

# Outbound Port 80 blocking のご提案

サイボウズ・ラボ株式会社

竹迫 良範

<takesako@shibuya.pm.org>

# JANOG31 Meeting in ROPPONGI

2013.01.24-25

## Outbound Port 80 blockingのご提案

### 概要

2012年11月にIETFで策定されたRFC 6797 - HTTP Strict Transport Security のプロトコルが誕生した背景と現在のブラウザの実装状況について解説します。

2010年10月にEric Butler氏によって公開されたFiresheepのデモでは、無線LAN上に流れているパケットを解析して、そこに流れているFacebookやTwitterの平文のHTTP Cookieの情報を盗み取り、マウスのクリックだけで簡単にセッションハイジャックできる様子が示されました。

安心してWebブラウジングを行えるインターネット接続環境としてユーザー様への「OP80B」メニューの提供をご提案いたします。

### 発表者

竹迫 良範 (サイボウズ・ラボ株式会社)

<http://www.janog.gr.jp/meeting/janog31/program/OP80B.html>

## [固有名詞] ドヤリング とは？

- スタバでMacBook Airをドヤ顔でいじる行為
- ドヤラー[名]
  - ドヤリングを行っている人
- ドヤリズム[名]
  - ドヤリングを取り入れたライフスタイル、思想
- ドヤリスト[名]
  - ドヤリズムを信奉し、また実践する人
- ドヤる[動ラ五]
  - ドヤリングをする（俗語）

# 本場のドヤリング（海外の上級者）



# ドヤ顔で Wireshark の画面を眺める人 ???

Capturing from Microsoft: %Device%NPF\_{C3F28BD5-A06E-4F87-A576-C0D782B85D24} [Wireshark 1.8.2 (SVN Rev 44520 from /trunk-1.8)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
9563	1224.211731000	192.168.0.1	239.255.255.250	SSDP	310	NOTIFY * HTTP/1.1
9567	1224.521872000	192.168.0.8	69.171.229.16	HTTP	719	POST /ajax/banzai/banzai.php HTTP/1.1 (application/x-www-form-urlencoded)
9569	1224.681535000	69.171.229.16	192.168.0.8	HTTP	983	HTTP/1.1 200 OK (application/x-javascript)
9603	1240.999802000	192.168.0.8	69.171.229.16	HTTP	561	GET / HTTP/1.1
9622	1241.393401000	69.171.229.16	192.168.0.8	HTTP	292	HTTP/1.1 200 OK (text/html)
9660	1254.175708000	192.168.0.1	239.255.255.250	SSDP	372	NOTIFY * HTTP/1.1

Internet Protocol Version 4, Src: 192.168.0.8 (192.168.0.8), Dst: 69.171.229.16 (69.171.229.16)

Transmission Control Protocol, Src Port: 49657 (49657), Dst Port: http (80), Seq: 666, Ack: 930, Len: 507

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

User-Agent: Opera/9.80 (Windows NT 6.1; WOW64; u; ja) Presto/2.10.289 Version/12.02\r\n

Host: www.facebook.com\r\n

Accept: text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/webp, image/jpeg, image/gif, image/x-bitmap, \*/\*;q=0.1\r\n

Accept-Language: ja-JP,ja;q=0.9,en;q=0.8\r\n

Accept-Encoding: gzip, deflate\r\n

Cookie: datr=WB9LUACppalGegtMseytB9Mq; reg\_fb\_gate=http%3A%2F%2Fwww.facebook.com%2F; reg\_fb\_ref=http%3A%2F%2Fwww.facebook.com%2F; wd=1205x628\r\n

Connection: Keep-Alive\r\n

\r\n

[Full request URI: <http://www.facebook.com/>]

```

0150  61 2d 4a 50 2c 6a 61 3b 71 3d 30 2e 39 2c 65 6e  a-JP,ja; q=0.9,en
0160  3b 71 3d 30 2e 38 0d 0a 41 63 63 65 70 74 2d 45  ;q=0.8.. Accept-E
0170  6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64  ncoding: gzip, d
0180  65 66 6c 61 74 65 0d 0a 43 6f 6f 6b 69 65 3a 20  eflate.. Cookie:
0190  64 61 74 72 3d 57 42 39 4c 55 41 43 50 70 61 4c  datr=WB9 LUACppal
01a0  67 45 67 74 4d 73 65 79 74 42 39 4d 71 3b 20 72  gEgtMsey tB9Mq; r
01b0  65 67 5f 66 62 5f 67 61 74 65 3d 68 74 74 70 25  eg_fb_ga te=http%
01c0  33 41 25 32 46 25 32 46 77 77 77 2e 66 61 63 65  3A%2F%2F www.face
01d0  62 6f 6f 6b 2e 63 6f 6d 25 32 46 3b 20 72 65 67  book.com %2F; reg
01e0  5f 66 62 5f 72 65 66 3d 68 74 74 70 25 33 41 25  _fb_ref= http%3A%
01f0  32 46 25 32 46 77 77 77 2e 66 61 63 65 62 6f 6f  2F%2Fwww .faceboo
0200  6b 2e 63 6f 6d 25 32 46 3b 20 77 64 3d 31 32 30  k.com%2F ; wd=120
0210  35 78 36 32 38 0d 0a 43 6f 6e 6e 65 63 74 69 6f  5x628..C onnectio
0220  6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 0d 0a 0d  n: Keep- Alive...
0230  0a

```

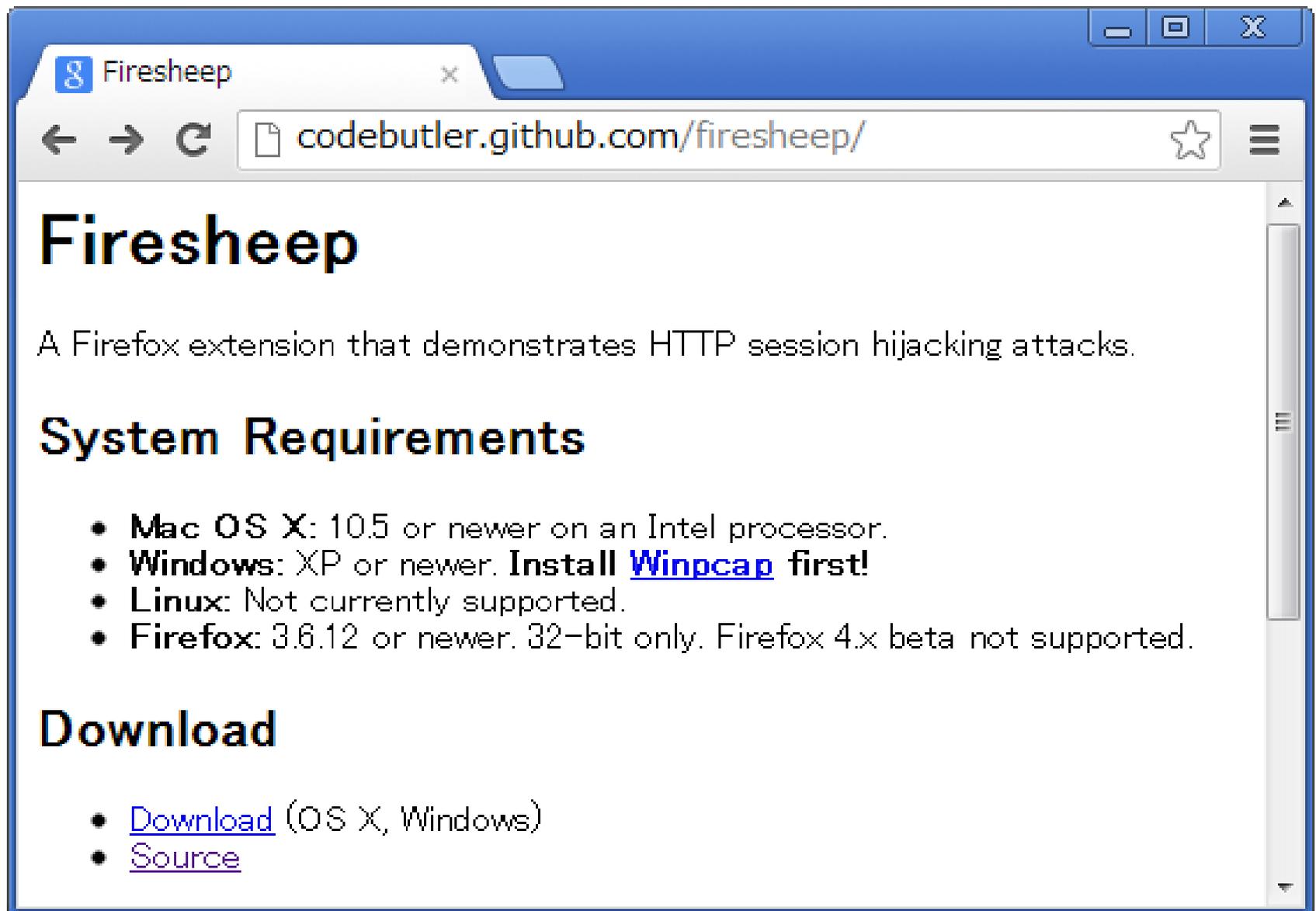
HTTP Cookie (http.cookie), 143 bytes      Packets: 10493 Displayed: 1562 Marked: 0      Profile: Default



> 突然のFiresheep <

# Firesheepとは

- 2010年10月
  - Firefoxの拡張として公開
- 開発者
  - Eric Butler氏
- 無線LAN上に流れているパケットを解析して、そこに流れているfacebookやTwitterの平文のHTTP Cookieの情報を盗み取り、マウスのクリックだけでセッションハイジャックする
- インストールがとても簡単
  - PoCとしてツールを公開し現実的な脅威を示した



Firesheep

codebutler.github.com/firesheep/

# Firesheep

A Firefox extension that demonstrates HTTP session hijacking attacks.

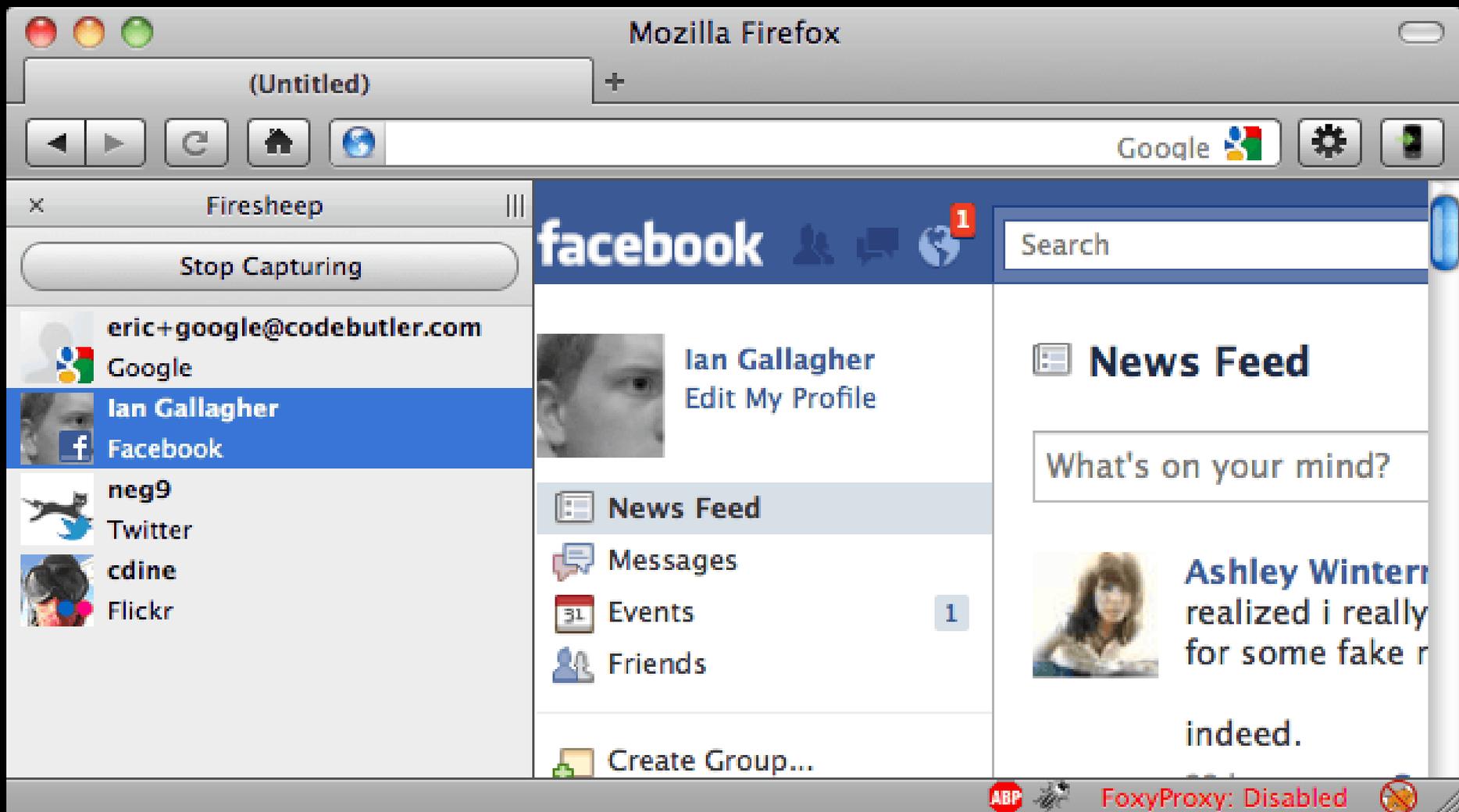
## System Requirements

- **Mac OS X:** 10.5 or newer on an Intel processor.
- **Windows:** XP or newer. **Install [Wingcap](#) first!**
- **Linux:** Not currently supported.
- **Firefox:** 3.6.12 or newer. 32-bit only. Firefox 4.x beta not supported.

## Download

- [Download](#) (OS X, Windows)
- [Source](#)

# 【参考】 Firesheep の動作画面



# Firesheepがデフォルトで対象とするサービス

- アカウントを乗っ取る対象のWebサービス
  - Amazon.com、CNET、dropbox、Evernote、Facebook、Flickr、Github、Google、WindowsLive、NY Times、tumblr、Twitter、WordPress、Yahoo . . .
- マウスのクリック操作だけで
  - 簡単になりすましができる
- 大手Webサービス運営各社に衝撃
  - → \突然のSSL化/

## github.com \ 突然のSSL化 /

GitHub, Inc. [US] <https://github.com>

github Search... Explore Gist Blog Help takesako

takesako

News Feed Pull Requests Issues

GitHub Bootcamp If you are still new to things, we've provided a few walkthroughs to get you started.

Elements Resources Network Sources Timeline Profiles Audits Console Search N

Name	Method	Status	Type	Initiator	Size	Time	Timeline		
Path		Text			Content	Latency		1.81s	2.72s
<a href="http://github.com/">http://github.com/</a>	GET	(pending)	Pending	Other	13B 0B	1ms 0.0 days			
<a href="https://github.com/">https://github.com/</a>	GET	200 OK	text/html	<a href="http://github.com/">http://github.com/</a>	9.70KB 49.89KB	254ms 0.50ms			

# Set-Cookieにsecureフラグが付くようになった

- (例) <https://github.com/> からのレスポンス

```
HTTP/1.1 200 OK
```

```
Server: nginx
```

```
Date: Fri, 14 Sep 2012 09:39:57 GMT
```

```
Content-Type: text/html; charset=utf-8
```

```
Transfer-Encoding: chunked
```

```
Connection: keep-alive
```

```
Status: 200 OK
```

```
Cache-Control: private, max-age=0, must-revalidate
```

```
Strict-Transport-Security: max-age=2592000
```

```
X-Frame-Options: deny
```

```
Set-Cookie: _gh_sess=XYZ; path=/; secure; HttpOnly
```



# Strict-Transport-Security ヘッダも出力！！

- (例) <https://github.com/> からのレスポンス

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 14 Sep 2012 09:39:57 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Status: 200 OK
Cache-Control: private, max-age=0, must-revalidate
Strict-Transport-Security: max-age=2592000
X-Frame-Options: deny
Set-Cookie: _gh_sess=XYZ; path=/; secure; HttpOnly
```



# HTTP Strict Transport Security (HSTS)

[tools.ietf.org/html/rfc6797](https://tools.ietf.org/html/rfc6797)

PROPOSED STANDARD

Internet Engineering Task Force (IETF)  
Request for Comments: 6797  
Category: Standards Track  
ISSN: 2070-1721

J. Hodges  
PayPal  
C. Jackson  
Carnegie Mellon University  
A. Barth  
Google, Inc.  
November 2012

## HTTP Strict Transport Security (HSTS)

### Abstract

This specification defines a mechanism enabling web sites to declare themselves accessible only via secure connections and/or for users to be able to direct their user agent(s) to interact with given sites only over secure connections. This overall policy is referred to as HTTP Strict Transport Security (HSTS). The policy is declared by web sites via the Strict-Transport-Security HTTP response header field and/or by other means, such as user agent configuration, for example.

# Strict-Transport-Securityヘッダで指定できる値

## ■ max-age=有効期限（秒）

- ブラウザがそのサイトに対してHTTPS通信のみでアクセスする有効期限を秒単位で指定
- 今後1年間のhttp://へのアクセスをhttps://に切り替えるには $365 \times 24 \times 3600 = 31536000$ を指定

```
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 14 Sep 2012 09:39:57 GMT
Content-Type: text/html; charset=utf-8
Cache-Control: private, max-age=0, must-revalidate
Strict-Transport-Security: max-age=31536000; includeSubdomains
X-Frame-Options: deny
Set-Cookie: _gh_sess=XYZ; path=/; secure; HttpOnly
```

## 【例】 Apache の httpd.conf の設定

```
<VirtualHost *:80>
# サイト上のすべてのHTTPアクセスをhttps://にリダイレクトする
ServerAlias *
RewriteEngine On
RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

<VirtualHost _default_:443>
# HTTPSサイト上でStrict-Transport-Securityヘッダを出力する
Header set Strict-Transport-Security ¥
      "max-age=31536000; includeSubDomains"

# SSL関連の設定
SSLEngine on
# ...
</VirtualHost>
```

## 【例】 Apache の httpd.conf の設定

```
<VirtualHost *:80>
# サイト上のすべてのHTTPアクセスをhttps://にリダイレクトする
ServerAlias *
RewriteEngine On
RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

<VirtualHost _default_:443>
# HTTPSサイト上でStrict-Transport-Securityヘッダを出力する
Header set Strict-Transport-Security ¥
           "max-age=31536000; includeSubDomains"

# SSL関連の設定
SSLEngine on
# ...
</VirtualHost>
```

# WebサイトをすべてSSL化するときは・・・

1. Webサイト上の通信をすべてHTTPSに変更
2. Set-Cookie時にsecure属性をつける
  - HTTPでの盗聴を防止（SSL通信時のみ送信）
3. Set-Cookie時にHttpOnly属性をつける
  - JavaScriptから読み取りを禁止（XSS被害を低減）
4. CookieのIDをSSL専用に変えておく
  - ses\_id => sess\_id（同じIDを使わない）
5. Strict-Transport-Securityヘッダを出力する
  - RFC 6797 (HSTS) を参考に

欠点

# HSTS 対応ブラウザ

- Google Chrome 4.0.211.0以降
- Firefox 4以降
- Opera 12以降
  
- HSTS未対応のブラウザ
  - Internet Explorer
  - Internet Explorer
  - Internet Explorer
    - 大事なことなので 3 回 . . .

ISP様へ  
ご提案

# Outbound Port80 Blocking (HTTP)

# Outbound Port 80 blocking (OP80B) のご提案

- HTTP (80) 通信をすべてブロック
  - HTTP は信頼できない通信経路を通ることがある
    - 盗聴、DNS汚染、中間者攻撃、ARP spoofing . . .
  - 児童ポルノも完全遮断（リストの管理必要なし）
- 信頼できる SSL/TLS (443) 通信のみを許可
  - Gmail、Google検索、Twitter、facebook、
    - 大手Webサイトは既に https:// に対応している
- WebサイトのSSL化を促す
  - SSLワイルドカード証明書 (\*.example.com)
  - SPDY、HTTP/2.0の登場でTLS通信が重要に

利用者  
として

# ブラウザのアドレスバー

Firefox ▼

新しいタブ



URL を入力します

何を入力していますか？

Firefox ▼

新しいタブ



facebook.com

何を入力していますか？

Firefox ▼

新しいタブ



twitter.com

JANOGメンバーならhttps://を入力しましょう

Firefox ▼

新しいタブ



<https://twitter.com>

