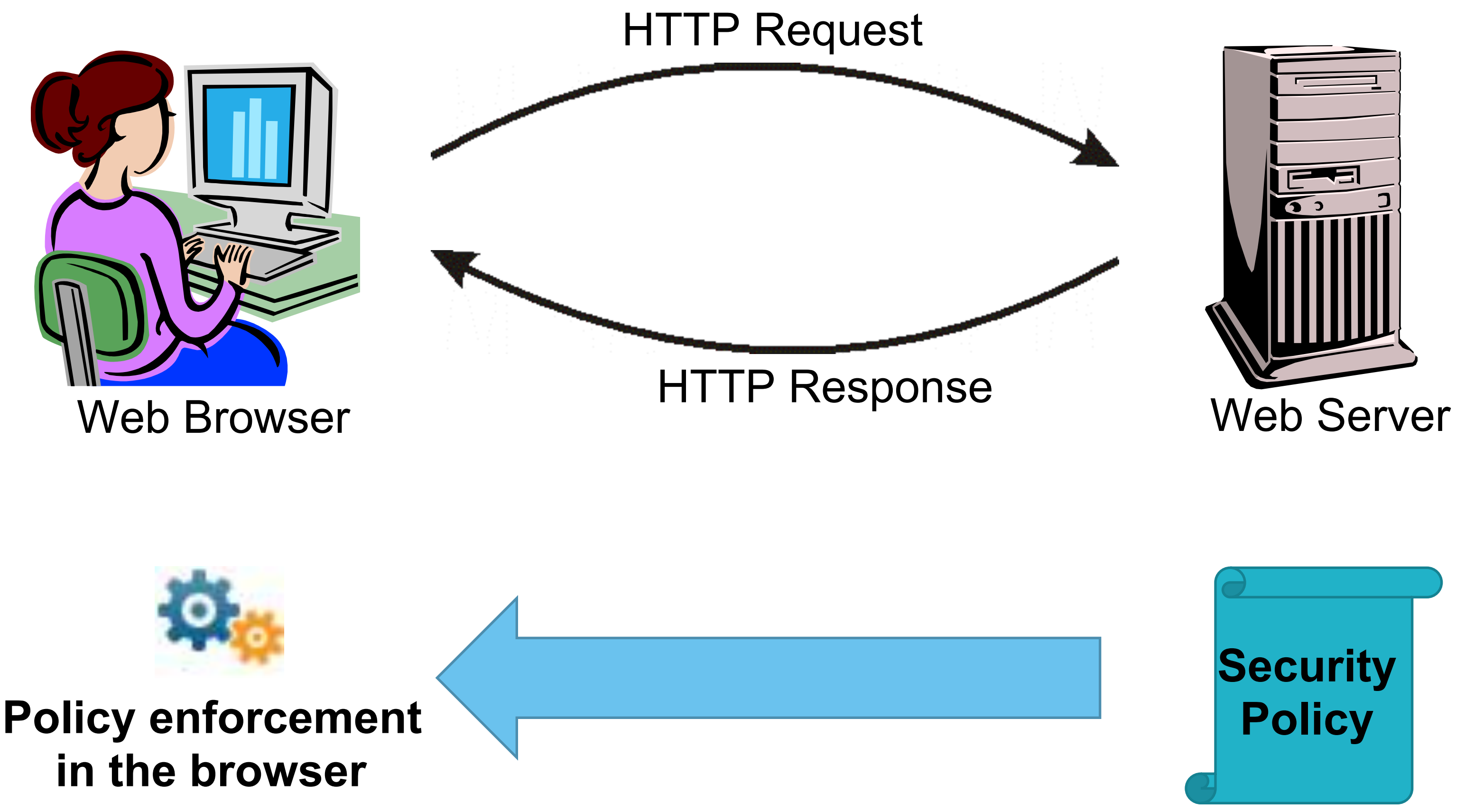
Securing Rich Internet Applications: Overview of best practices

Lieven Desmet – iMinds-DistriNet, KU Leuven
Lieven.Desmet@cs.kuleuven.be

Recent security technology on the web



Overview

- Basic security model of the Web
- #1 Securing browser-server communication
- #2 Mitigating script injection attacks
- #3 Framing content securely
- Wrap-up

Basic security model of the web

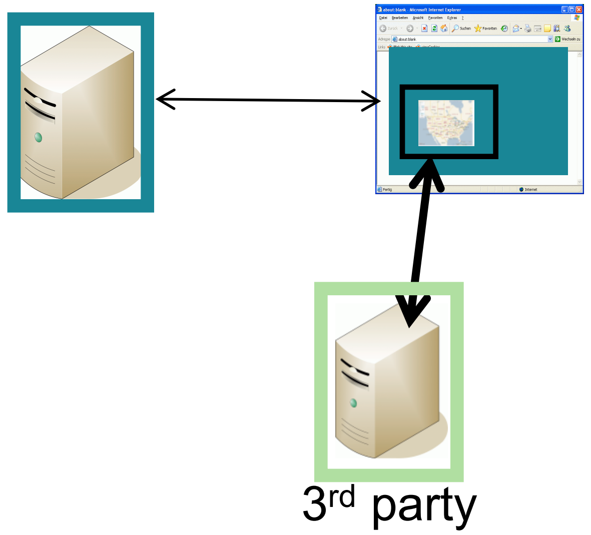
Introduction

- Basic security policy for the web:
  - Same-Origin Policy
- What does it mean for scripts running on your page?
- What does it mean for frames included in your page?

Two basic composition techniques

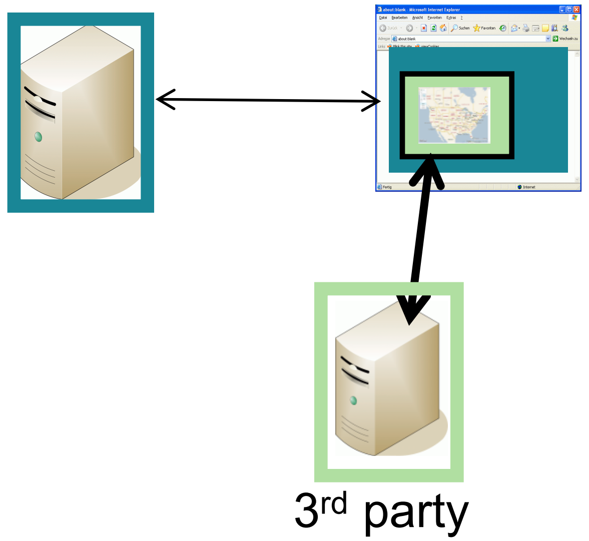
Script inclusion

```
<html><body>
…
<script src="http://3rdparty.com/script.js"></script>
…
</body></html>
```



Iframe integration

```
<html><body>
…
<iframe src="http://3rdparty.com/frame.html"></iframe>
…
</body></html>
```

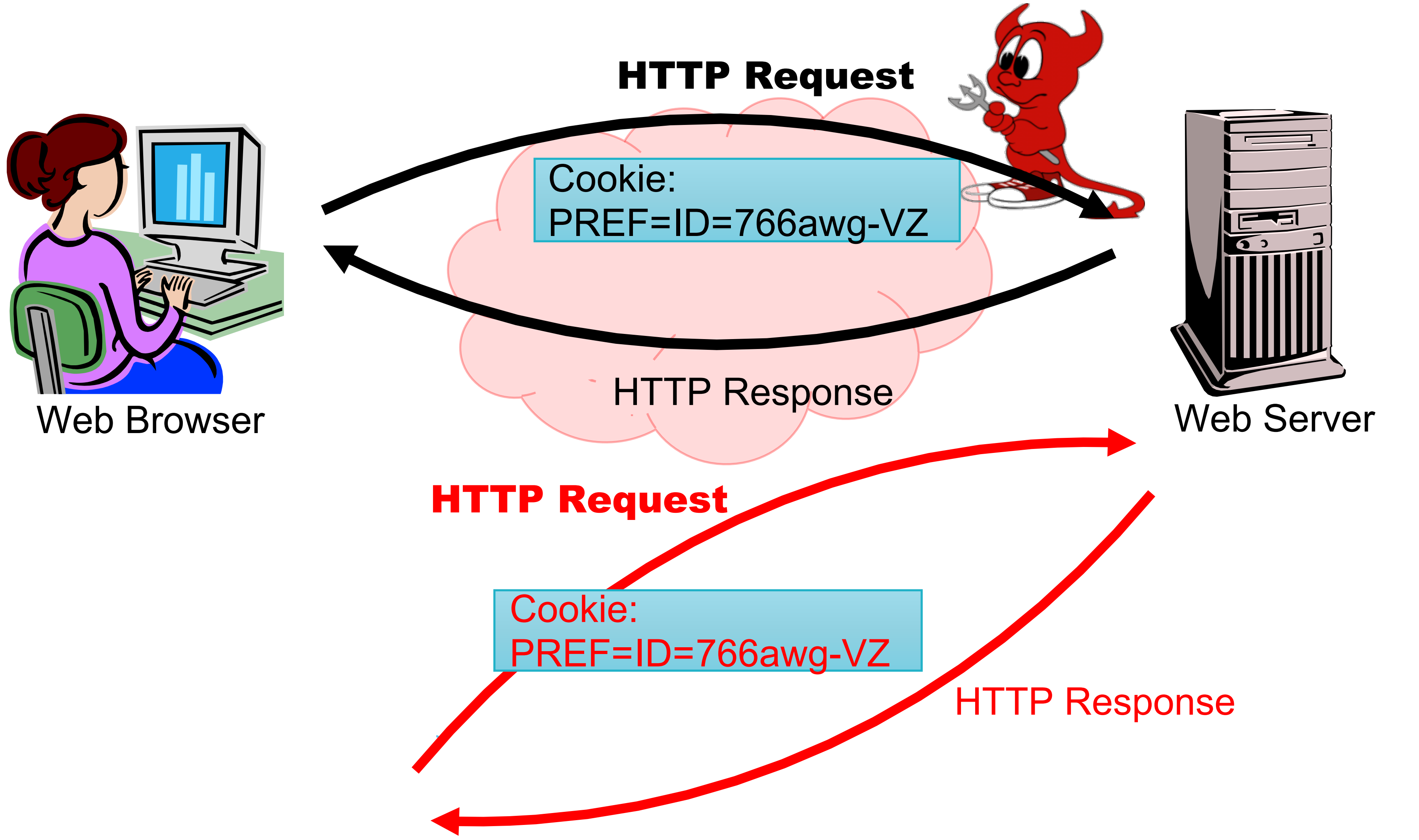


Securing browser-server communication

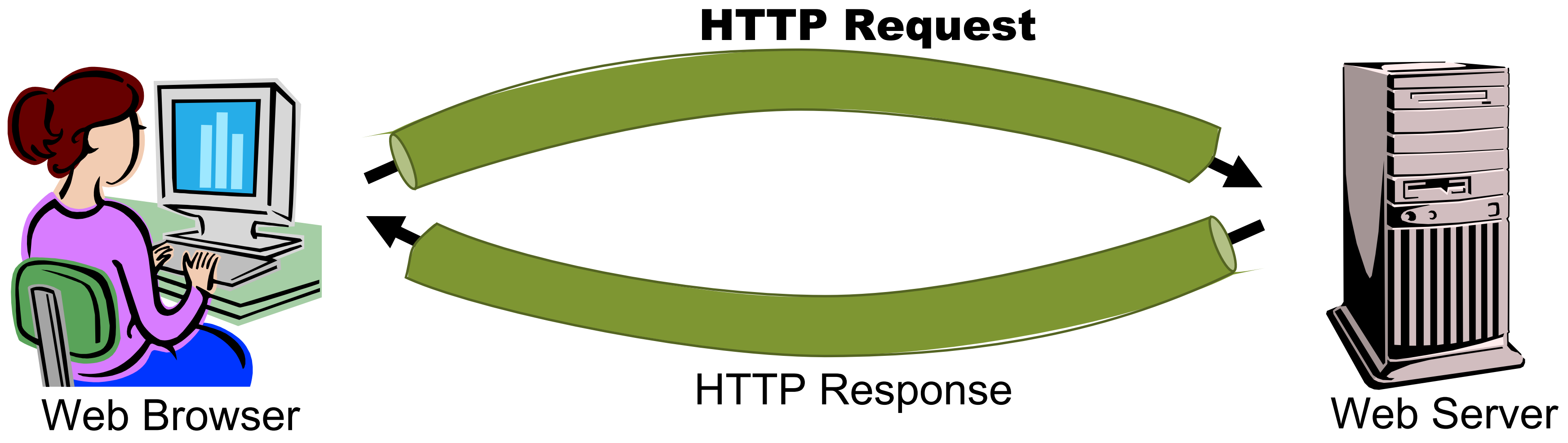
Overview

- Attacks:
  - Session hijacking
  - SSL Stripping
- Countermeasures:
  - Use of SSL/TLS
  - Secure flag for session cookies
  - HSTS header
  - Public Key Pinning

Network attacks: Session hijacking



HTTPS to the rescue…



Problem cured?

- TLS usage statistics:
  - 0.78% of active domains use TLS (with valid SSL certificate)
  - For Alexa top 1 million: 27.86% use TLS

Internet SSL Survey 2010, Qualys

- Remaining problems:
  - Mixed use of HTTPS/HTTP and session cookies
  - Mixed content websites
  - SSL Stripping attacks

Mixed use of HTTPS/HTTP

- Cookies are bound to domains, not origins
- By default, cookies are sent both over HTTPS and HTTP
- Any request to your domain over HTTP leaks the (session) cookies…

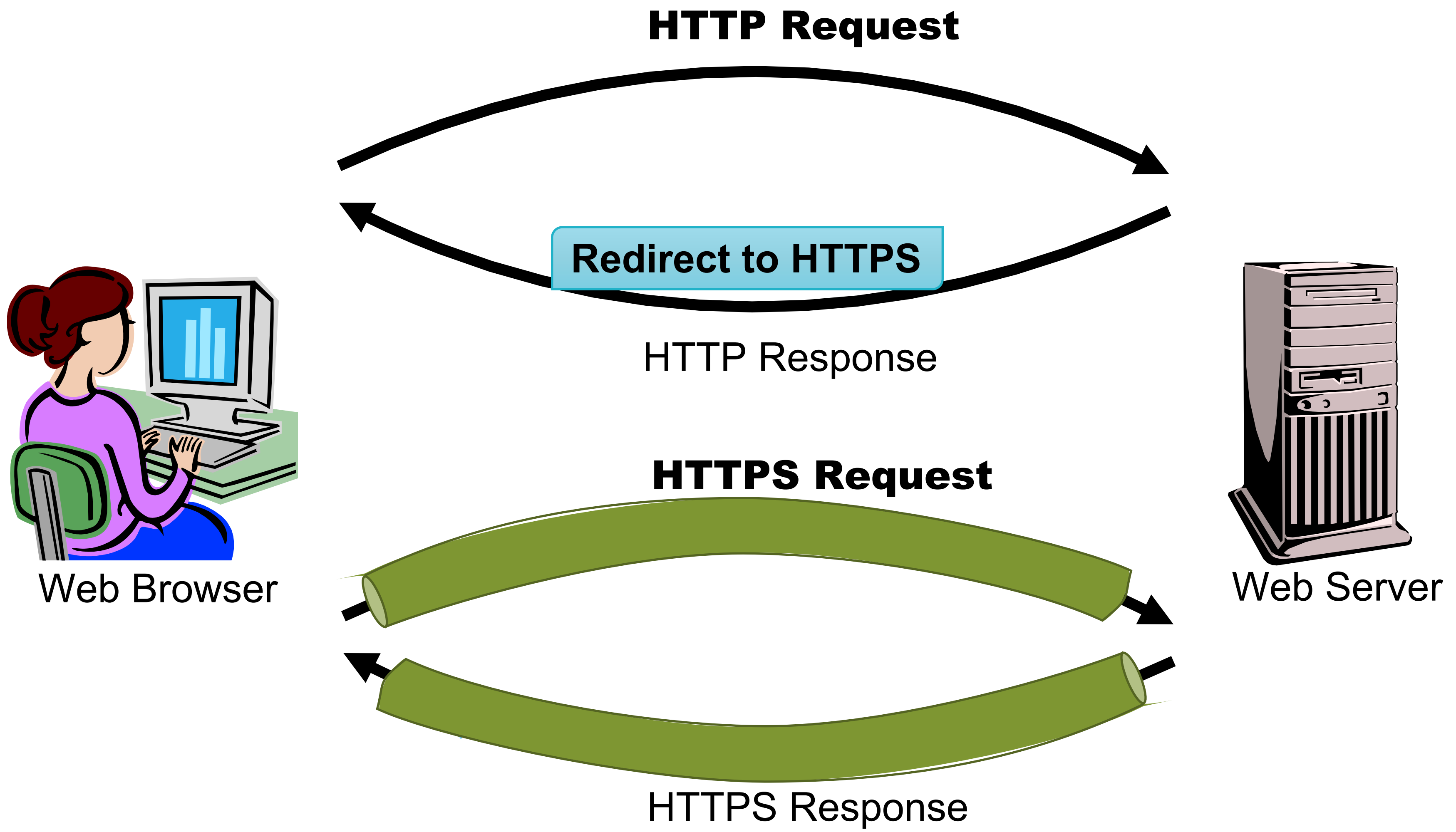
Secure flag for cookies

- Issued at cookie creation (HTTP response)
  - Set-Cookie: PREF=766awg-VZ; Domain=yourdomain.com; Secure

- If set, the cookie is only sent over an encrypted channel

- Should be enabled by default for your session cookies!

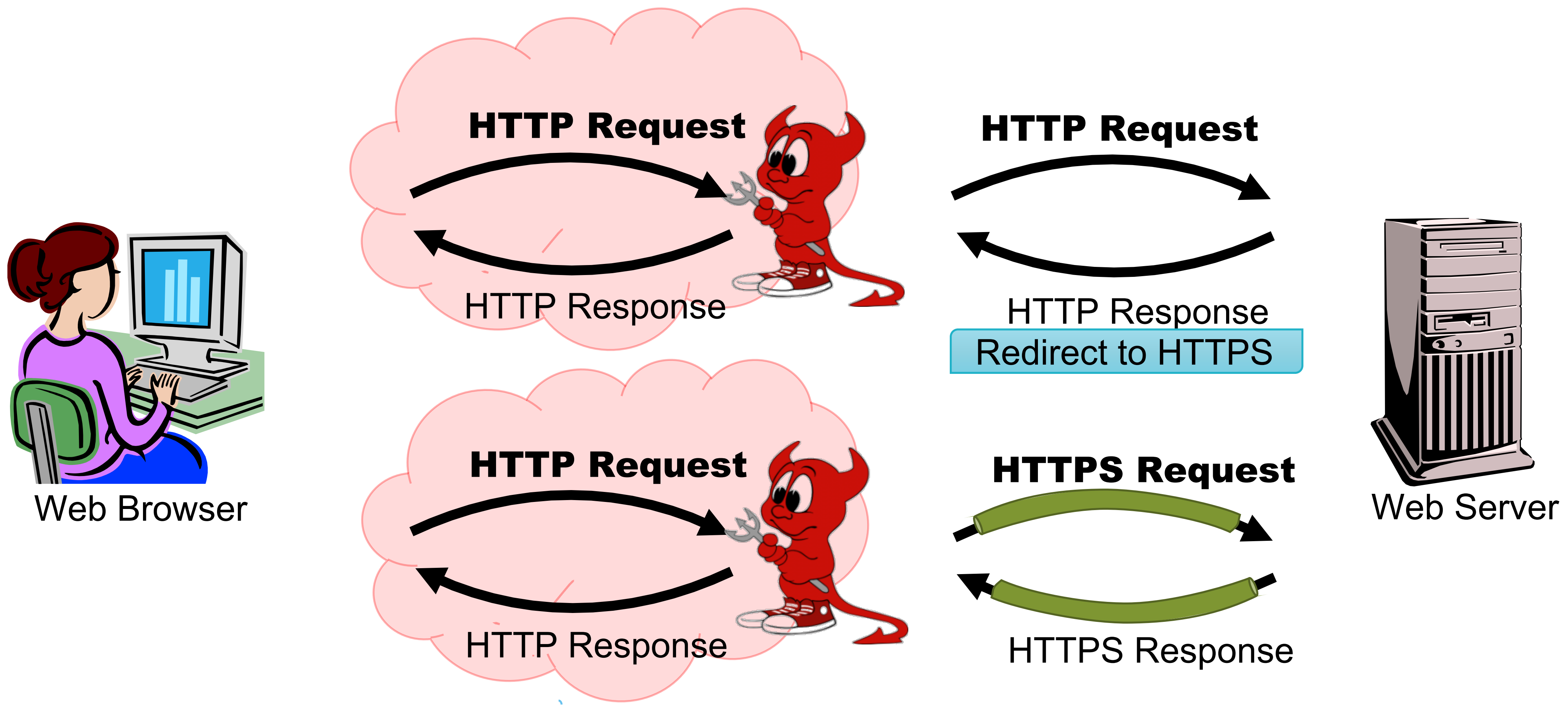
HTTP to HTTPS bootstrapping



HTTP to HTTPS bootstrapping

- HTTP 301/302 response
  - Location header redirects browser to the resource over HTTPS
  - Location: https://mysite.com/
- Meta refresh
  - Meta-tag in HEAD of HTML page
  - <meta http-equiv="refresh" content="0;URL='https://mysite.com/'">
- Via JavaScript
  - document.location = “https://mysite.com”

Network attacks: SSL Stripping



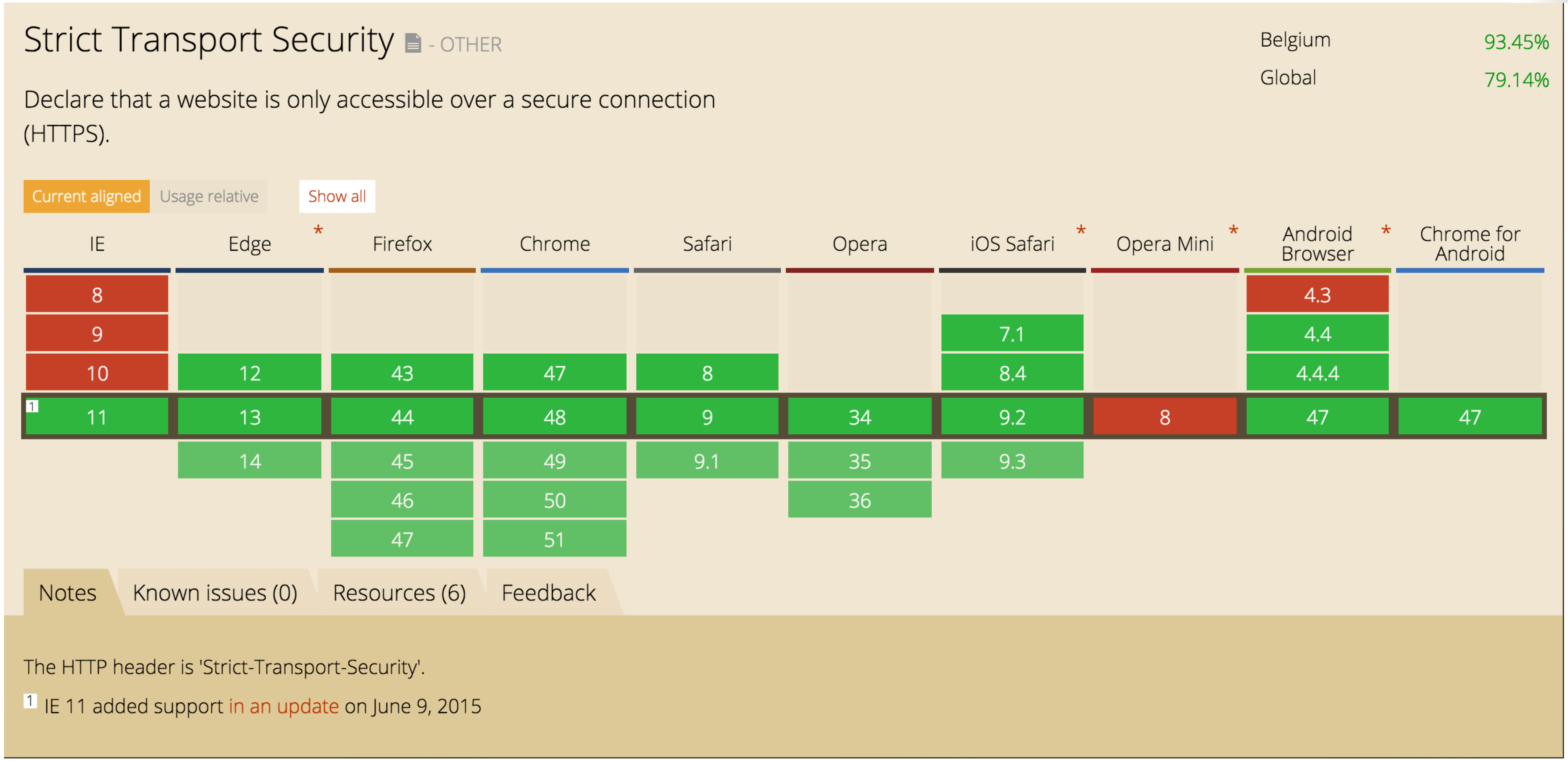
Moxie Marlinspike, BlackHat DC 2009

Strict Transport Security (HSTS)

- Issued by the HTTP response header
  - Strict-Transport-Security: max-age=60000
- If set, the browser is instructed to visit this domain only via HTTPS
  - No HTTP traffic to this domain will leave the browser
- Optionally, also protect all subdomains
  - Strict-Transport-Security: max-age=60000; includeSubDomains

DistriNet iMinds KU LEUVEN

HSTS: state-of-practice



Strict Transport Security - OTHER

Belgium 93.45%

Global 79.14%

Declare that a website is only accessible over a secure connection (HTTPS).

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
🟥 8								🟥 4.3	
🟥 9						🟩 7.1		🟩 4.4	
🟥 10	🟩 12	🟩 43	🟩 47	🟩 8		🟩 8.4		🟩 4.4.4	
🟩 [1] 11	🟩 13	🟩 44	🟩 48	🟩 9	🟩 34	🟩 9.2	🟥 8	🟩 47	🟩 47
	🟩 14	🟩 45	🟩 49	🟩 9.1	🟩 35	🟩 9.3			
		🟩 46	🟩 50		🟩 36				
		🟩 47	🟩 51						

Notes | Known issues (0) | Resources (6) | Feedback

The HTTP header is 'Strict-Transport-Security'.

[1] IE 11 added support in an update on June 9, 2015

But can I trust the CAs ?

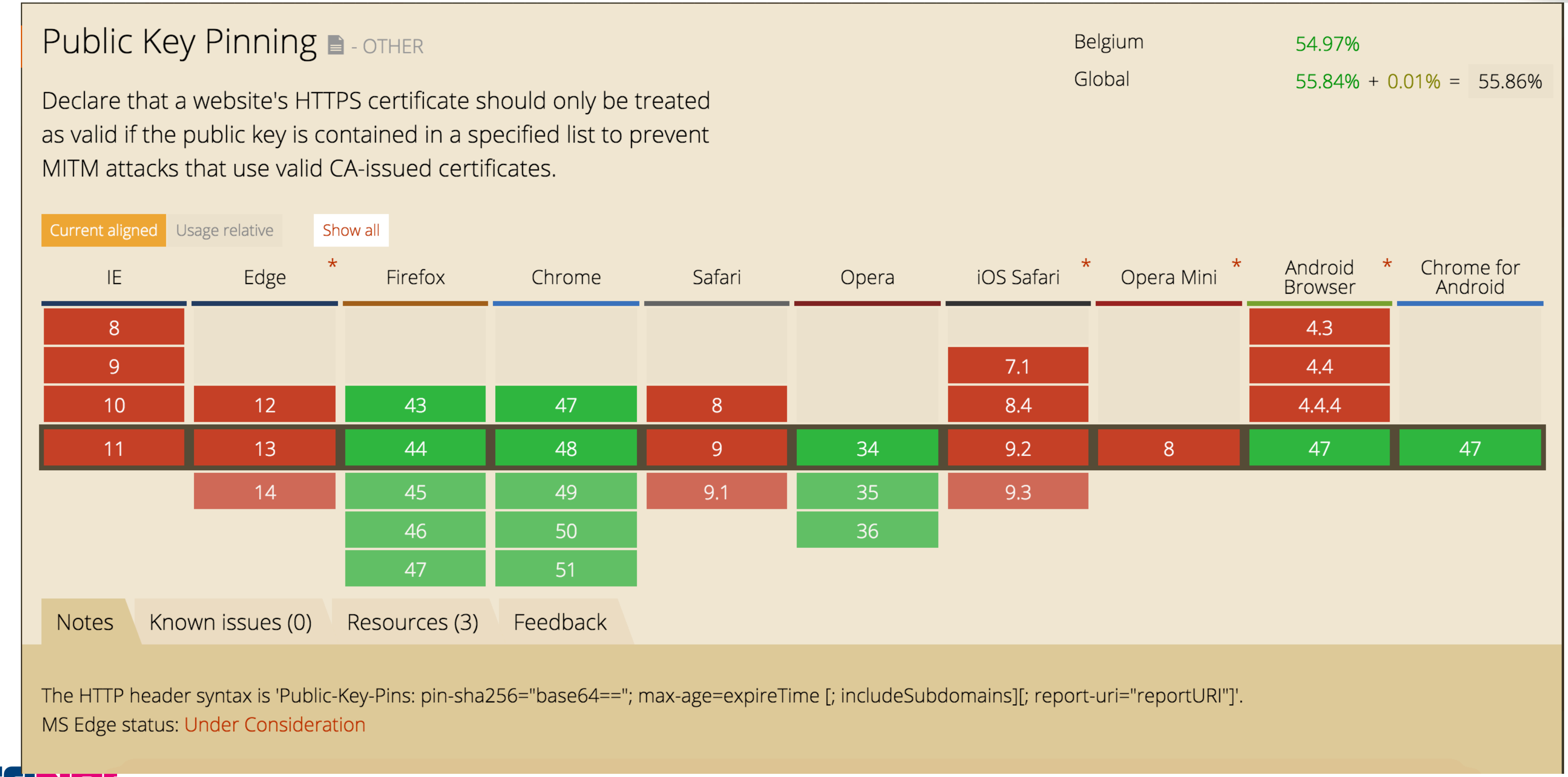
- Comodo (March 2011)
  - 9 fraudelent SSL certificates
- Diginotar (July 2011)
  - Wildcard certificates for Google, Yahoo!, Mozilla, WordPress, …
- Breaches at StartSSL (June 2011) and GlobalSign (Sept 2012) reported unsuccessful
- …

Public Key Pinning (HPKP)

- Issued as HTTP response header
  - Public-Key-Pins: max-age=500; pin-sha1="4n972HfV354KP560yw4uqe/baXc="; pin-sha1="IvGeLsbqzPxdI0b0wuj2xVTdXgc="
- Freezes the certificate by pushing a fingerprint of (parts of) the certificate chain to the browser
  - Options: max-age, includeSubdomains, report-uri

DistriNet iMinds KU LEUVEN

HPKP: state-of-practice



Public Key Pinning - OTHER

Declare that a website's HTTPS certificate should only be treated as valid if the public key is contained in a specified list to prevent MITM attacks that use valid CA-issued certificates.

Belgium	54.97%
Global	55.84% + 0.01% = 55.86%

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
🟥 8								🟥 4.3	
🟥 9						🟥 7.1		🟥 4.4	
🟥 10	🟥 12	🟩 43	🟩 47	🟥 8		🟥 8.4		🟥 4.4.4	
🟥 11	🟥 13	🟩 44	🟩 48	🟥 9	🟩 34	🟥 9.2	🟥 8	🟩 47	🟩 47
	🟥 14	🟩 45	🟩 49	🟥 9.1	🟩 35	🟥 9.3			
		🟩 46	🟩 50		🟩 36				
		🟩 47	🟩 51						

Notes Known issues (0) Resources (3) Feedback

The HTTP header syntax is 'Public-Key-Pins: pin-sha256="base64=="; max-age=expireTime [; includeSubdomains][; report-uri="reportURI"]'.
MS Edge status: Under Consideration

Recap: Securing browser-server communication

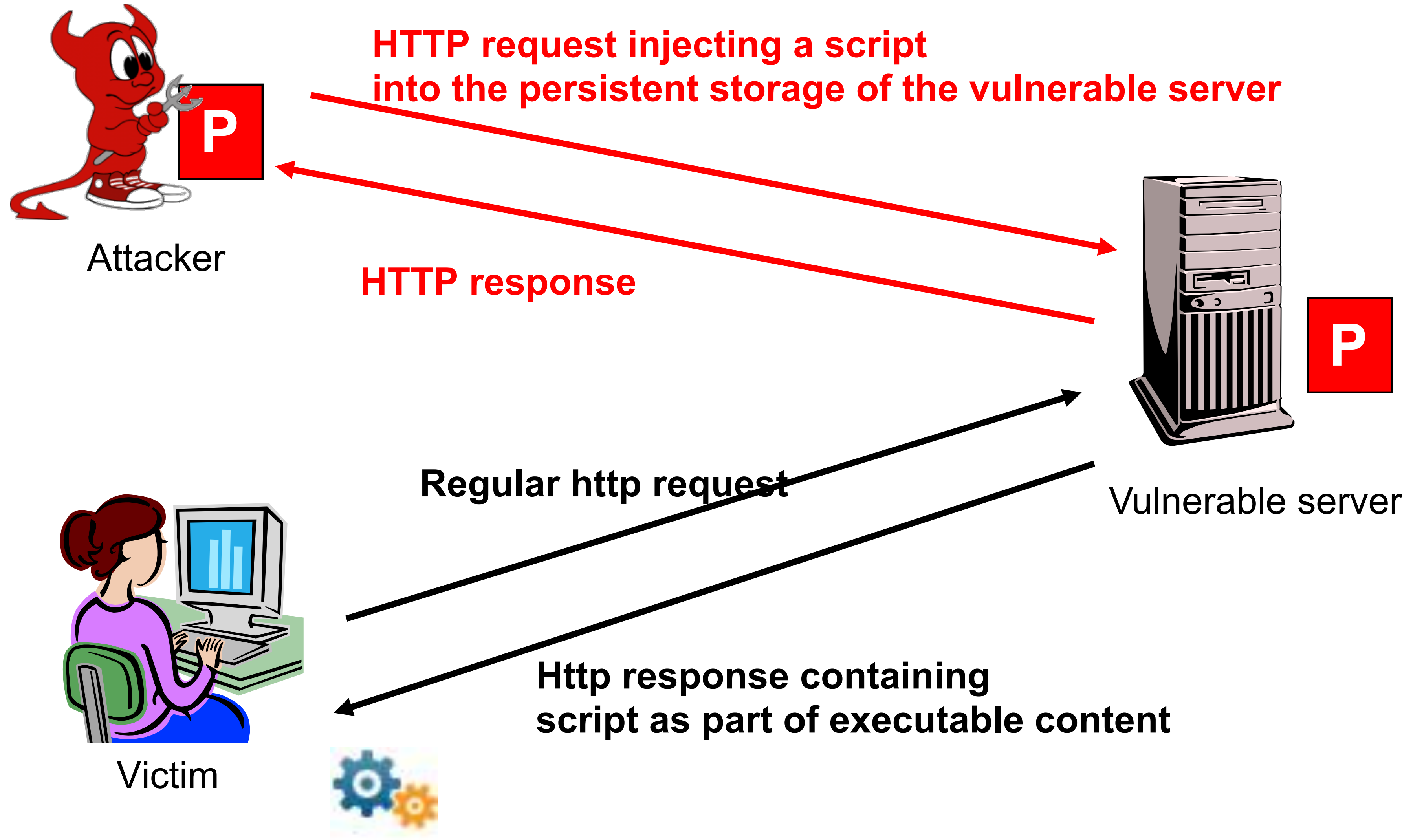
- Use of TLS
  - be aware of mixed-content inclusions!
- Secure flag for cookies
  - to protect cookies against leaking over HTTP
- HSTS header
  - to force TLS for all future connections
- Public Key Pinning
  - to protect against fraudulent certificates

#2 Mitigating script injection attacks

Overview

- Attack:
  - Cross-Site Scripting (XSS)
- Countermeasures:
  - HttpOnly flag for session cookies
  - Content Security Policy (CSP)
  - Subresrouce Integrity (SRI)

Example: Stored or persistent XSS


HTTP request injecting a script
into the persistent storage of the vulnerable server
P
Attacker
HTTP response
P
Vulnerable server
Regular http request
Http response containing
script as part of executable content
Victim

HttpOnly flag for cookies

- Issued at cookie creation (HTTP response)
  - Set-Cookie: PREF=766awg-VZ; Domain=yourdomain.com; Secure; HttpOnly
- If set, the cookie is not accessible via DOM
  - JavaScript can not read or write this cookie
- Mitigates XSS impact on session cookies
  - Protects against hijacking and fixation
- Should be enabled by default for your session cookies!

Content Security Policy (CSP)

- Issued as HTTP response header
  - Content-Security-Policy: script-src 'self'; object-src 'none'
- Specifies which resources are allowed to be loaded as part of your page
- Extremely promising as an additional layer of defense against script injection

CSP set of directives

- There are a whole set of directives
  - Here we discuss CSP v1.1 (February 11, 2014)
- default-src
  - Takes a sourcelist as value
  - Default for all resources, unless overridden by specific directives
  - Only allowed resources are loaded

CSP source lists

- Space delimited list of sources
  - ‘self’
  - ‘none’
  - origin(s)
- Examples
  - https://mydomain.com
  - https://mydomain.com:443
  - http://134.58.40.10
  - https://*.mydomain.com
  - https:
  - *://mydomain.com

CSP set of directives (2)

- script-src
  - From which sources, scripts are allowed to be included
- object-src
  - Flash and other plugins
- style-src
  - stylesheets
- img-src
  - images
- media-src
  - sources of video and audio

CSP set of directives (3)

- child-src
  - list of origins allowed to be embedded as frames
  - replaces the deprecated frame-src directive
- font-src
  - web fonts
- connect-src
  - To which origins can you connect (e.g. XHR, websockets)
- frame-options
  - Control framing of the page
- sandbox
  - Trigger sandboxing attribute of embeded iframes

CSP requires sites to “behave”

- Inline scripts and CSS is not allowed
  - All scripts need to be externalized in dedicated JS files
  - All style directives need to be externalized in dedicated style files
  - Clean code separation
- The use of *eval* is not allowed
  - To prevent unsafe string (e.g. user input) to be executed

Example: inline scripts

page.html

```
<script>
  function runMyScript() {
    alert('My alert');
  }
</script>

<a href="#" onClick="runMyScript();">
This link shows an alert!</a>
```

Example: externalized scripts

External JS

page.html

```
<script src="myscript.js"></script>
<a href="#" id="myLink">This link shows an alert!</a>
```

JavaScript code

Binding to page

myscript.js

```
function runMyScript() {
  alert('My alert');
}
document.addEventListener('DOMContentReady',
function () {
  document.getElementById('myLink')
          .addEventListener('click', runMyScript);
});
```

Insecure relaxations, but be careful!

- To temporary allow inline scripts
  - Content-Security-Policy: script-src 'self' 'unsafe-inline'
- To temporary allow eval
  - Content-Security-Policy: script-src 'self' 'unsafe-inline' 'unsafe-eval'
- To temporary allow inline style directives
  - Content-Security-Policy: style-src 'self' 'unsafe-inline'

Be careful!

DistriNet iMinds KU LEUVEN

Script/style nonces and hashes

CSP 1.1

- To allow controlled inline-scripts:
  - Mark your script with a nonce

```
Content-Security-Policy: default-src 'self'; script-src 'self'
https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

```
<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">
alert("Allowed because nonce is valid.")
</script>
```

  - Add a hash of your inline script to the policy

```
Content-Security-Policy: script-src 'sha256-
YWIzOWNiNzJjNDRlYzc4MTgwMDhmZDlkOWI0NTAyMjgyY2MyMWJl
MWUyNjc1ODJlYWJhNjU5MGU4NmZmNGU3OAo='
```

sha256

```
<script>alert('Hello, world.');</script>
```

CSP reporting feature

- CSP reports violations back to the server owner
  - server owner gets insides in actual attacks
    - i.e. violations against the supplied policy
  - allows to further fine-tune the CSP policy
    - e.g. if the policy is too restrictive

- report-uri directive
  - report-uri /my-csp-reporting-handler
  - URI to which the violation report will be posted

Example violation report

```
Content-Security-Policy: script-src 'self' https://apis.google.com;
report-uri http://example.org/my_amazing_csp_report_parser
```

CSP violation report

```
{
  "csp-report": {
    "document-uri": "http://example.org/page.html",
    "referrer": "http://evil.example.com/",
    "blocked-uri": "http://evil.example.com/evil.js",
    "violated-directive": "script-src 'self' https://apis.google.com",
    "original-policy": "script-src 'self' https://apis.google.com; report-
uri http://example.org/my_amazing_csp_report_parser"
  }
}
```

Based on “HTML5Rocks: An introduction to Content Security Policy” (Mike West)

CSP Reporting: one step further

- Apart from reporting violations via the report-uri directive

- CSP can also run in report only mode
  - Content-Security-Policy-Report-Only: default-src: 'none'; script-src 'self'; report-uri /my-csp-reporting-handler
  - Violation are reported
  - Policies are not enforced

DistriNet iMinds KU LEUVEN

Some CSP examples

- Examples:
  - Mybank.net lockdown
  - SSL only
  - Social media integration
  - Facebook snapshot

Based on “HTML5Rocks: An introduction to Content Security Policy” (Mike West)

Example: mybank.net lockdown

- Scripts, images, stylesheets
  - from a CDN at https://cdn.mybank.net
- XHR requests
  - Interaction with the mybank APIs at https://api.mybank.com
- Iframes
  - From the website itself
- No flash, java, ….

```
Content-Security-Policy: default-src 'none';
script-src https://cdn.mybank.net;
style-src https://cdn.mybank.net;
img-src https://cdn.mybank.net;
connect-src https://api.mybank.com;
child-src 'self'
```

DistriNet iMinds KU LEUVEN

Based on “HTML5Rocks: An introduction to Content Security Policy” (Mike West)

Example: SSL only

- Can we ensure to only include HTTPS content in our website?

```
Content-Security-Policy: default-src https: ;
script-src https: 'unsafe-inline';
style-src https: 'unsafe-inline'
```

- Obviously, this should only be the first step, not the final one!

Based on “HTML5Rocks: An introduction to Content Security Policy” (Mike West)

Example: social media integration

- Google +1 button
  - Script from https://apis.google.com
  - Iframe from https://plusone.google.com
- Facebook
  - Iframe from https://facebook.com
- Twitter tweet button
  - Script from https://platform.twitter.com
  - Iframe from https://platform.twitter.com

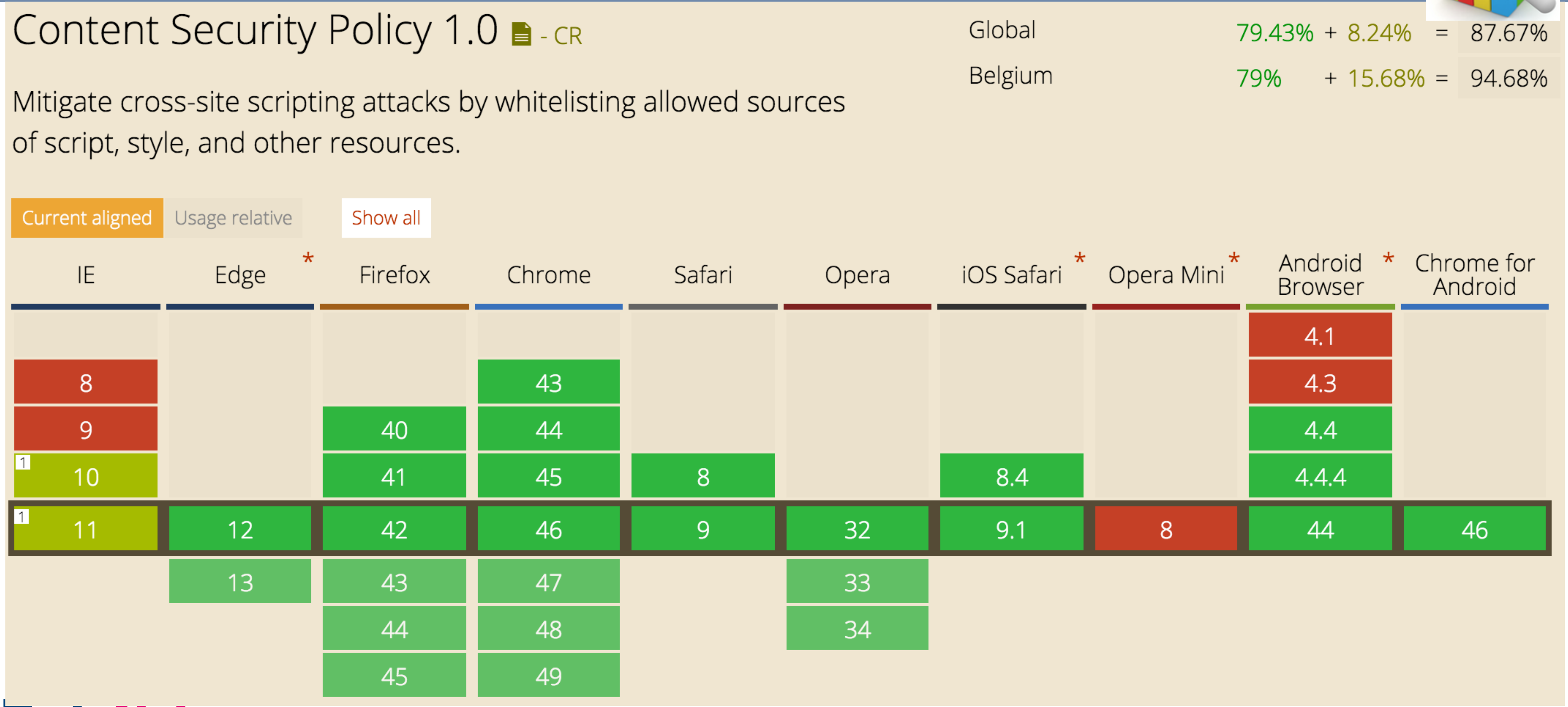
```
Content-Security-Policy: script-src https://apis.google.com
https://platform.twitter.com;
child-src https://plusone.google.com https://facebook.com
https://platform.twitter.com
```

Based on “HTML5Rocks: An introduction to Content Security Policy” (Mike West)

Example: Facebook snapshot

```
X-WebKit-CSP: default-src *;
script-src https://*.facebook.com http://*.facebook.com
https://*.fbcdn.net http://*.fbcdn.net *.facebook.net *.google-
analytics.com *.virtualearth.net *.google.com *.spotilocal.com:*
chrome-extension://lifbcibllhkdhoafpjfnlhfpfgnpldfl 'unsafe-inline'
'unsafe-eval' https://*.akamaihd.net http://*.akamaihd.net;style-
src * 'unsafe-inline';
connect-src https://*.facebook.com http://*.facebook.com
https://*.fbcdn.net http://*.fbcdn.net *.facebook.net
*.spotilocal.com:* https://*.akamaihd.net ws://*.facebook.com:*
http://*.akamaihd.net;
```

CSP 1.0: state-of-practice



Content Security Policy 1.0 - CR

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

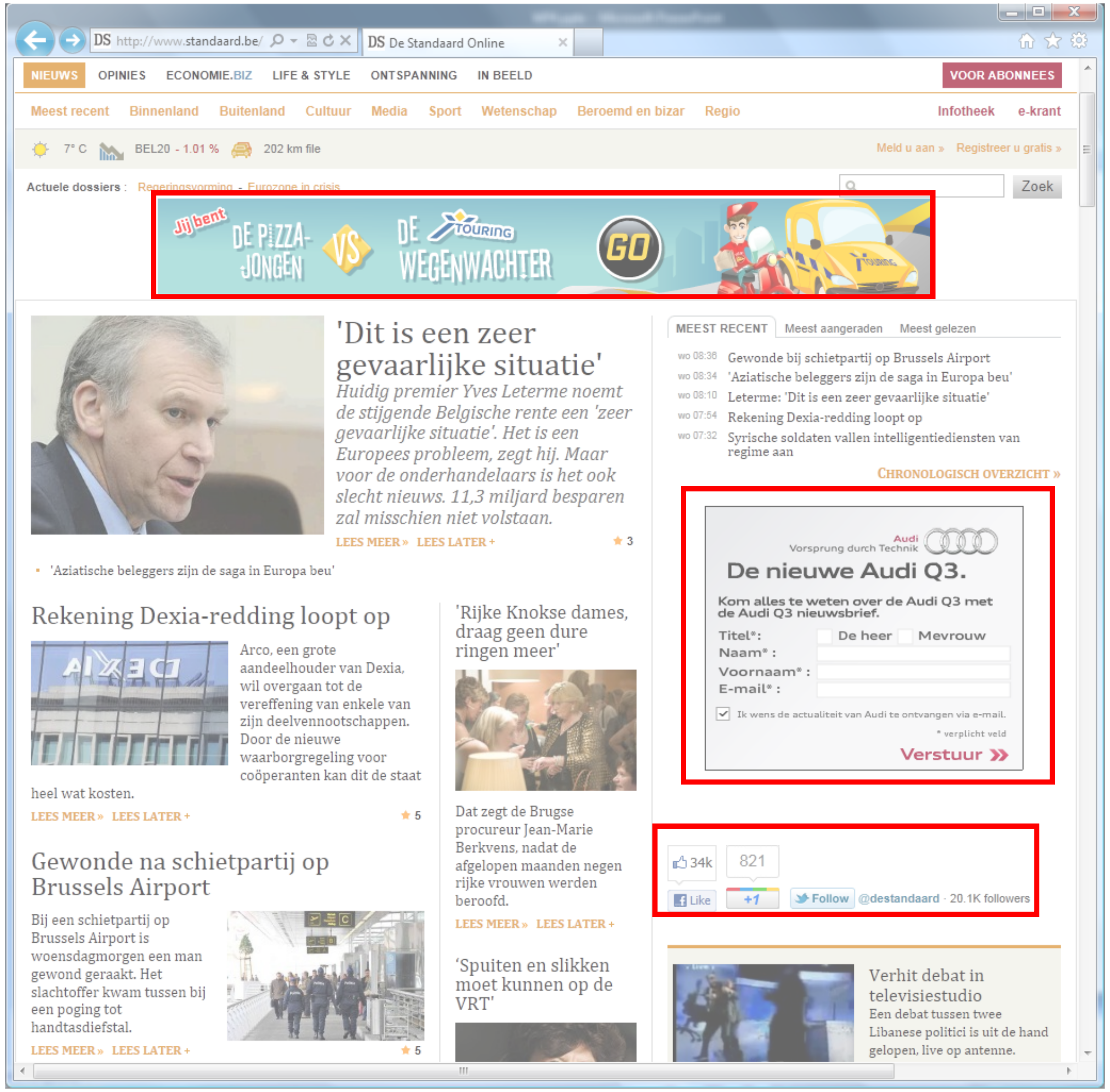
Global	79.43%	+ 8.24%	=	87.67%
Belgium	79%	+ 15.68%	=	94.68%

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
								🟥 4.1	
🟥 8			🟩 43					🟥 4.3	
🟥 9		🟩 40	🟩 44					🟩 4.4	
1 🟨 10		🟩 41	🟩 45	🟩 8		🟩 8.4		🟩 4.4.4	
1 🟨 11	🟩 12	🟩 42	🟩 46	🟩 9	🟩 32	🟩 9.1	🟥 8	🟩 44	🟩 46
	🟩 13	🟩 43	🟩 47		🟩 33				
		🟩 44	🟩 48		🟩 34				
		🟩 45	🟩 49						

Third-party JavaScript is everywhere

- Advertisements
  - Adhese ad network
- Social web
  - Facebook Connect
  - Google+
  - Twitter
  - Feedsburner
- Tracking
  - Scorecardresearch
- Web Analytics
  - Yahoo! Web Analytics
  - Google Analytics
- …



http://www.standaard.be/
DS De Standaard Online
NIEUWS OPINIES ECONOMIE.BIZ LIFE & STYLE ONTSPANNING IN BEELD
VOOR ABONNEES
Meest recent Binnenland Buitenland Cultuur Media Sport Wetenschap Beroemd en bizar Regio
Infotheek e-krant
7° C BEL20 - 1.01 % 202 km file
Meld u aan » Registreer u gratis »
Actuele dossiers : Regeringsvorming - Eurozone in crisis
Zoek
Jij bent DE PIZZA-JONGEN VS DE TOURING WEGENWACHTER GO
MEEST RECENT Meest aangeraden Meest gelezen

'Dit is een zeer gevaarlijke situatie'

Huidig premier Yves Leterme noemt de stijgende Belgische rente een 'zeer gevaarlijke situatie'. Het is een Europees probleem, zegt hij. Maar voor de onderhandelaars is het ook slecht nieuws. 11,3 miljard besparen zal misschien niet volstaan.

LEES MEER » LEES LATER + ★ 3

- 'Aziatische beleggers zijn de saga in Europa beu'

wo 08:36	Gewonde bij schietpartij op Brussels Airport
wo 08:34	'Aziatische beleggers zijn de saga in Europa beu'
wo 08:10	Leterme: 'Dit is een zeer gevaarlijke situatie'
wo 07:54	Rekening Dexia-redding loopt op
wo 07:32	Syrische soldaten vallen intelligentiediensten van regime aan

CHRONOLOGISCH OVERZICHT »

Audi Vorsprung durch Technik

De nieuwe Audi Q3.

Kom alles te weten over de Audi Q3 met de Audi Q3 nieuwsbrief.

Titel*: De heer Mevrouw
Naam*:
Voornaam*:
E-mail*:

☑ Ik wens de actualiteit van Audi te ontvangen via e-mail.

* verplicht veld

Verstuur »

Rekening Dexia-redding loopt op

Arco, een grote aandeelhouder van Dexia, wil overgaan tot de vereffening van enkele van zijn deelvennootschappen. Door de nieuwe waarborgregeling voor coöperanten kan dit de staat heel wat kosten.

LEES MEER » LEES LATER + ★ 5

'Rijke Knokse dames, draag geen dure ringen meer'

Dat zegt de Brugse procureur Jean-Marie Berkvens, nadat de afgelopen maanden negen rijke vrouwen werden beroofd.

LEES MEER » LEES LATER +

34k Like | 821 +1 | Follow @destandaard · 20.1K followers

Gewonde na schietpartij op Brussels Airport

Bij een schietpartij op Brussels Airport is woensdagmorgen een man gewond geraakt. Het slachtoffer kwam tussen bij een poging tot handtasdiefstal.

LEES MEER » LEES LATER + ★ 5

'Spuiten en slikken moet kunnen op de VRT'

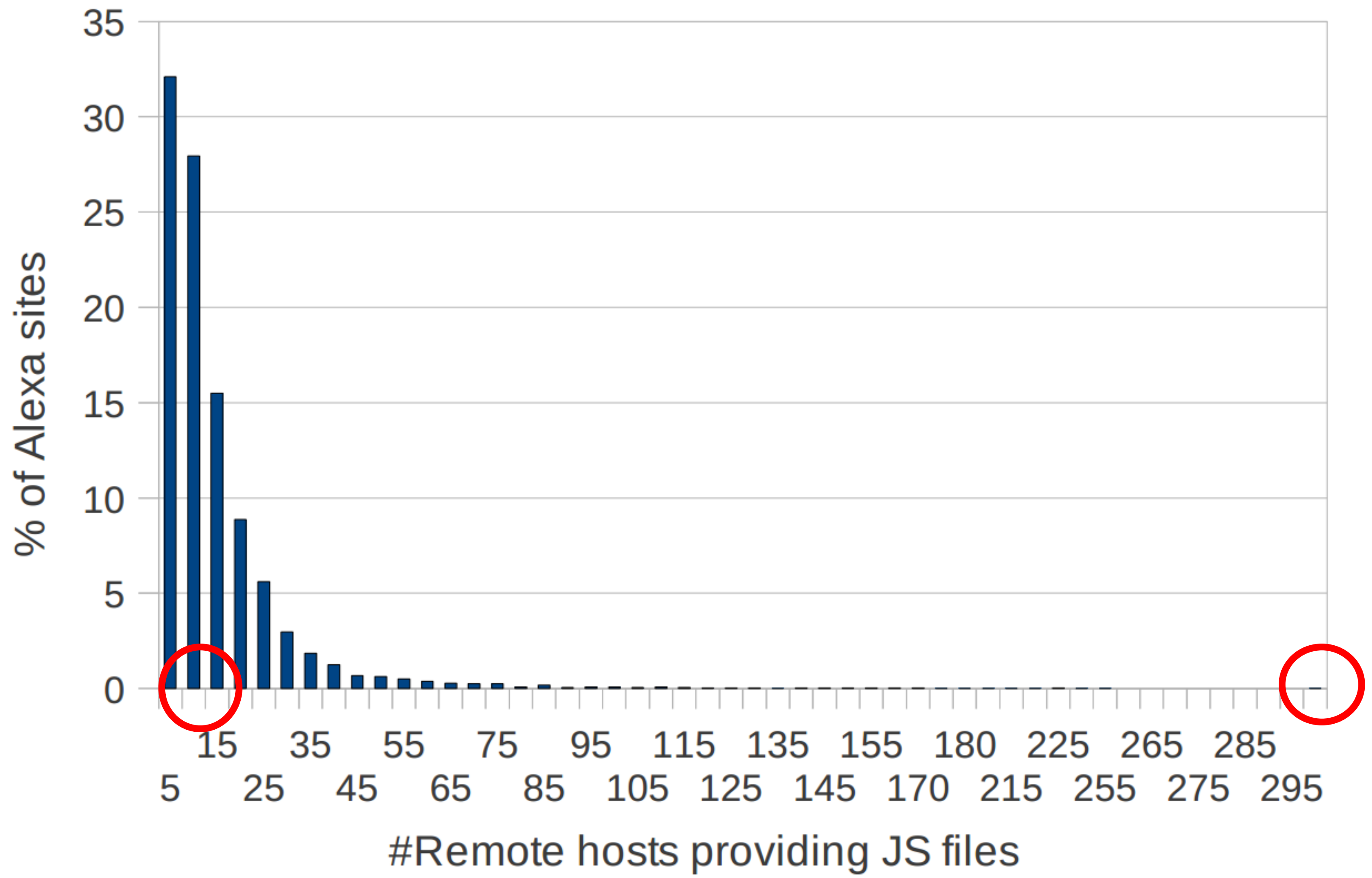
Verhit debat in televisiestudio

Een debat tussen twee Libanese politici is uit de hand gelopen, live op antenne.

Spongecell
Hula
kiko
Trumba
eskobo
mayomi
Pageflakes
vimeo
skobee
shadows
gravee
YouTube
Zimbra
LookSmart FURL Your Personal Web File
smugmug
newsgator
Blogniscient
TinFinger
shutterfly
memeflux
PodDater
Feedster
favoor
Planzo.com
ZAZZLE
Tailrank
TagWorld
nuvvo
dogear
yakalike
grouper
ODDPOST
QOOP
iNods
Lulu
rbloc.com
blish
flagr
FireAnt
simplyhired
veoh
theadcloud
pixagogo
Renkoo
gather
Agatra
browsr
oyogi
cafepress
standpoint
meebo
EXTRA TASTY!
last.fm
YEDDA
Jotspot
Frappr!
jeteye
dabble db
Writeboard
SHOUTWIRE
iKarma
Rollyo
AirSet
tech.memeorandum
CalendarHub
Suprglu
PIECING YOUR WEB TOGETHER
pando
zigtag
Findory
backfence
clipmarks
wayfaring
gOFFICE
AllPeers
orb
Rallypoint
Zoozio
blogbeat
Ziggs
zooto
vSocial
Boltfolio
wink
riya
Audible Wordcast
Opinity
Because your reputation matters!
reddit
measure map
gumshoo
bluepulse
imvu BETA
STREAMLOAD
Freedom for Your Digital Lifestyle
Ta-da Lists
FeedSky
jellyBarn.INC
FeedTier
phanfare
Fruitcast
PubSub
nativetext
CONGOO BETA
PODZINGER
RSS MAD
WIKIPEDIA The Free Encyclopedia
yubnub
dPolls
flickr
Ning
Ookles
Strongspace
zoominfo
CASTPOST

Number of remote script providers per site

- 88.45% includes at least 1 remote JavaScript library
- 2 out of 3 sites relies on 5 or more script providers
- 1 site includes up to 295 remote script providers



<u>Source:</u> *Nick Nikiforakis et. al. You are what you include: Large-scale evaluation of remote JavaScript inclusions. CCS 2012*

Most popular JavaScript libraries and APIs

Offered service	JavaScript file	% Alexa Top 10K
Web analytics	www.google-analytics.com/ga.js	68,37%
Dynamic Ads	pagead2.googlesyndication.com/pagead/show_ads.js	23,87%
Web analytics	www.google-analytics.com/urchin.js	17,32%
Social Networking	connect.facebook.net/en_us/all.js	16,82%
Social Networking	platform.twitter.com/widgets.js	13,87%
Social Networking & Web analytics	s7.addthis.com/js/250/addthis_widget.js	12,68%
Web analytics & Tracking	edge.quantserve.com/quant.js	11,98%
Market Research	b.scorecardresearch.com/beacon.js	10,45%
Google Helper Functions	www.google.com/jsapi	10,14%
Web analytics	ssl.google-analytics.com/ga.js	10,12%

Source: *Nick Nikiforakis et. al. You are what you include: Large-scale evaluation of remote JavaScript inclusions. CCS 2012*

Subresource Integrity

- Either you trust a CDN, or you host it yourself

```
<script src="https://code.jquery.com/jquery-2.1.3.min.js"
        integrity="sha256-TXuiaAJuML3…uMLTXuiaAJ3"
        crossorigin="anonymous"></script>
```

- Welcome **Subresource Integrity (SRI)**
  - W3C Candidate Recommendation since November 12, 2015

DistriNet iMinds KU LEUVEN

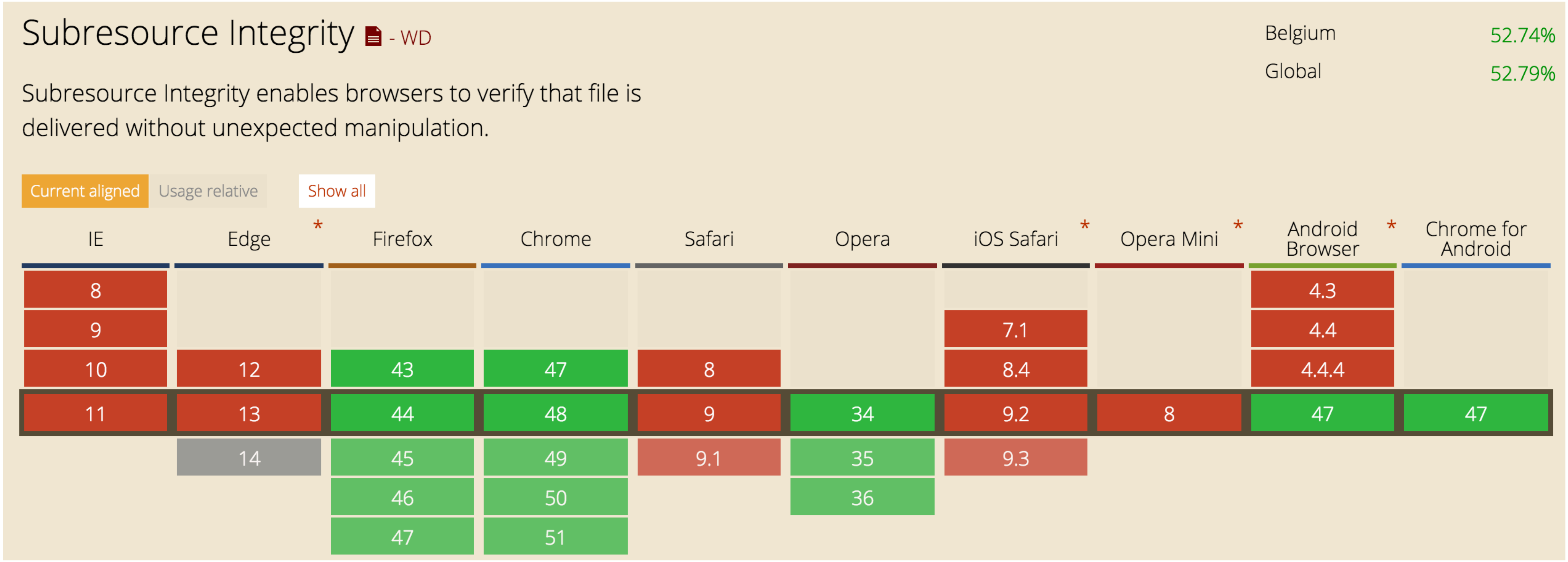
Subresource Integrity

- Allows you to specify a hash of an external resource
  - Using the *integrity* attribute on *script* or *link* tags
- Browsers verify this hash before loading the file
  - Refuse to load the file if the hash does not match
- SRI supports the specification of multiple hashes
  - The strongest one available will be used by the browser

```
<script src="myapplication.js"
  integrity="sha256-… sha512-… ">
</script>
```

```
<link href="myapp.css" type="text/css"
   integrity="sha384-… sha512-…" />
```

SRI: state-of-practice



Subresource Integrity - WD

Subresource Integrity enables browsers to verify that file is delivered without unexpected manipulation.

Belgium	52.74%
Global	52.79%

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
🟥 8								🟥 4.3	
🟥 9						🟥 7.1		🟥 4.4	
🟥 10	🟥 12	🟩 43	🟩 47	🟥 8		🟥 8.4		🟥 4.4.4	
🟥 11	🟥 13	🟩 44	🟩 48	🟥 9	🟩 34	🟥 9.2	🟥 8	🟩 47	🟩 47
	⬜ 14	🟩 45	🟩 49	🟥 9.1	🟩 35	🟥 9.3			
		🟩 46	🟩 50		🟩 36				
		🟩 47	🟩 51						

Recap: Mitigating script injection attacks

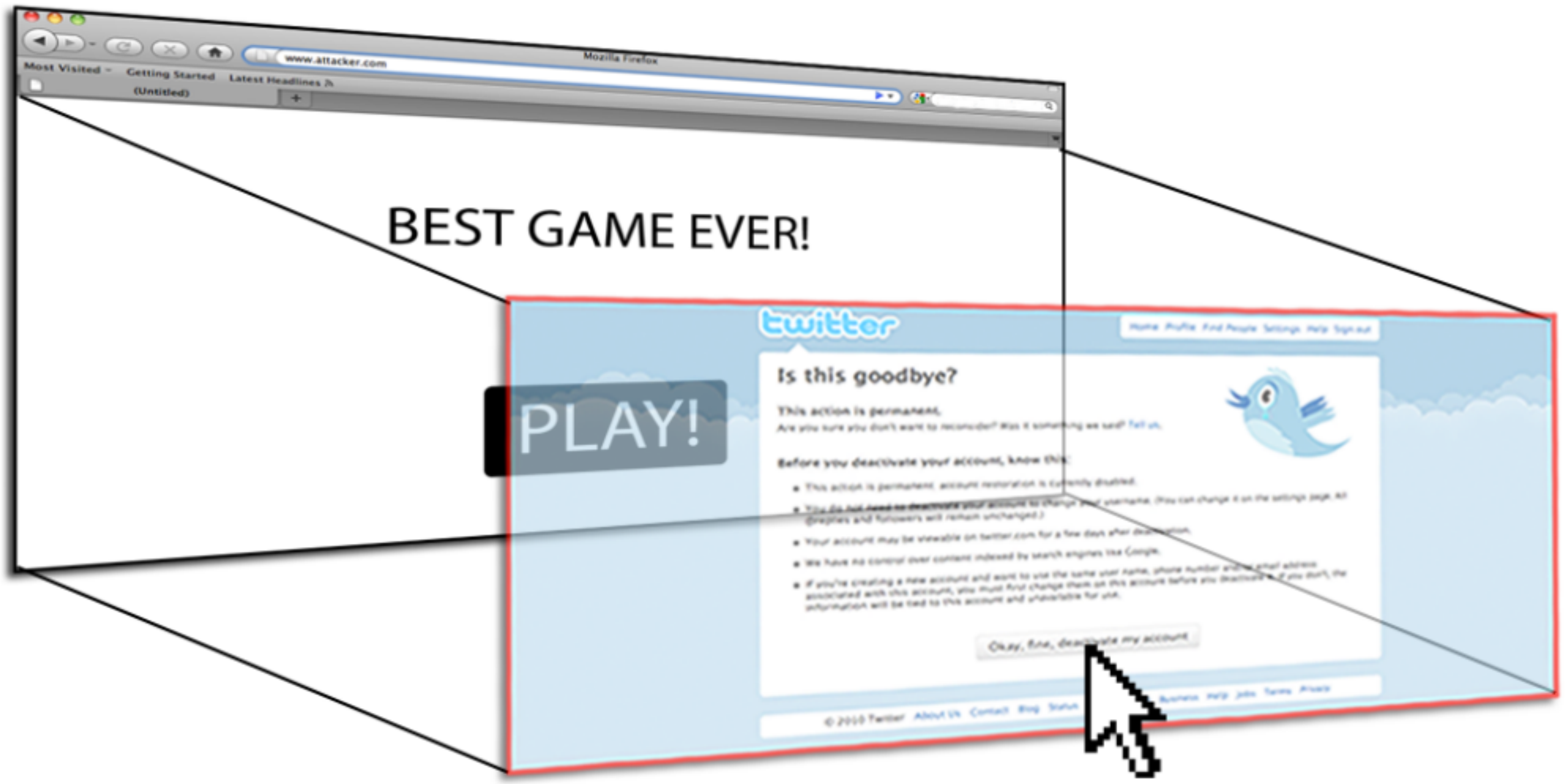
- HttpOnly flag for session cookies
  - To protect cookies against hijacking and fixation from JavaScript
- Content Security Policy (CSP)
  - Domain-level control over resources to be included
  - Most promising infrastructural technique against XSS
  - Interesting reporting-only mode
- Subresource integrity (SRI)
  - Guarantee the integrity of scripts delivered via third-parties

#3 Framing content securely

Overview

- Attacks:
  - Click-jacking
  - Same domain XSS
- Countermeasures:
  - X-Frame-Options / frame-ancestors
  - HTML5 sandbox attribute for iframes

Click-jacking



Source: “Busting Frame Busting: a Study of Clickjacking Vulnerabilities on Popular Sites” (W2SP 2010)

Unsafe countermeasures

- A lot of unsafe ways exist to protect against clickjacking

  -
    ```
    if (top.location != location)
    top.location = self.location;
    ```

  -
    ```
    if (parent.location != self.location)
    parent.location = self.location;
    ```

- Can easily be defeated by
  - Script disabling/sandboxing techniques
  - Frame navigation policies
  - XSS filters in browsers

Source: "Busting Frame Busting: a Study of Clickjacking Vulnerabilities on Popular Sites" (W2SP 2010)

X-Frame-Options

- Issued by the HTTP response header
  - X-Frame-Options: SAMEORIGIN
  - Indicates if and by who the page might be framed

- 3 options:
  - DENY
  - SAMEORIGIN
  - ALLOW-FROM uri

XFO has been integrated in CSP

- New CSP directive: frame-ancestors
  - Content-Security-Policy: frame-ancestors https://partnerA.com https://partnerB.com

- In contrast to X-Frame-Options, a sourcelist is allowed
  - Common advice is to tailor per partner

Limitations of framing content in same origin

- Iframe integration provides a good isolation mechanism
  - Each origin runs in its own security context, thanks to the Same-Origin Policy
  - Isolation only holds if outer and inner frame belong to a different origin
- Hard to isolate untrusted content within the same origin

HTML5 sandbox attribute

- Expressed as attribute of the iframe tag
  - `<iframe src= "/untrusted-path/index.html" sandbox></iframe>`
  - `<iframe src="/untrusted-path/index.html" sandbox= "allow-scripts"></iframe>`

- Level of Protection
  - Coarse-grained sandboxing
  - ‘SOP but within the same domain’

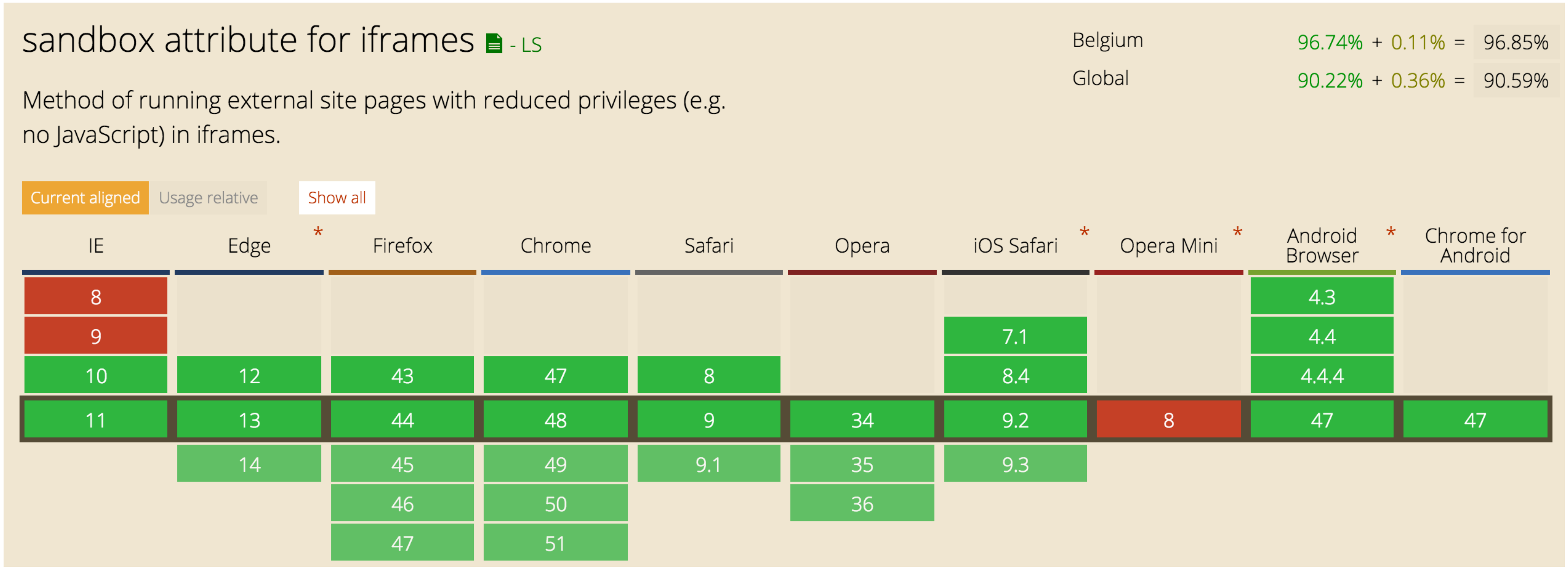
Default sandbox behavior

- Plugins are disabled
- Frame runs in a unique origin
- Scripts can not execute
- Form submission is not allowed
- Top-level context can not be navigated
- Popups are blocked
- No access to raw mouse movements data

Sandbox relaxation directives

- Relaxations:
  - allow-forms
  - allow-popups
  - allow-pointer-lock
  - allow-same-origin
  - allow-scripts
  - allow-top-navigation
- Careful!
  - Combining allow-scripts & allow-same-origin voids the sandbox isolation
- Plugins can not be re-enabled

HTML5 sandbox



sandbox attribute for iframes - LS

Method of running external site pages with reduced privileges (e.g. no JavaScript) in iframes.

Belgium	96.74%	+ 0.11%	=	96.85%
Global	90.22%	+ 0.36%	=	90.59%

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
🟥 8								🟩 4.3	
🟥 9						🟩 7.1		🟩 4.4	
🟩 10	🟩 12	🟩 43	🟩 47	🟩 8		🟩 8.4		🟩 4.4.4	
🟩 11	🟩 13	🟩 44	🟩 48	🟩 9	🟩 34	🟩 9.2	🟥 8	🟩 47	🟩 47
	🟩 14	🟩 45	🟩 49	🟩 9.1	🟩 35	🟩 9.3			
		🟩 46	🟩 50		🟩 36				
		🟩 47	🟩 51						

Sandbox has been integrated in CSP

CSP 1.1

- New CSP directive: sandbox
  - `Content-Security-Policy: sandbox`
  - `Content-Security-Policy: sandbox allow-scripts`
- Similar options apply:
  - allow-forms
  - allow-pointer-lock
  - allow-popups
  - allow-same-origin
  - allow-scripts
  - allow-top-navigation

DistriNet iMinds KU LEUVEN

Recap: Framing content securely

- CSP: Frame ancestors
  - Robust defense against click-jacking
  - Any state-changing page should be protected
- CSP: Sandbox attribute
  - Coarse-grained sandboxing of resources and JavaScript
  - Interesting enabler for security architectures

Wrap-up

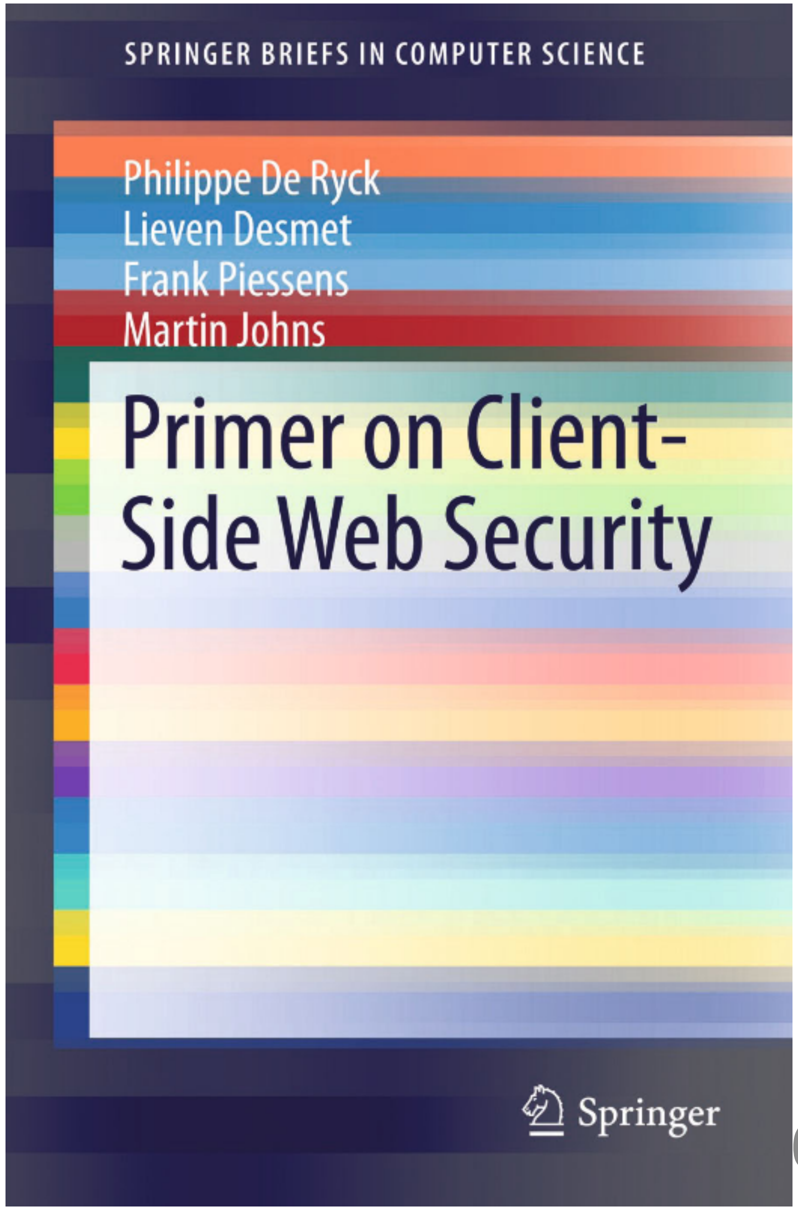
Conclusion

- Whole new range of security features
  - Browser-side enforcement, under control of the server
- NOT a replacement of secure coding guidelines, but an interesting additional line of defense for
  - Legacy applications
  - Newly deployed applications
- And most probably, there is many more to come in the next few years…

Primer on Client-Side Web Security

STREWS

- Covers the landscape of client-side Web security
  - State-of-the-art in web security
  - State-of-practice on the Web
  - Recent research and standardization activities
  - Security best practices per category


SPRINGER BRIEFS IN COMPUTER SCIENCE

Philippe De Ryck
Lieven Desmet
Frank Piessens
Martin Johns

Primer on Client-Side Web Security

Springer

DistriNet
iMinds KU LEUVEN

References

- Ph. De Ryck, M. Decat, L. Desmet, F. Piessens, W. Joosen. Security of web mashups: a survey (NordSec 2010)
- P. Chen, N. Nikiforakis, L. Desmet and Ch. Huygens. A Dangerous Mix: Large-scale analysis of mixed-content websites (ISC 2013)
- N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, Ch. Kruegel, F. Piessens, G. Vigna, You are what you include: Large-scale evaluation of remote JavaScript inclusions (CCS 2012)
- Ph. De Ryck et al., Web-platform security guide: Security assessment of the Web ecosystem (STREWS Deliverable D1.1)
- A. Barth, D. Veditz, M. West, Content Security Policy 1.1, W3C Working Draft 11 February 2014
- G.Rydstedt, E. Bursztein, D. Boneh, and C. Jackson. Busting frame busting: a study of clickjacking vulnerabilities at popular sites (W2SP 2010)
- Mike West. An introduction to Content Security Policy (HTML5 Rocks tutorials)
- Mike West. Confound Malicious Middlemen with HTTPS and HTTP Strict Transport Security (HTML5 Rocks tutorials)

References (2)

- Mike West. Play safely in sandboxed iframes (HTML5 Rocks tutorials)
- Ivan Ristic. Internet SSL Survey 2010 (Black Hat USA 2010)
- Moxie Marlinspike. New Tricks for Defeating SSL in Practice (BlackHat DC 2009)
- Mike West. Securing the Client-Side: Building safe web applications with HTML5 (Devoxx 2012)
- B. Sterne, A. Barth. Content Security Policy 1.0 (W3C Candidate Recommendation)
- D. Ross, T. Gondrom. HTTP Header Frame Options (IETF Internet Draft)
- J. Hodges, C. Jackson, A. Barth. HTTP Strict Transport Security (HSTS) (IETF RFC 6797)
- C. Evans, C. Palmer, R. Sleevi. Public Key Pinning Extension for HTTP (IETF Internet Draft)
- Can I use … ?, http://caniuse.com/