HTTP Security Headers

A technology history through scar tissue

@SecTinkerer

$ whoami

- Benjamin Hering
- SANS Community Mentor
- Minor minion of BSidesSF organizers
- Security @ ASAPP

(We bought you lunch yesterday!)

@SecTinkerer

$ sudo !! --help

Wait, can I see that slide again?

Slides are linked in sched
`https://bsidessf2019.sched.com`

I have a question. I'd like to understand more of something.

Go to `https://sli.do`
Event code `#BSidesSF2019`
Select `City View Track`

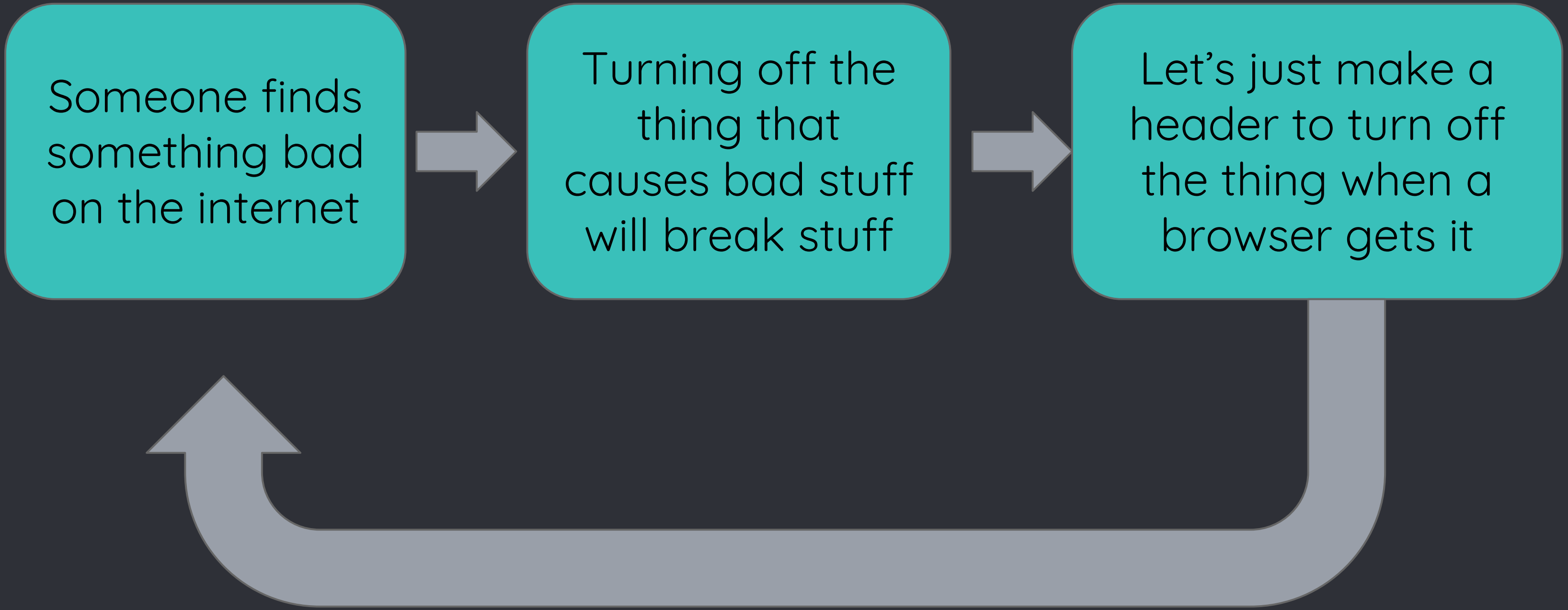
This question is really more of a comment but…

I need to say smart and witty things on how terrible you and/or your talk and/or BsidesSF are

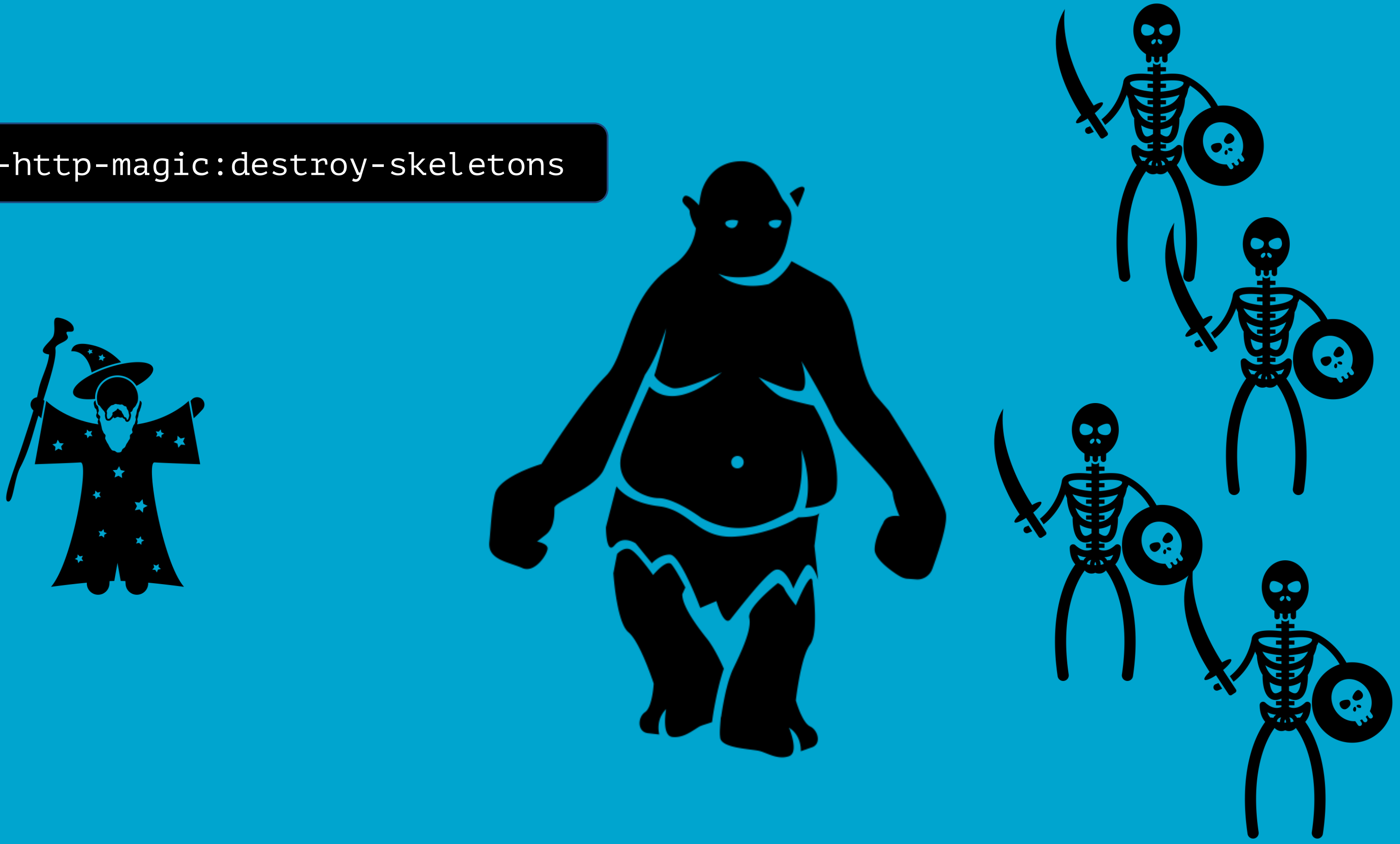
Wait! There wasn't time to get to my question after I submitted it to `sli.do` and I still want to understand more!

@SecTinkerer

Security Header Flowchart


Someone finds something bad on the internet
Turning off the thing that causes bad stuff will break stuff
Let’s just make a header to turn off the thing when a browser gets it

```
x-http-magic:destroy-skeletons
```



@SecTinkerer

Headers we'll cover

- x-content-type-options
- x-xss-protection
- x-frame-options
- strict-transport-security
- referrer-policy
- feature-policy (kinda, it's not 100% here yet)
- content-security-policy

`x-content-type-options`

Implemented in Chrome 1, IE 8

Implemented across browsers Dec 2008 - Oct 2009

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

What’s the bad thing?

Upload executable files with a non-executable MIME type

Attacker tricks upload of malicious JavaScript as foo.txt (`text/plain` MIME)

Browser ‘sniffs’ the file as actually JavaScript. Changes to `application/javascript` MIME and executes

Set to

```
x-content-type-options: nosniff
```

Tells browsers to not try and guess MIME types of files

Just set your MIME types correctly

`x-xss-protection`

Implemented in Chrome 1, IE 8

Implemented across browsers Dec 2008 - Oct 2009

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

What’s the bad thing?

Reflected Cross Site Scripting (XSS)

Rolled out in IE 8, off by default could be enabled by headers

Stops the lowest hanging XSS fruit

```
GET /index.php&name=<script>badstuff</script>
```

```
<html>
…
<body>
<p>Hello<script>badstuff</script>!
</p>
…
```

@SecTinkerer

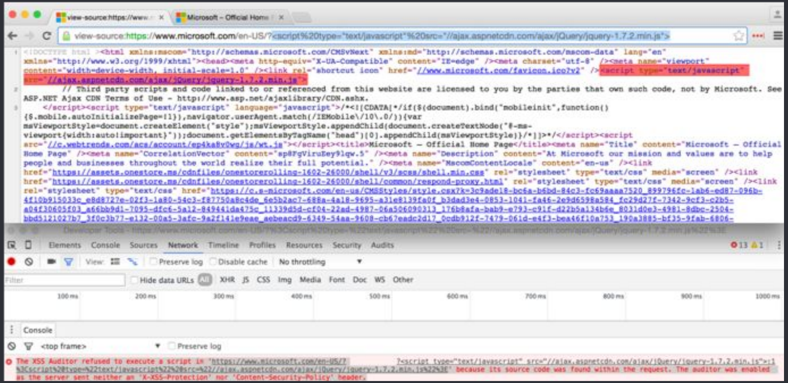
What to set?

`x-xss-protection: 0`

- Turns off the filter. Protect other ways

`x-xss-protection: 1; mode=block`

- Instead of stripping out what it thinks, blocks the entire request
- MS10-002 abused the filter to perform XSS attacks through the XSS filter
- XSS filter can be used to knock out legit scripts
- **CSP header is better**



view-source:https://www.microsoft.com/en-US/?<script%20type="text/javascript"%20src="//ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js">

```
<!DOCTYPE html ><html xmlns:mscom="http://schemas.microsoft.com/CMSvNext" xmlns:md="http://schemas.microsoft.com/mscom-data" lang="en"
xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="X-UA-Compatible" content="IE=edge" /><meta charset="utf-8" /><meta name="viewport"
content="width=device-width, initial-scale=1.0" /><link rel="shortcut icon" href="//www.microsoft.com/favicon.ico?v2" /><script type="text/javascript"
src="//ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js">
        // Third party scripts and code linked to or referenced from this website are licensed to you by the parties that own such code, not by Microsoft. See
ASP.NET Ajax CDN Terms of Use - http://www.asp.net/ajaxlibrary/CDN.ashx.
    </script><script type="text/javascript" language="javascript">/*<![CDATA[*/if($(document).bind("mobileinit",function()
{$.mobile.autoInitializePage=!1}),navigator.userAgent.match(/IEMobile\/10\.0/)){var
msViewportStyle=document.createElement("style");msViewportStyle.appendChild(document.createTextNode("@-ms-
viewport{width:auto!important}"));document.getElementsByTagName("head")[0].appendChild(msViewportStyle)}/*]]>*/</script><script
src="//c.webtrends.com/acs/account/ep4ka8vOwg/js/wt.js"></script><title>Microsoft – Official Home Page</title><meta name="Title" content="Microsoft – Official
Home Page" /><meta name="CorrelationVector" content="sp8FgViruEey9lqw.5" /><meta name="Description" content="At Microsoft our mission and values are to help
people and businesses throughout the world realize their full potential." /><meta name="MscomContentLocale" content="en-us" /><link
href="https://assets.onestore.ms/cdnfiles/onestorerolling-1602-26000/shell/v3/scss/shell.min.css" rel="stylesheet" type="text/css" media="screen" /><link
href="https://assets.onestore.ms/cdnfiles/onestorerolling-1602-26000/shell/common/respond-proxy.html" rel="stylesheet" type="text/css" media="screen" /><link
rel="stylesheet" type="text/css" href="https://c.s-microsoft.com/en-us/CMSStyles/style.csx?k=3c9ade18-bc6a-b6bd-84c3-fc69aaaa7520_899796fc-1ab6-ed87-096b-
4f10b915033c_e8d8727e-02f3-1a80-54c3-f87750a8c4de_6e5b2ac7-688a-4a18-9695-a31e8139fa0f_b3dad3e4-0853-1041-fa46-2e9d6598a584_fc29d27f-7342-9cf3-c2b5-
a04f30605f03_a66bb9d1-7095-dfc6-5a12-849441da475c_11339d5d-cf04-22ad-4987-06a506090313_176b8afa-bab9-e793-c91f-d22b5a134b6e_8031d0e3-4981-8dbc-2504-
bbd5121027b7_3f0c3b77-e132-00a5-3afc-9a2f141e9eae_aebeacd9-6349-54aa-9608-cb67eadc2d17_0cdb912f-7479-061d-e4f3-bea46f10a753_190a3885-bf35-9fab-6806-
```

```
The XSS Auditor refused to execute a script in 'https://www.microsoft.com/en-US/?        ?<script type="text/javascript" src="//ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js">:1
%3Cscript%20type=%22text/javascript%22%20src=%22//ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js%22%3E' because its source code was found within the request. The auditor was enabled
as the server sent neither an 'X-XSS-Protection' nor 'Content-Security-Policy' header.
```

Real life?

```
$ curl -I https://www.facebook.com
x-content-type-options: nosniff
x-xss-protection: 0
x-frame-options: DENY

$ curl -I https://www.google.com
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
```

x-frame-options

Implemented in Chrome 4, Firefox 3.6.9, IE 8

Implemented across browsers Oct 2009 - Sept 2010

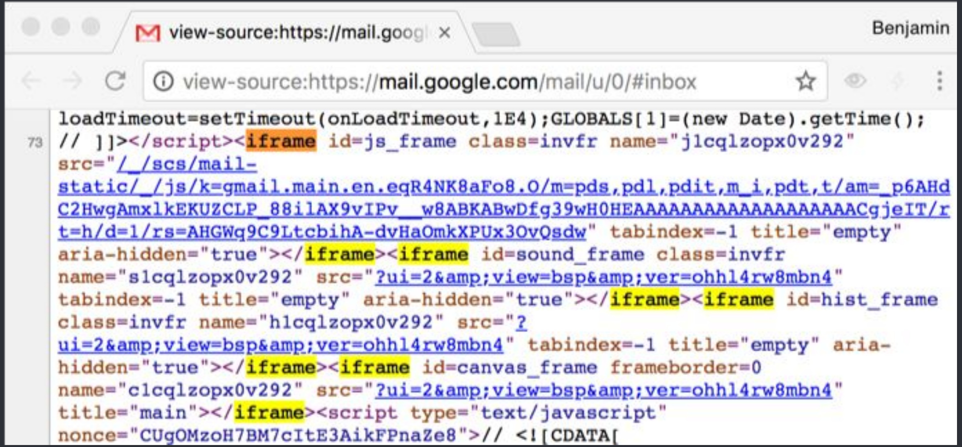
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

`<frame>` (deprecated) `<iframe>` `<object>`

Embeds pages in other pages

`<frame>` is obsolete in HTML5, but

`<iframe>` is still in gmail and will probably never die



view-source:https://mail.googl ×   Benjamin

view-source:https://mail.google.com/mail/u/0/#inbox

```
loadTimeout=setTimeout(onLoadTimeout,1E4);GLOBALS[1]=(new Date).getTime();
// ]]></script><iframe id=js_frame class=invfr name="j1cqlzopx0v292"
src="/_/scs/mail-
static/_/js/k=gmail.main.en.eqR4NK8aFo8.O/m=pds,pdl,pdit,m_i,pdt,t/am=_p6AHd
C2HwgAmx1kEKUZCLP_88i1AX9vIPv__w8ABKABwDfg39wH0HEAAAAAAAAAAAAAAAAAAACgjeIT/r
t=h/d=1/rs=AHGWq9C9LtcbihA-dvHaOmkXPUx3OvQsdw" tabindex=-1 title="empty"
aria-hidden="true"></iframe><iframe id=sound_frame class=invfr
name="s1cqlzopx0v292" src="?ui=2&amp;view=bsp&amp;ver=ohhl4rw8mbn4"
tabindex=-1 title="empty" aria-hidden="true"></iframe><iframe id=hist_frame
class=invfr name="h1cqlzopx0v292" src="?
ui=2&amp;view=bsp&amp;ver=ohhl4rw8mbn4" tabindex=-1 title="empty" aria-
hidden="true"></iframe><iframe id=canvas_frame frameborder=0
name="c1cqlzopx0v292" src="?ui=2&amp;view=bsp&amp;ver=ohhl4rw8mbn4"
title="main"></iframe><script type="text/javascript"
nonce="CUgOMzoH7BM7cItE3AikFPnaZe8">// <![CDATA[
```

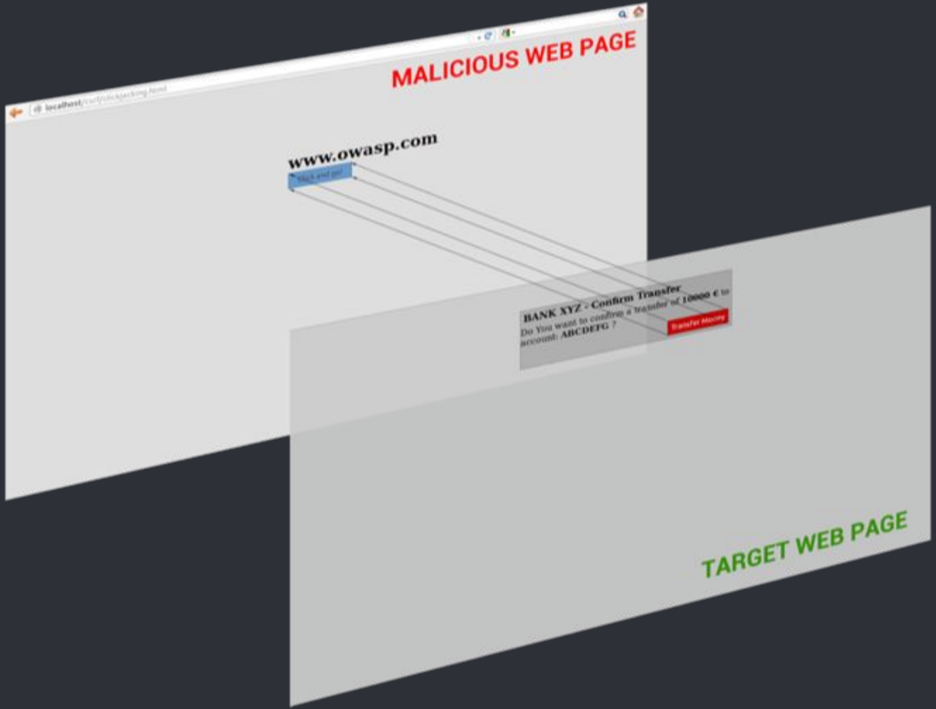
@SecTinkerer

What's the bad thing?

Clickjacking

Drawing a cosmetic top above actual buttons

Add keystroke logging JS


MALICIOUS WEB PAGE
localhost
www.owasp.com
Click and get
BANK XYZ - Confirm Transfer
Do You want to confirm a transfer of 10000 € to account: ABCDEFG ?
Transfer Money
TARGET WEB PAGE

@SecTinkerer

What to set?

`x-frame-options: DENY`

- No embedding at all

`x-frame-options: SAMEORIGIN`

- Allow embedding from the same origin as embedded page

`x-frame-options: ALLOW-FROM https://example.com`

- Allow embedding only on specifically outlined locations

(`content-security-policy: frame-ancestors ‘none’;` also works, but only in some modern browsers. No IE, No Microsoft Edge on mobile)

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors

`strict-transport-security`

Implemented in Chrome 4, Firefox 4, IE 11

Implemented across browsers Jan 2010 - Oct 2013

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security

What's the bad thing?

Where's example.com?

example.com is at `42.42.42.42`

Great. Give me
http://example.com
`42.42.42.42 port 80`

Let's do HTTPS instead.
`301 Redirect`
`https://example.com`

No problem. Give me
https://example.com
`42.42.42.42 port 443`

Here's my TLS cert

```
<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta content=..
```

I trust that cert.
I'll load the page

@SecTinkerer

What’s the bad thing?

Where’s example.com?

example.com happens to be me! `192.168.1.1`

Great. Give me http://example.com
`192.168.1.1 port 80`

Give me https://example.com
`42.42.42.42 port 443`

Here’s my TLS cert

Yeah, whatever, just give me the page

```
<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><meta content=..
```

```
<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage"
lang="en"><head><script src=“evil.com/bad.js” …
```

Sweet. Lemme just go run evil.com/bad.js

HTTP

HTTPS

@SecTinkerer

What’s the bad thing?

Where’s example.com?

example.com happens to be me! `192.168.1.1`

Great. Give me
http://example.com
`192.168.1.1 port 80`

Wait, example.com has
HSTS enabled! Give me
https://example.com
`192.168.1.1 port 443`

Uh, okay, here’s my cert

This cert is junk!
Don’t load!

HSTS cert error **cannot be clicked through**

HTTPS

@SecTinkerer

What to set?

`strict-transport-security: max-age=<expire-time>`

- All further requests to that site will be over HTTPS until age expires

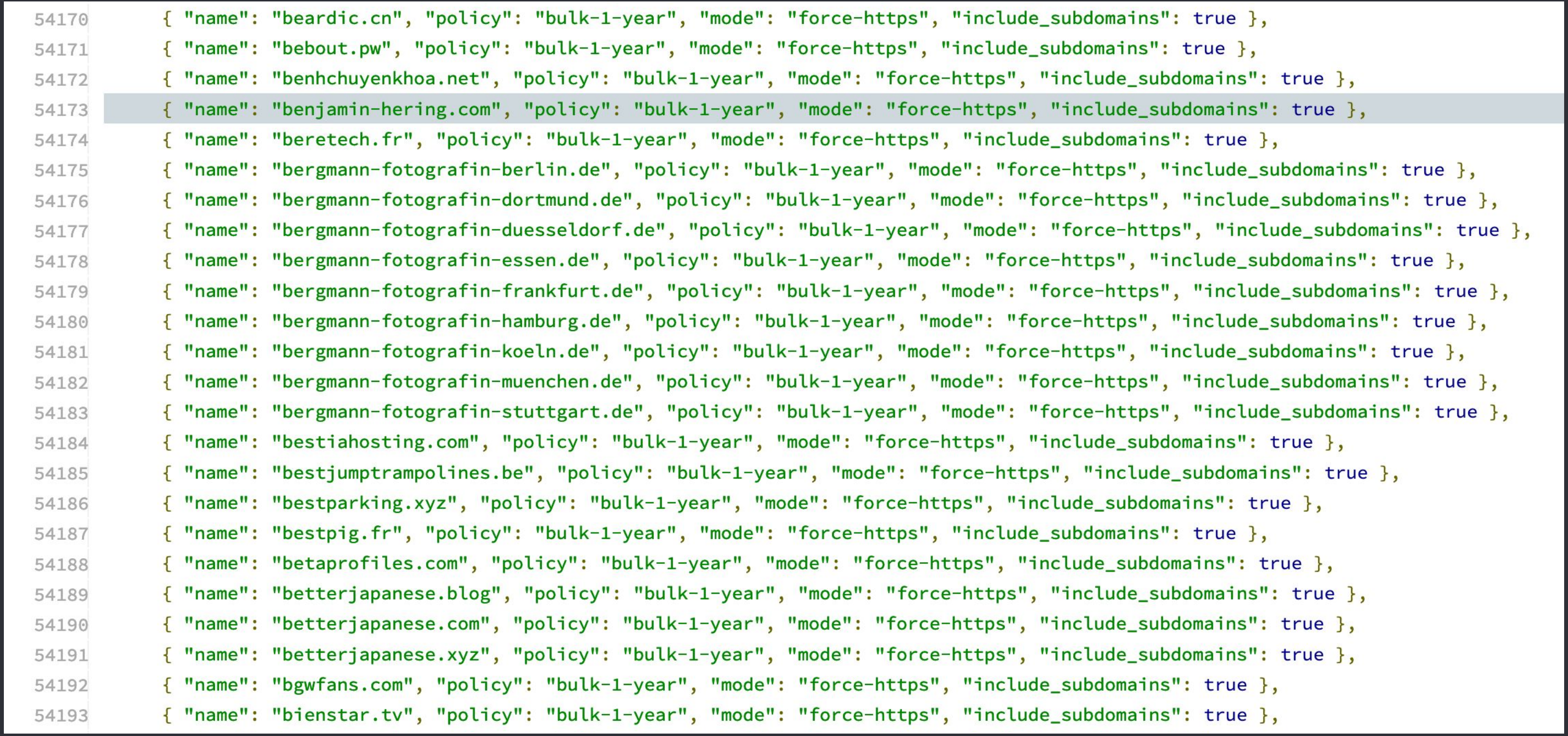
`strict-transport-security: max-age=<expire-time>; includeSubDomains`

- All further requests to that site & subdomains will be over HTTPS until age expires

`strict-transport-security: max-age=<expire-time>; includeSubDomains; preload`

- All further requests to that site & subdomains will be over HTTPS until age expires and it will get preloaded into browsers
- Preload requires:
  - `max-age=31536000` or higher (one year)
  - `includeSubDomains`
  - valid HTTPS cert, serving HTTPS on all subdomains
  - redirect port 80 to port 443 (if listening on port 80)
- If any subdomain can't handle HTTPS it will break. By design, no quick rollbacks.
- https://hstspreload.org/

My tiny blog domain is on your computer and phone! (probably multiple times!)



```
54170        { "name": "beardic.cn", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54171        { "name": "bebout.pw", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54172        { "name": "benhchuyenkhoa.net", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54173        { "name": "benjamin-hering.com", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54174        { "name": "beretech.fr", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54175        { "name": "bergmann-fotografin-berlin.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54176        { "name": "bergmann-fotografin-dortmund.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54177        { "name": "bergmann-fotografin-duesseldorf.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54178        { "name": "bergmann-fotografin-essen.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54179        { "name": "bergmann-fotografin-frankfurt.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54180        { "name": "bergmann-fotografin-hamburg.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54181        { "name": "bergmann-fotografin-koeln.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54182        { "name": "bergmann-fotografin-muenchen.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54183        { "name": "bergmann-fotografin-stuttgart.de", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54184        { "name": "bestiahosting.com", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54185        { "name": "bestjumptrampolines.be", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54186        { "name": "bestparking.xyz", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54187        { "name": "bestpig.fr", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54188        { "name": "betaprofiles.com", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54189        { "name": "betterjapanese.blog", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54190        { "name": "betterjapanese.com", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54191        { "name": "betterjapanese.xyz", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54192        { "name": "bgwfans.com", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
54193        { "name": "bienstar.tv", "policy": "bulk-1-year", "mode": "force-https", "include_subdomains": true },
```

https://chromium.googlesource.com/chromium/src/net/+/refs/heads/master/http/transport_security_state_static.json#54173
(plus or minus other new adds)

@SecTinkerer

`referrer-policy`

Implemented in Chrome 61, Firefox 52, no Safari or Microsoft Edge support

Implemented across browsers March 2017

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy

What’s the bad thing?

When someone clicks a link off your site, you can leak specific information about a user in the “referer” (sic) header

Maybe your site is sensitive

Or specific URLs can be used to identify users

```
referer: https://www.bank.com/account/loanPerf.actio
n?loan_id=88783584&customer_id=124783505

referer: https://site-govt-doesnt-like.com

referer: https://twitter.com/SecTinkerer
```

What to set?

Depends on your site. Common good choices are:

`referrer-policy: strict-origin`

- Only sends the origin and not URI info. Won't send over HTTP (`referer: https://benjamin-hering.com`)

`referrer-policy: strict-origin-when-cross-origin`

- Only sends origin as a referer when links off your site. Internally, the header keeps the full uri. Won't send the header over HTTP

`referrer-policy: no-referrer`

- Just don't send the referer header ever

`feature-policy`

Chrome 2018; no IE, Edge, or Safari support. Firefox defaults off.

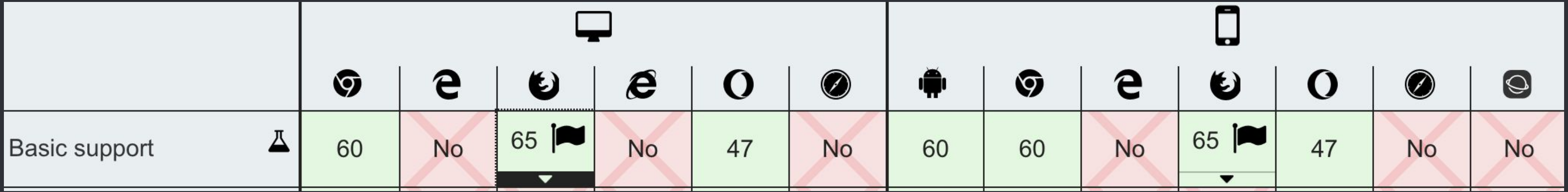
Allows or denies browser features

“Like CSP for features!”

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy

https://developers.google.com/web/updates/2018/06/feature-policy

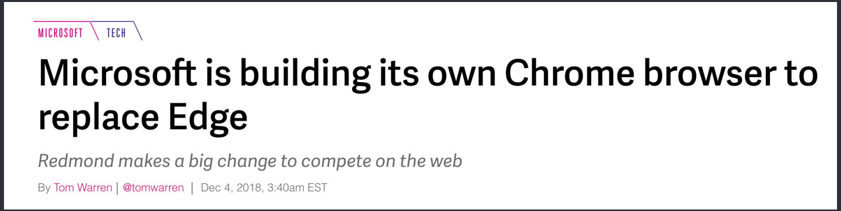
Only Chrome!



	Chrome	Edge	Firefox	IE	Opera	Safari	Android	Chrome Android	Edge Mobile	Firefox Android	Opera Android	Safari iOS	Samsung Internet
Basic support	60	No	65	No	47	No	60	60	No	65	47	No	No

Firefox is off by default. Set `dom.security.featurePolicy.header.enabled` to `true`

But everything's Chrome? ¯_(ツ)_/¯

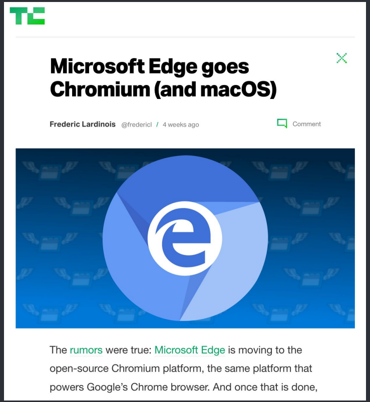


MICROSOFT \ TECH

Microsoft is building its own Chrome browser to replace Edge

Redmond makes a big change to compete on the web

By Tom Warren | @tomwarren | Dec 4, 2018, 3:40am EST



Microsoft Edge goes Chromium (and macOS)

Frederic Lardinois @fredericl / 4 weeks ago Comment

The rumors were true: Microsoft Edge is moving to the open-source Chromium platform, the same platform that powers Google's Chrome browser. And once that is done,

https://blogs.windows.com/windowsexperience/2018/12/06/microsoft-edge-making-the-web-better-through-more-open-source-collaboration/

@SecTinkerer

`content-security-policy`

Implemented in Chrome 25, Firefox 23, IE 10

Implemented across browsers Sept 2012– April 2013

Supersedes the previous x-xss-protection, x-frame-options headers

https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

What’s the bad thing?

Somewhere, somehow, something on your site that you don’t want running is there

CSP let’s you lock down content that page can load to a whitelist

And it will report back to you anything that’s in violation

What content? Any content!

- URLs that are loaded with script interfaces
- Fonts
- Frames
- Form submission endpoints
- What plugins can run
- Images
- Manifest files
- What <audio> <video> and <track> elements can load
- What <object>, <embed> and <applet> elements can load
- JavaScript
- CSS
- What can embed the page
- Require subresource integrity
- Restrict the set of base-uri
- Block HTTP content when loaded over HTTPS
- Upgrade HTTP content to HTTPS
- And defaults for everything else

Let’s build from the ground up

```
content-security-policy: default-src 'self';
report-uri /csp/report
```

- Only load content from the same origin (no subdomains!) anything else send a report

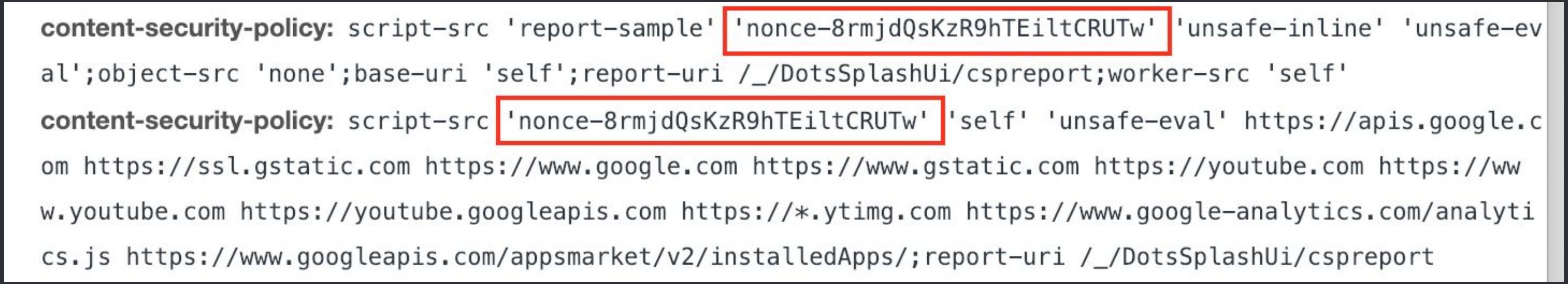
```
content-security-policy: default-src 'self';
connect-src: www.google-analytics.com 'self';
report-uri /csp/report
```

- Adds the ability for Google Analytics script connect and report back

```
content-security-policy: default-src 'self';
connect-src: www.google-analytics.com 'self';
script-src: 'unsafe-inline' connect.facebook.net 'self';
report-uri /csp/report
```

- Adds the ability to run inline JavaScript and a script from connect.facebook.net

What's real life? (news.google.com)



```
content-security-policy: script-src 'report-sample' 'nonce-8rmjdQsKzR9hTEiltCRUTw' 'unsafe-inline' 'unsafe-ev
al';object-src 'none';base-uri 'self';report-uri /_/DotsSplashUi/cspreport;worker-src 'self'
content-security-policy: script-src 'nonce-8rmjdQsKzR9hTEiltCRUTw' 'self' 'unsafe-eval' https://apis.google.c
om https://ssl.gstatic.com https://www.google.com https://www.gstatic.com https://youtube.com https://ww
w.youtube.com https://youtube.googleapis.com https://*.ytimg.com https://www.google-analytics.com/analyti
cs.js https://www.googleapis.com/appsmarket/v2/installedApps/;report-uri /_/DotsSplashUi/cspreport
```

Specifying a nonce (a unique per request, base64 value string) makes modern browsers ignore "unsafe-inline"

```
<script nonce="8rmjdQsKzR9hTEiltCRUTw"> … </script>
```

Yes, you can have multiple content-security-policy headers. Additions can only restrict. Most strict policy wins.

Other ways to run inline

Hashes! Take your script, drop <script> tags and hash

(capitals and whitespace matters!)

```
<script>var inline = 1;</script>
content-security-policy: script-src
'sha256-B2yPHKaXnvFWtRChIbabYmUBFZdVfKKXHbWtWidDVF8='
```

(Just pull the hash from the browser console error)

Similar to subresource integrity, where you specify a hash for a third party scripts

```
<script src="https://example.com/example-framework.js"
integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC"
crossorigin="anonymous"></script>
```

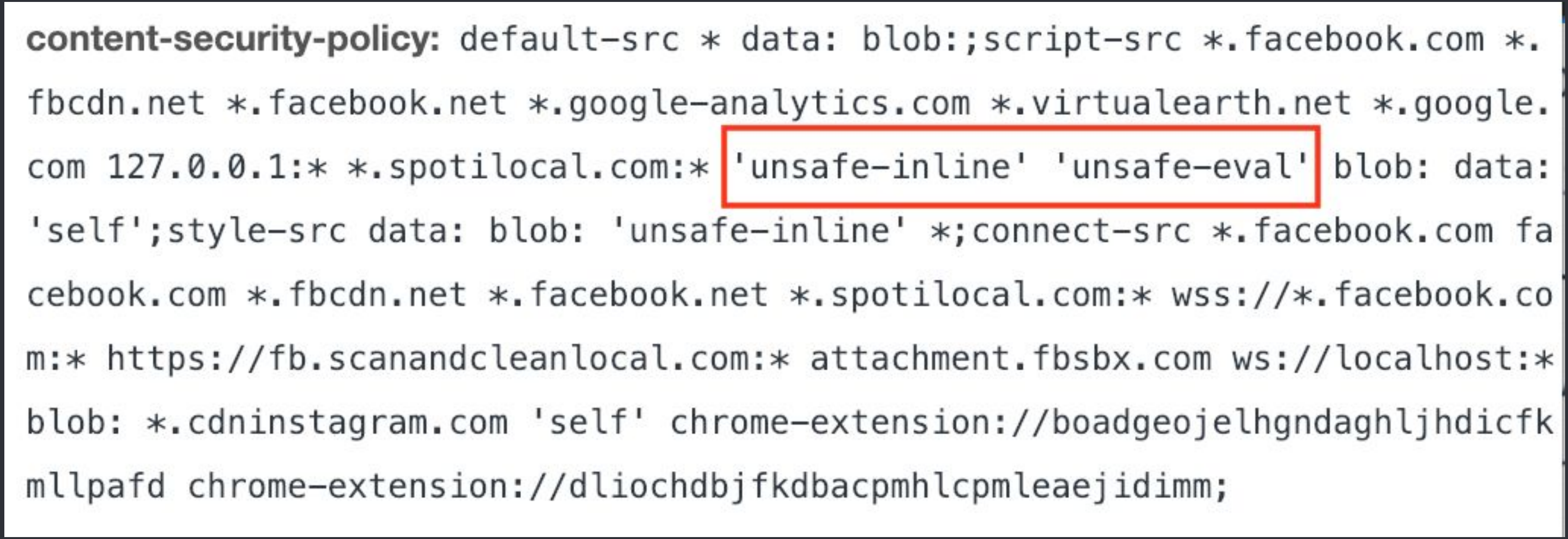
(BTW, you can force SRI with CSP!)

```
content-security-policy: require-sri-for script;
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/script-src

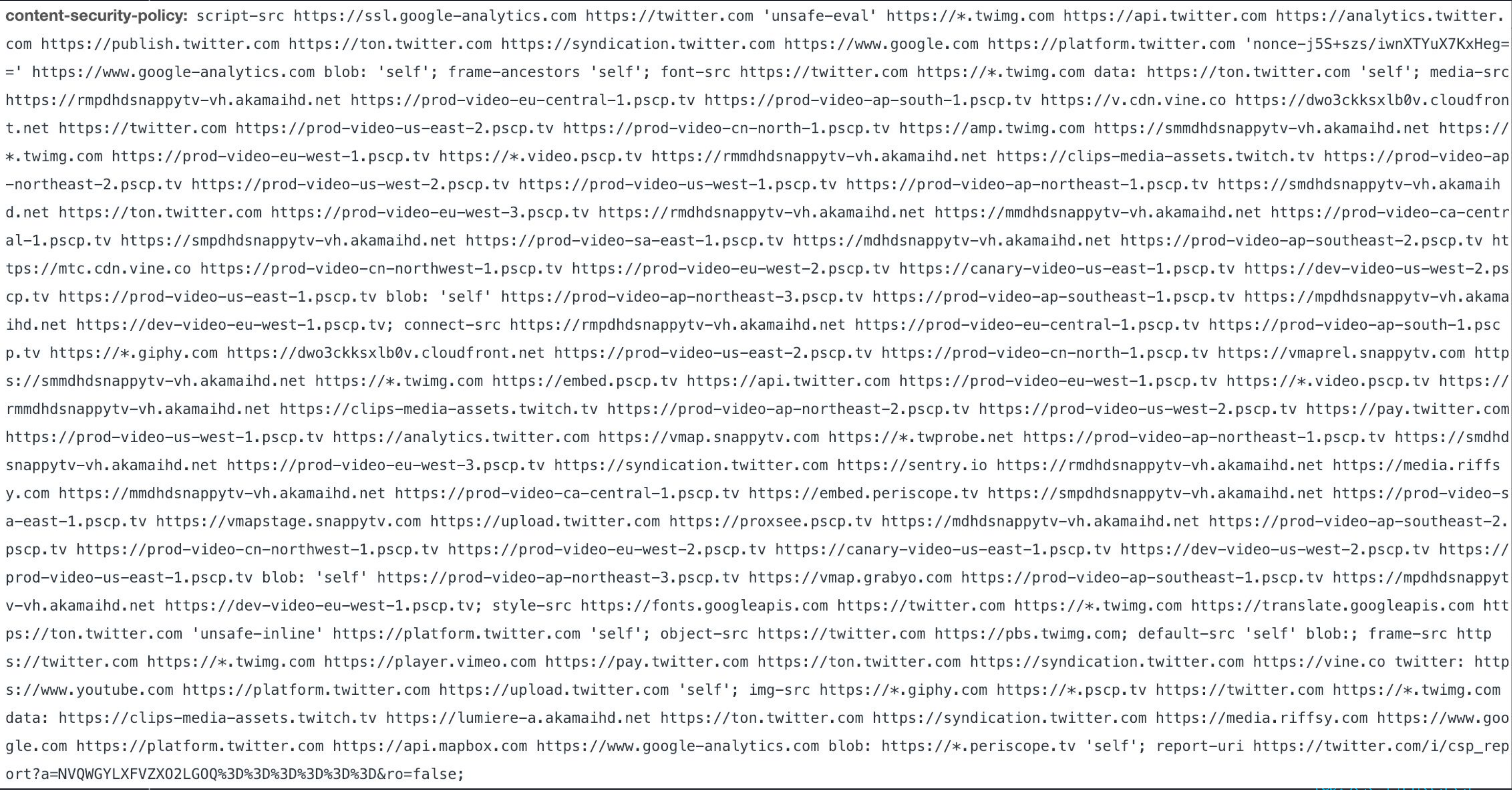
https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity

What’s real life? (facebook.com)



```
content-security-policy: default-src * data: blob:;script-src *.facebook.com *.
fbcdn.net *.facebook.net *.google-analytics.com *.virtualearth.net *.google.
com 127.0.0.1:* *.spotilocal.com:* 'unsafe-inline' 'unsafe-eval' blob: data:
'self';style-src data: blob: 'unsafe-inline' *;connect-src *.facebook.com fa
cebook.com *.fbcdn.net *.facebook.net *.spotilocal.com:* wss://*.facebook.co
m:* https://fb.scanandcleanlocal.com:* attachment.fbsbx.com ws://localhost:*
blob: *.cdninstagram.com 'self' chrome-extension://boadgeojelhgndaghljhdicfk
mllpafd chrome-extension://dliochdbjfkdbacpmhlcpmleaejidimm;
```

@SecTinkerer



```
content-security-policy: script-src https://ssl.google-analytics.com https://twitter.com 'unsafe-eval' https://*.twimg.com https://api.twitter.com https://analytics.twitter.
com https://publish.twitter.com https://ton.twitter.com https://syndication.twitter.com https://www.google.com https://platform.twitter.com 'nonce-j5S+szs/iwnXTYuX7KxHeg=
=' https://www.google-analytics.com blob: 'self'; frame-ancestors 'self'; font-src https://twitter.com https://*.twimg.com data: https://ton.twitter.com 'self'; media-src
https://rmpdhdsnappytv-vh.akamaihd.net https://prod-video-eu-central-1.pscp.tv https://prod-video-ap-south-1.pscp.tv https://v.cdn.vine.co https://dwo3ckksxlb0v.cloudfron
t.net https://twitter.com https://prod-video-us-east-2.pscp.tv https://prod-video-cn-north-1.pscp.tv https://amp.twimg.com https://smmdhdsnappytv-vh.akamaihd.net https://
*.twimg.com https://prod-video-eu-west-1.pscp.tv https://*.video.pscp.tv https://rmmdhdsnappytv-vh.akamaihd.net https://clips-media-assets.twitch.tv https://prod-video-ap
-northeast-2.pscp.tv https://prod-video-us-west-2.pscp.tv https://prod-video-us-west-1.pscp.tv https://prod-video-ap-northeast-1.pscp.tv https://smdhdsnappytv-vh.akamaih
d.net https://ton.twitter.com https://prod-video-eu-west-3.pscp.tv https://rmdhdsnappytv-vh.akamaihd.net https://mmdhdsnappytv-vh.akamaihd.net https://prod-video-ca-centr
al-1.pscp.tv https://smpdhdsnappytv-vh.akamaihd.net https://prod-video-sa-east-1.pscp.tv https://mdhdsnappytv-vh.akamaihd.net https://prod-video-ap-southeast-2.pscp.tv ht
tps://mtc.cdn.vine.co https://prod-video-cn-northwest-1.pscp.tv https://prod-video-eu-west-2.pscp.tv https://canary-video-us-east-1.pscp.tv https://dev-video-us-west-2.ps
cp.tv https://prod-video-us-east-1.pscp.tv blob: 'self' https://prod-video-ap-northeast-3.pscp.tv https://prod-video-ap-southeast-1.pscp.tv https://mpdhdsnappytv-vh.akama
ihd.net https://dev-video-eu-west-1.pscp.tv; connect-src https://rmpdhdsnappytv-vh.akamaihd.net https://prod-video-eu-central-1.pscp.tv https://prod-video-ap-south-1.psc
p.tv https://*.giphy.com https://dwo3ckksxlb0v.cloudfront.net https://prod-video-us-east-2.pscp.tv https://prod-video-cn-north-1.pscp.tv https://vmaprel.snappytv.com http
s://smmdhdsnappytv-vh.akamaihd.net https://*.twimg.com https://embed.pscp.tv https://api.twitter.com https://prod-video-eu-west-1.pscp.tv https://*.video.pscp.tv https://
rmmdhdsnappytv-vh.akamaihd.net https://clips-media-assets.twitch.tv https://prod-video-ap-northeast-2.pscp.tv https://prod-video-us-west-2.pscp.tv https://pay.twitter.com
https://prod-video-us-west-1.pscp.tv https://analytics.twitter.com https://vmap.snappytv.com https://*.twprobe.net https://prod-video-ap-northeast-1.pscp.tv https://smdhd
snappytv-vh.akamaihd.net https://prod-video-eu-west-3.pscp.tv https://syndication.twitter.com https://sentry.io https://rmdhdsnappytv-vh.akamaihd.net https://media.riffs
y.com https://mmdhdsnappytv-vh.akamaihd.net https://prod-video-ca-central-1.pscp.tv https://embed.periscope.tv https://smpdhdsnappytv-vh.akamaihd.net https://prod-video-s
a-east-1.pscp.tv https://vmapstage.snappytv.com https://upload.twitter.com https://proxsee.pscp.tv https://mdhdsnappytv-vh.akamaihd.net https://prod-video-ap-southeast-2.
pscp.tv https://prod-video-cn-northwest-1.pscp.tv https://prod-video-eu-west-2.pscp.tv https://canary-video-us-east-1.pscp.tv https://dev-video-us-west-2.pscp.tv https://
prod-video-us-east-1.pscp.tv blob: 'self' https://prod-video-ap-northeast-3.pscp.tv https://vmap.grabyo.com https://prod-video-ap-southeast-1.pscp.tv https://mpdhdsnappyt
v-vh.akamaihd.net https://dev-video-eu-west-1.pscp.tv; style-src https://fonts.googleapis.com https://twitter.com https://*.twimg.com https://translate.googleapis.com htt
ps://ton.twitter.com 'unsafe-inline' https://platform.twitter.com 'self'; object-src https://twitter.com https://pbs.twimg.com; default-src 'self' blob:; frame-src http
s://twitter.com https://*.twimg.com https://player.vimeo.com https://pay.twitter.com https://ton.twitter.com https://syndication.twitter.com https://vine.co twitter: http
s://www.youtube.com https://platform.twitter.com https://upload.twitter.com 'self'; img-src https://*.giphy.com https://*.pscp.tv https://twitter.com https://*.twimg.com
data: https://clips-media-assets.twitch.tv https://lumiere-a.akamaihd.net https://ton.twitter.com https://syndication.twitter.com https://media.riffsy.com https://www.goo
gle.com https://platform.twitter.com https://api.mapbox.com https://www.google-analytics.com blob: https://*.periscope.tv 'self'; report-uri https://twitter.com/i/csp_rep
ort?a=NVQWGYLXFVZXO2LGOQ%3D%3D%3D%3D%3D%3D&ro=false;
```

Phew, glad we have a security/SRE/foobar team to worry about this mess!

`content-security-policy` *only exists to break things!*

A strict `content-security-policy` can functionally only be set by the people that write the code. They know the `content`!

Here's what's worked for me

- Set a global failback for legacy applications.
- Fail open? Add `default-src *;` Even correctly configured `content-security-policy` can break slower adopting and older browsers
- Leverage `content-security-policy-report-only`

Fail open or fail close?

Let’s assume a new CSP directive foobar-src

Updated browsers will enforce foobar-src
Older browsers will fall back to default-src

```
content-security-policy: default-src: *;
foobar-src: super-foo.com 'self';
report-uri /csp/report
```

```
content-security-policy: default-src: 'none';
foobar-src: super-foo.com 'self';
report-uri /csp/report
```

A perfectly to spec CSP can still break things!

`content-security-policy-report-only`

Basically “audit mode” for `content-security-policy`

Displays violations in the browser console, but won’t block functionality

Syntax is identical; only appending `-report-only` to the header name

Reports violations via report-uri

Example Report

JSON formatted POST to your report URI

```
{
  "csp-report": {
    "document-uri": "http://example.com/signup.html",
    "referrer": "",
    "blocked-uri": "http://example.com/css/style.css",
    "violated-directive": "style-src cdn.example.com",
    "original-policy": "default-src 'none'; style-src
cdn.example.com; report-uri /csp-reports",
    "disposition": "report/enforce"
  }
}
```

@SecTinkerer

Questions? Comments? Snide Remarks?

Go to `https://sli.do`

Event code `#BSidesSF2019`

Select **`City View Track`**

inbox@benjamin-hering.com

@SecTinkerer