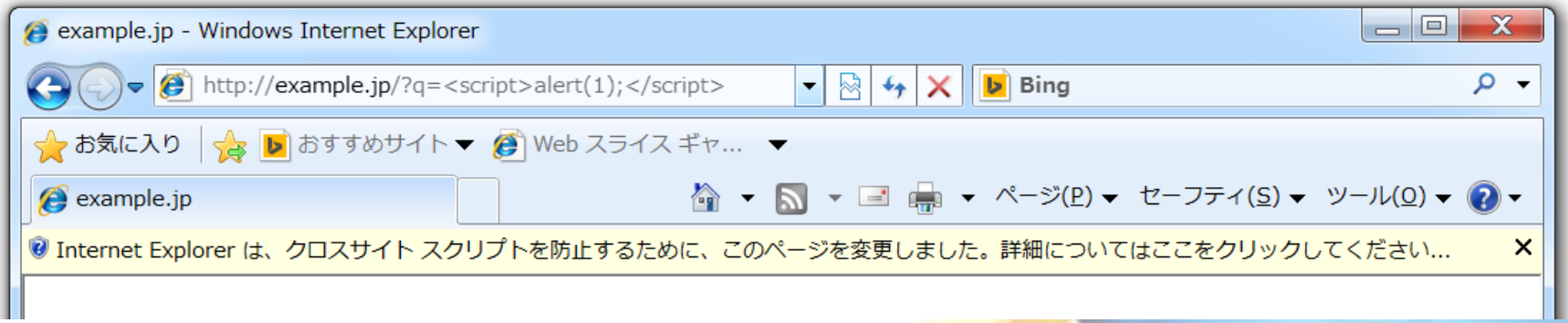
XSSフィルター

- リクエストとレスポンスを比較し、スクリプトっぽい文字列があれば実行を阻止

```
GET /?q=<script>alert(1);</script> HTTP/1.1

HTTP/1.1 200 OK
Content-Type: text/html

<script>alert(1);</script>
```



XSSフィルターここがすごい

- 保守的と言われるMSがよく導入した
- 他のブラウザに先行
  - NoScript(Fx)、XSS Auditor(WebKit)、CSP(Fx)
- 「XSSに関連する脆弱性の37%をXSSフィルターによって低減できた」
  http://download.microsoft.com/download/C/1/F/C1F6A2B2-F45F-45F7-B788-32D2CCA48D29/Microsoft_Security_Intelligence_Report_Volume_13_English.pdf

XSSフィルターここがすごい

- 当然、誤検出・副作用も多数発生

XSSフィルターここがすごい

- XSSフィルターの誤検出の悪用
  - 任意サイトでXSSを発生させる
  - 安全なサイトがXSSフィルターのせいで脆弱になる
- Universal XSS via IE8s XSS Filters

  Eduardo Vela Nava, David Lindsay
  https://media.blackhat.com/bh-eu-10/presentations/Lindsay_Nava/BlackHat-EU-2010-Lindsay-Nava-IE8-XSS-Filters-slides.pdf
- IE/EdgeのXSSフィルターを利用したXSS

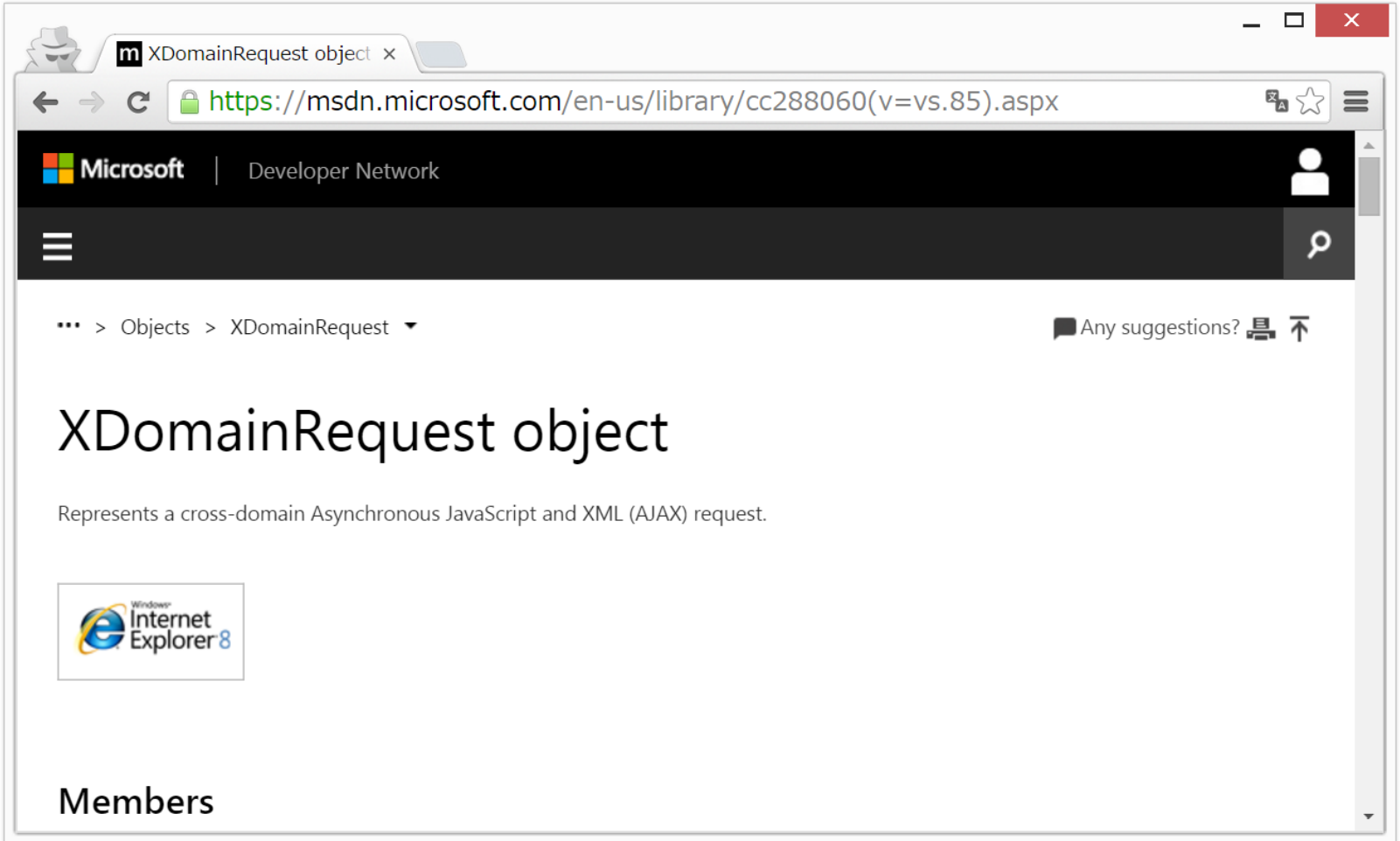
  Masato Kinugawa
  http://masatokinugawa.l0.cm/2015/12/xxn.html

IE8のここがすごい その2

XDomainRequest

XDomainRequest

- クロスオリジンで使えるXMLHttpRequest(ﾓﾄﾞｷ
  - XHR Level 2なんて必要なかった!



XDomainRequest ここがすごい

- CORSヘッダを先取り

```
GET /text.txt HTTP/1.1
Host: another.example.com
Origin: http://from.example.com

HTTP/1.1 200 OK
Content-Type: text/plain
Access-Control-Allow-Origin: http://from.example.com
```

- MSDNにはURLを指定って書いてあるのはご愛敬

> The document will request data from the domain's server by sending an **Origin** header with the value of the origin. It will only complete the connection if the server responds with an **Access-Control-Allow-Origin** header of either * or the exact URL of the requesting document. This behavior is part of the World Wide Web Consortium (W3C)'s Web Application Working Group's draft framework on client-side cross-domain communication that the **XDomainRequest** object integrates with.

XDomainRequest ここがすごい

- エラー情報が取れない

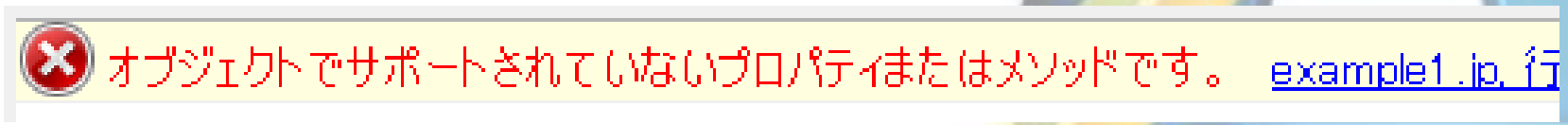
```
var xdr = new XDomainRequest();
xdr.open( "GET", "http://another.example.com/" );
xdr.onload = function(){ ... };
xdr.onerror = function(){
    alert( "詳細わからないけどとにかくエラー!" );
};
xdr.send( null );
```

  - エラー情報を示すようなプロパティ等何もない

XDomainRequest ここがすごい

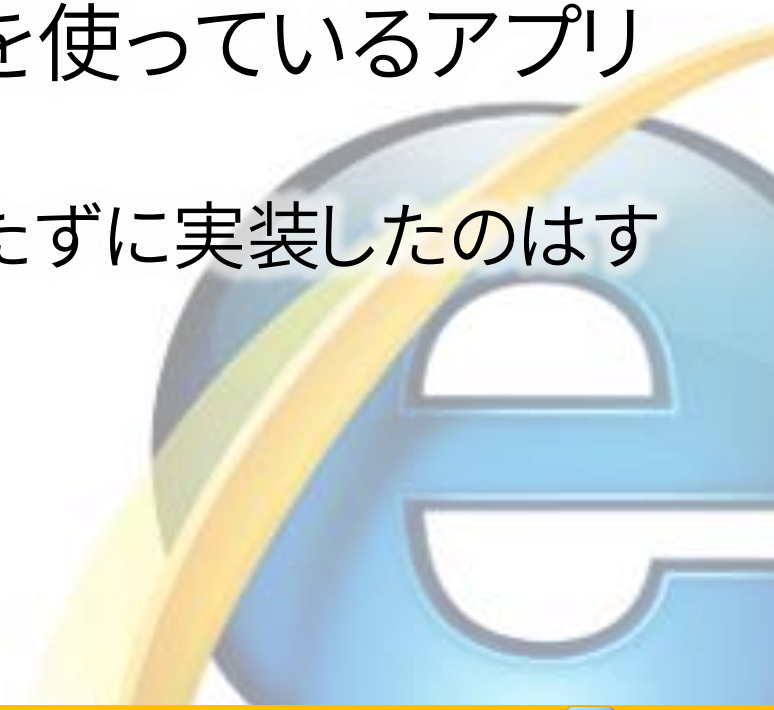
- リクエストヘッダ、レスポンスヘッダを操作できない

```
var xdr = new XDomainRequest();
xdr.open( "GET", "http://from.example.com/" );
xdr.setRequestHeader( "X-test", "42" );
xdr.onload = function(){
    xdr.getResponseHeader( "X-test" );
}
xdr.send();
```



XDomainRequest

- XDRここがすごい
  - CORSヘッダを先取り
  - エラー情報が取れない
  - リクエストヘッダ、レスポンスヘッダを操作できない
- まともにXDomainRequestを使っているアプリは見たことがない
  - XMLHttpRequest Lv.2を待たずに実装したのはすごい

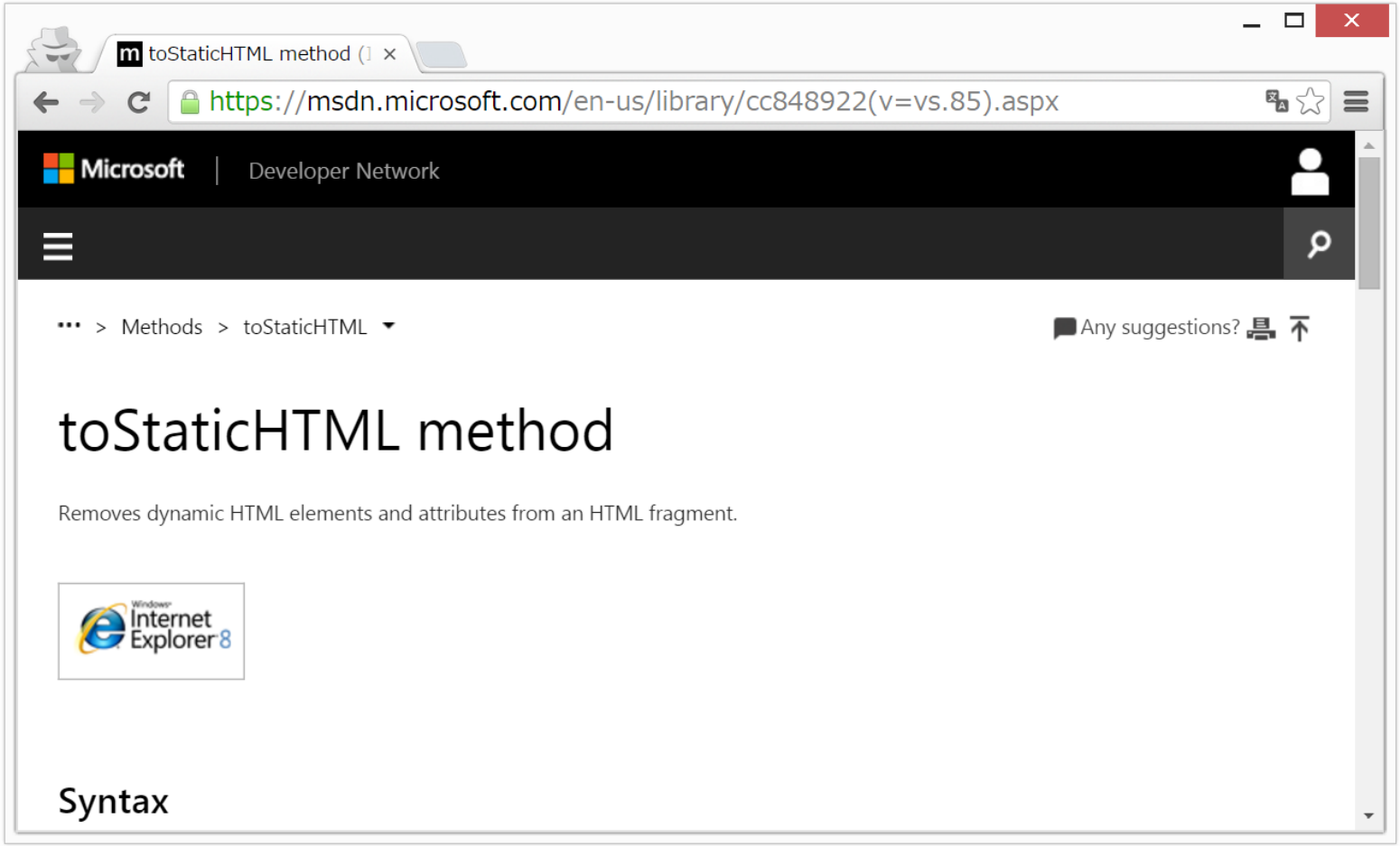


IE8のここがすごい その3

toStaticHTML

toStaticHTML

- HTMLを表す文字列内から危険そうな文字列を削除して安全なHTMLを返す



toStaticHTMLここがすごい

- いい感じに安全なHTMLにしてくれる

```
toStaticHTML("<img src=1 onerror=alert(1)>");
→ "<img src=1>"

toStaticHTML("<div><script></script></div>");
→ "<div></div>"

toStaticHTML("<s style='color:red;x:expression(alert(1))'>a</s>")
→"<s style="color:red">a</s>"
```

toStaticHTMLここがすごい

- いい感じに安全なHTMLにしてくれる

```
div.innerHTML = toStaticHTML( insecureInput );
```

  - HTMLメールとかMarkdownなど部分的にHTMLを許すアプリケーションに便利
  - 「このタグとこの属性は許す」みたいな細かい指定は何もできない

toStaticHTMLここがすごい

- ときどきtoStaticHTMLの漏れが発生してる
  - CVE-2010-1257 CVE-2010-3243
    CVE-2011-1252 CVE-2012-1858
- 「ブラウザの問題」でありWebアプリ側の責任ではない

toStaticHTMLここがすごい

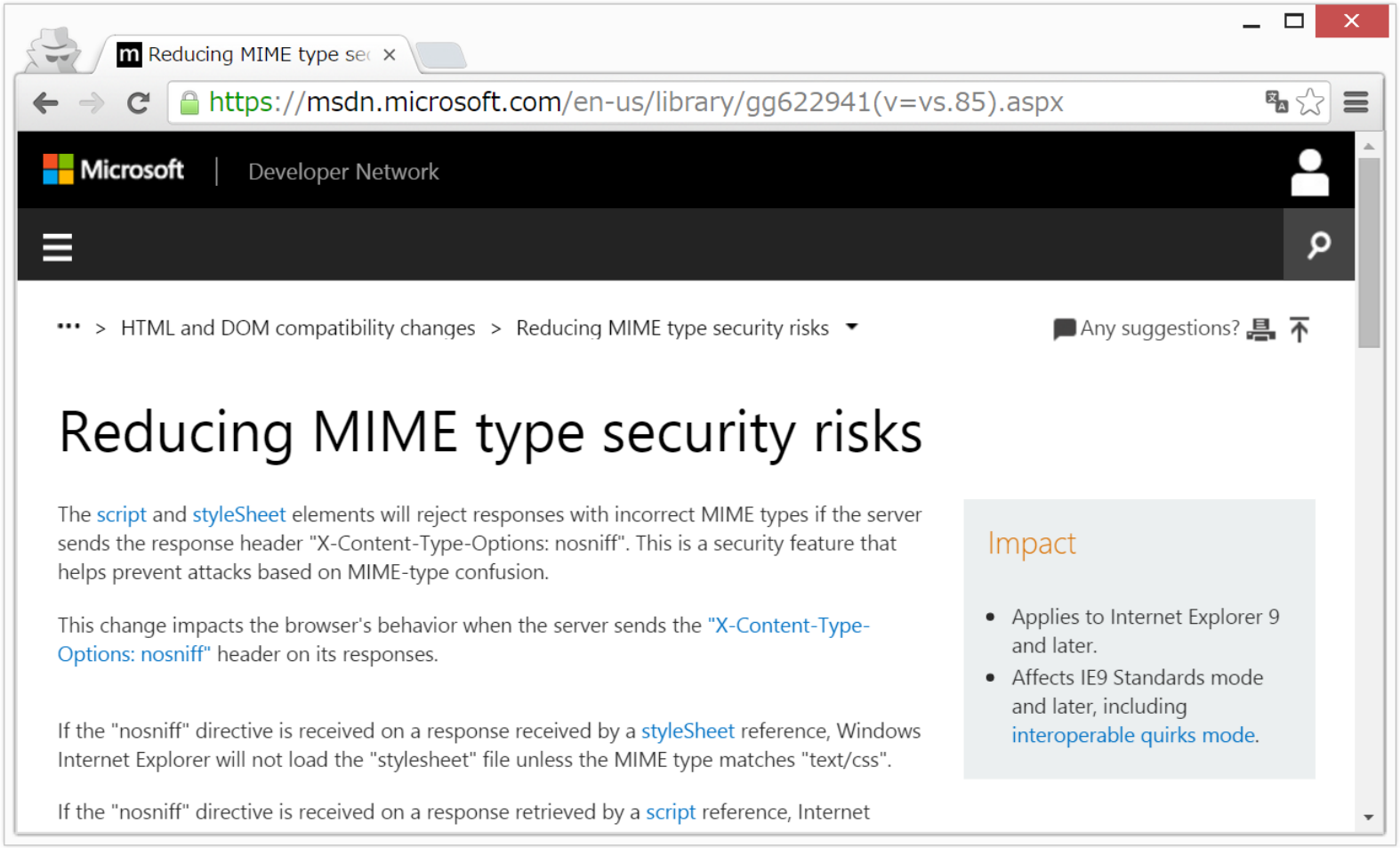
- 他のブラウザでは簡単な代替手段はない
  - DOMPurify などの外部ライブラリ
  - HTML5 iframe sandbox
- 標準化もされていないのであまり使われていない
  - 便利なはずなのに…。

IE8のここがすごい その4

X-Content-Type-Options

X-Content-Type-Options

- 返されたContent-Typeに従ってコンテンツを取り扱う。テキストなどをHTML扱いしない。



X-C-T-Oここがすごい

- Content-Typeに従うようになる
  - テキストファイルなどでXSSが発生しない

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Sat, 16 Jan 2016 12:34:56 GMT
X-Content-Type-Options: nosniff

これはテキストファイルです。
<script>alert(1);</script>
```

X-C-T-Oここがすごい

- Content-Typeに従うようになる

  …それって普通じゃん？

- 過去との互換性維持が必要なため、普通の挙動にするために特別なヘッダを導入する必要があった

X-C-T-Oここがすごい(余談)

- IE9以降ではスクリプト、CSSの読み込みも制限される
  - 以下のような場合、JSとしては読み込まれない

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Sat, 16 Jan 2016 12:34:56 GMT
X-Content-Type-Options: nosniff

document.addEventListener(
    "DOMContentLoaded",
    function(){ ... }
);
```

  - JSやCSSを通じた情報漏えいの防止に。

https://msdn.microsoft.com/en-us/library/gg622941(v=vs.85).aspx

IE8のここがすごい その5

X-Frame-Options

X-Frame-Options

- クリックジャッキング対策
  - 自ページのフレーム内への埋め込みを禁止する

X-Frame-Optionsここがすごい

- レスポンスヘッダに含めておくことで、自ページのフレームへの埋め込みが禁止できる
  - クリックジャッキングへの対応

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Date: Sat, 16 Jan 2016 12:34:56 GMT
X-Frame-Options: DENY
```

  - DENY、SAMEORIGIN、ALLOW-FROM url のいずれか

X-Frame-Optionsここがすごい

- 特に大きな副作用もない
- X-F-Oを付けたからといって特別な問題が発生するということもない
- 攻撃者的には全くおもしろくない機能

まとめ

まとめ

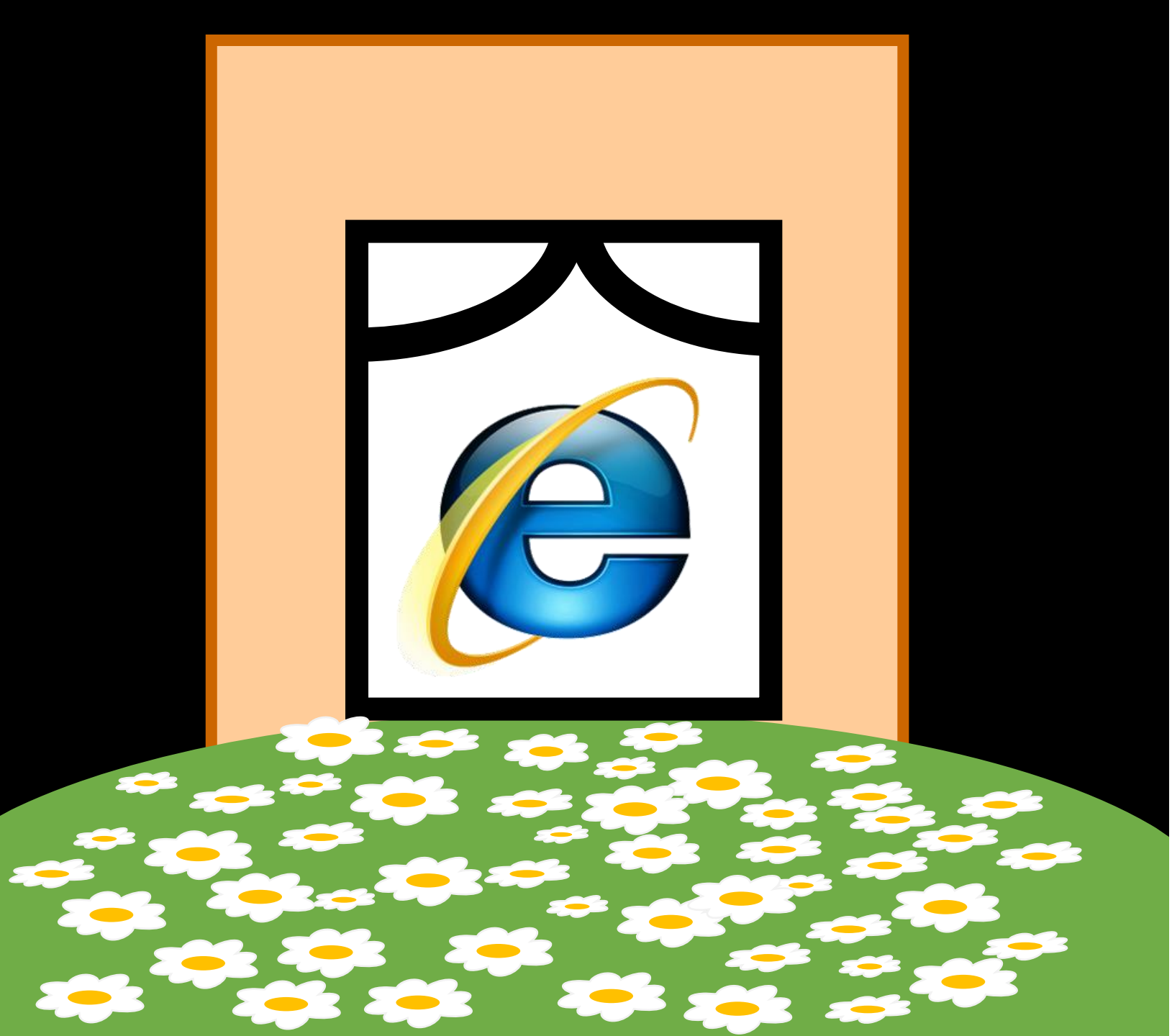
- IE8は、アグレッシブにセキュリティ強化のための機能を含めていた
  - 互換性を犠牲にするような変更も含む
- Web標準化路線への転換前夜

まとめ

- 標準化されたり他のブラウザにも取り込まれたり
  - XSSフィルター → XSS Auditor、NoScript、CSP
  - XDomainRequest → XMLHttpRequest
  - toStaticHTML → 代替なし
  - X-Content-Type-Options → Chromeも導入
  - X-Frame-Options → 他ブラウザも導入、CSP

まとめ

- Internet Explorer 8は新しい流れを確かに作った
  - 機能不足ばかり叫ばれ忌み嫌われていたが、実際には互換性を犠牲にしても挑戦的な機能をたくさん実装していた
- 僕たちはIE8のことを忘れない！知らんけど。



さよならあいいい8