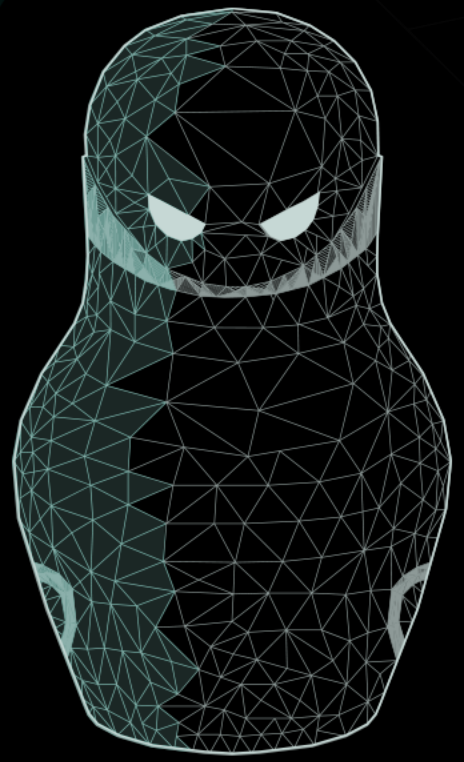


ZERO NIGHTS 2015

Getting The Most Out Of CSP: A Deep Dive

Sergey Shekyan

Canadian Staging Professionals (CSP)

CSP CERTIFIED STAGING PROFESSIONALS™

- CSP® Home Staging Certification Course Is Simply The Best Education and Support For Your New Career
- Mastering the Art of Difficult Conversations with Sellers
- Chemicals Hiding In Plain Sight
- Environmentally Responsible Home Design and Remodeling

Content Security Policy

Sike!

Content Security Policy

is a mechanism to declare content restrictions for web resources in a browser

- disables normally enabled capabilities in web pages
- helps to detect and prevent XSS, UI Redressing, mixed-content and other attacks

ZERO NIGHTS 2015

History

Revision history of "Security/CSP"

View logs for this page

Browse history

From year (and earlier): 2009 From month (and earlier): March Tag filter: Go

Diff selection: Mark the radio boxes of the revisions to compare and hit enter or the button at the bottom.
Legend: **(cur)** = difference with latest revision, **(prev)** = difference with preceding revision, **m** = minor edit.

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

Compare selected revisions

- (cur | prev) 15:22, 16 March 2009 Sidstamm (Talk | contribs) . . (460 bytes) (+38) . . *(→Tasks to Complete)*
- (cur | prev) 15:21, 16 March 2009 Sidstamm (Talk | contribs) . . (422 bytes) (+123) . . *(→Welcome)*
- (cur | prev) 10:53, 11 March 2009 Bsterne (Talk | contribs) . . (299 bytes) (+299) . . *(New page: == Welcome == This page is a placeholder for now and will be expanded as decisions are made about Content Security Policy. == Tasks to Complete == * Finish documentation of restrictions *...)*

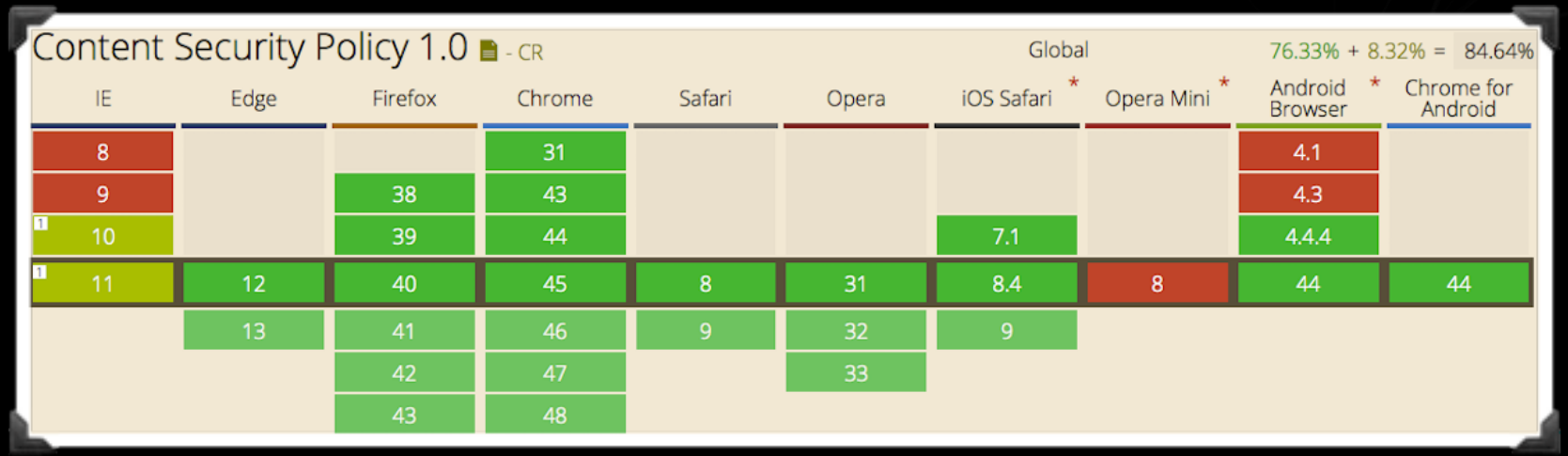
Compare selected revisions

(newest | oldest) View (newer 50 | older 50) (20 | 50 | 100 | 250 | 500)

- Initial proposal by Brandon Sterne - 2009
- CSP level 1 - Nov 2012
- CSP level 2 - July 2014
- CSP level 3 will introduces A LOT of changes

ZERO NIGHTS 2015

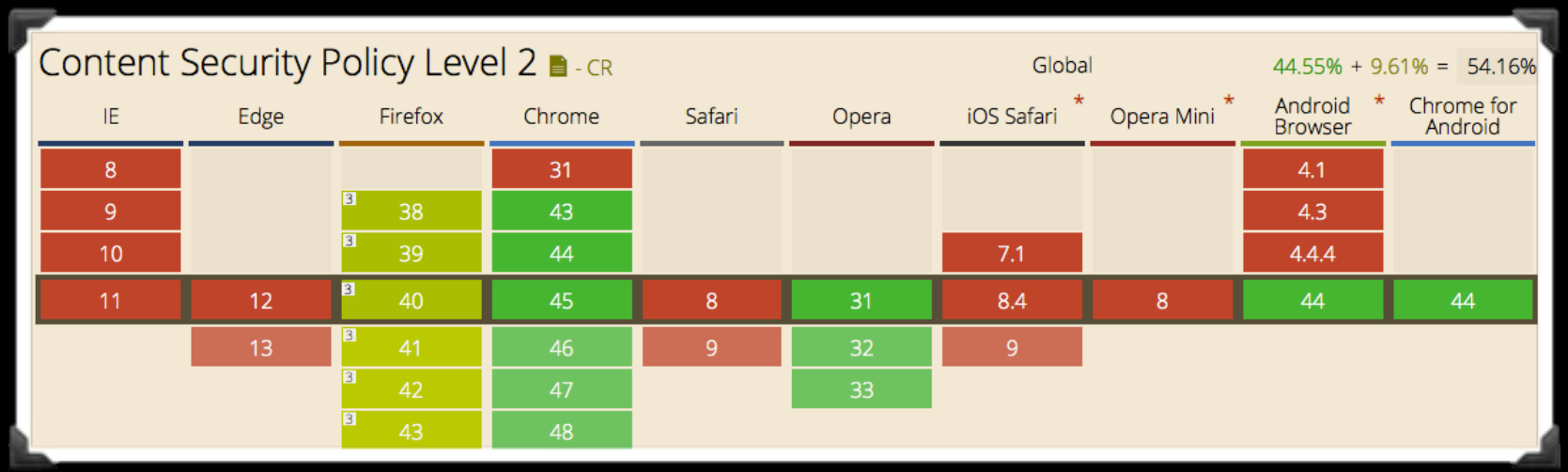
Adoption: Browsers



Content Security Policy 1.0 - CR

Global 76.33% + 8.32% = 84.64%

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			31					4.1	
9		38	43					4.3	
10 [1]		39	44			7.1		4.4.4	
11 [1]	12	40	45	8	31	8.4	8	44	44
	13	41	46	9	32	9			
		42	47		33				
		43	48						



Content Security Policy Level 2 - CR

Global 44.55% + 9.61% = 54.16%

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			31					4.1	
9		38 [3]	43					4.3	
10		39 [3]	44			7.1		4.4.4	
11	12	40 [3]	45	8	31	8.4	8	44	44
	13	41 [3]	46	9	32	9			
		42 [3]	47		33				
		43 [3]	48						

Adoption: Websites

Less than **0.2%** of Alexa top 100000 use CSP on landing page

- many nonsensical policies
- many invalid policies
- few web properties use nonce-source/hash-source
- most use `'unsafe-inline'` and `'unsafe-eval'`
- some use deprecated policies

Content Security Policy

source-list

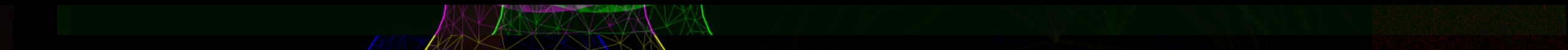
Delivered over HTTP header

```
Content-Security-Policy: script-src https: example.com 'self'
```

directive-name scheme-source host-source keyword-source

Delivered over meta tag

```
<meta http-equiv="Content-Security-Policy" content="script-src https: example.com; 'self'">
```

ZERO NIGHTS 2015

Combining source expressions

```
script-src 'self' https: *.example.com
```

<= **served by http://www.site.com**

A URL url is said to match a source list for protected resource if at ***least* one** source expression in the set of source expressions obtained by parsing the source list matches url for protected resource.

```
<script src='http://www.site.com'>  ✔
<script src='http://api.site.com'>  ✘
<script src='https://google.com'>   ✔
<script src='http://example.com'>   ✘
<script src='http://api.example.com'> ✔
<script src='http://api.example.com'> ✔
```

ZERO NIGHTS 2015

Matching

```
script-src https:;
```

```
<script src='https://a.com'>
<script src='https://a.com:8080'>
<script src='ws://a.com:80'>
```

```
script-src a.com;
```

```
<script src='https://a.com'>
<script src='http://a.com:8080'>
<script src='http://a.com'>
```

```
script-src a.com/b;
```

```
<script src='http://a.com/b'>
<script src='http://a.com/bbb'>
<script src='http://a.com/b/'>
```

```
script-src a.com/b/;
```

```
<script src='https://a.com/b'>
<script src='https://a.com/b/a'>
<script src='https://a.com/b/'>
```

ZERO NIGHTS 2015

Directives

source-list

- default-src
- child-src
- script-src
- frame-src
- style-src
- object-src
- img-src
- connect-src
- font-src
- media-src
- form-action
- base-uri

ancestor-source-list

- frame-ancestors

media-type-list

- plugin-types

uri-reference

- report-uri

ZERO NIGHTS 2015

Dealing with keywords

Good

```
script-src 'none';
style-src 'none';
```

HTML

```
<script src='evil.com'></script>
<body onclick='alert(1337)'>
<script>eval(evilCode);<script>
```

Bad

```
script-src * 'unsafe-inline';
style-src * 'unsafe-inline'
```

HTML

```
<script src='evil.com'></script>
<body onclick='alert(1337)'>
<script>eval(evilCode);<script>
```

Bad

```
script-src * 'unsafe-eval';
style-src * 'unsafe-eval'
```

HTML

```
<script src='evil.com'></script>
<body onclick='alert(1337)'>
<script>eval(evilCode);<script>
```

Granular whitelisting

```
script-src 'sha256-nnn';
style-src 'sha256-mmm';
```

```
<script src='evil.com'></script>
<body onclick='alert(1337)'>
<script>eval(evilCode);<script>
<script>goodCode()<script>
```

HTML

Use hash (sha256, sha-384, sha-512) if have to have inline scripts or styles. Good for whitelisting static content.

```
script-src 'nonce-123';
style-src 'nonce-123';
```

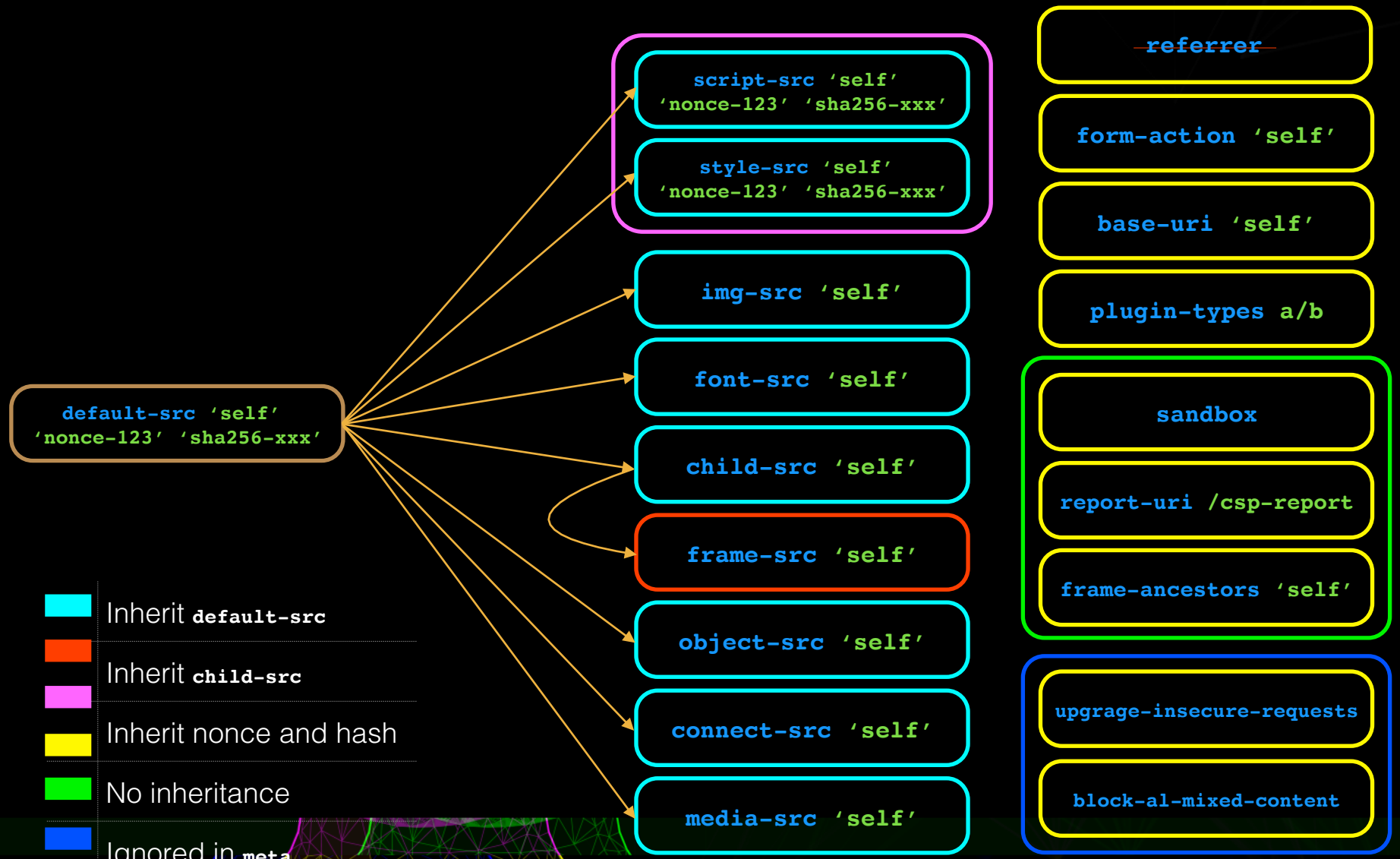
```
<script src='evil.com'></script>
<body onclick='alert(1337)'>
<script>eval(evilCode);<script>
<script nonce='123'>goodCode()<script>
```

HTML

Use nonce if have to have inline scripts or styles. Good for whitelisting dynamic content.

ZERO NIGHTS 2015

CSP2 Cheatsheet



ZERO NIGHTS 2015

Combining source lists

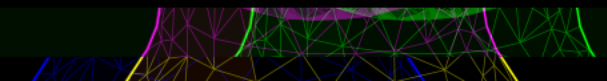
When multiple policies are present, the resource must match **all** of them

```
Content-Security-Policy: script-src *.example.com;
Content-Security-Policy: script-src 'self';
Content-Security-Policy: script-src https: report-uri /report;
```

==

Combine multiple headers into one by comma-separated

```
Content-Security-Policy: script-src *.example.com, script-src 'self',
script-src https: report-uri /report;
```



ZERO NIGHTS 2015

Combining source lists

```
Content-Security-Policy: script-src *.example.com, script-src 'self',
script-src https: report-uri /report;
```

!=

If you instead try to combine policies by combining source lists of given directives, the resulting policy will behave differently

```
Content-Security-Policy: script-src *.example.com 'self' https:;
report-uri /report;
```

```
Content-Security-Policy: script-src *.example.com, script-src 'self',
script-src https: report-uri /report;
```

==

Match all, remember? do not forget about reporting context

```
Content-Security-Policy: 'none'; report-uri /report;
```

Grammar & Semantics

```
default-src a b c;
script-src d;
style-src d;
img-src d;
font-src d;
child-src d;
connect-src d;
object-src d;
```

==

```
default-src d
```

frame-src is missing. Are there policies identical? Yes, because:
frame-src is always derived from child-src, if not explicitly specified
frame-src => **frame-src** *

Grammar & Semantics

```
script-src 'none'
```

==

```
script-src
```

```
default-src 'none'
```

==

```
default-src
```

ZERO NIGHTS 2015

Defaults in scheme and port

```
script-src http://example.com:80
```

==

```
script-src http://example.com
```

==

```
script-src example.com
```

Default scheme – 4.2.2 Matching Source Expressions
If the source expression does not have a scheme, return does not match if any of the following are true:
the scheme of the protected resource's URL is a case insensitive match for HTTP, and url-scheme is not a case insensitive match for either HTTP or HTTPS
the scheme of the protected resource's URL is not a case insensitive match for HTTP, and url-scheme is not a case insensitive match for the scheme of the protected resource's URL.
In non-spec English, default scheme is `http`

Default port	
ftp	21
file	
gopher	70
http	80
https	443
ws	80
was	443

Most permissive policy

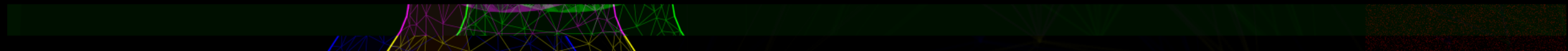
==

```
Content-Security-Policy:
```

==

```
Content-Security-Policy: default-src * ‘unsafe-inline’ ‘unsafe-eval’
data: filesystem: blob:
```

Missing Content-Security doesn’t touch any web page capabilities



Most restrictive policy

```
Content-Security-Policy: default-src; frame-ancestors 'none'; form-action
                         'none'; sandbox
```

CSP host-source

`connect-src 'self' https:;` **served by https://example.com**

`XMLHttpRequest` **to http://example.com** ✗

`XMLHttpRequest` **to https://example.com:8080** ✗

`XMLHttpRequest` **to https://example.com** ✔

`XMLHttpRequest` **to https://example.com:443** ✔

`WebSocket` **to ws://example.com:80** ✗

`WebSocket` **to wss://example.com** ✗

CSP Examples

```
default-src 'self' ;
script-src 'unsafe-inline' 'unsafe-eval' 'self' * ;
style-src 'unsafe-inline' 'self' * ;
frame-src 'self' * ;
object-src 'self' * ;
img-src 'self' data: * ;
media-src 'self' * ;
font-src 'self' * ;
connect-src 'self' *
```

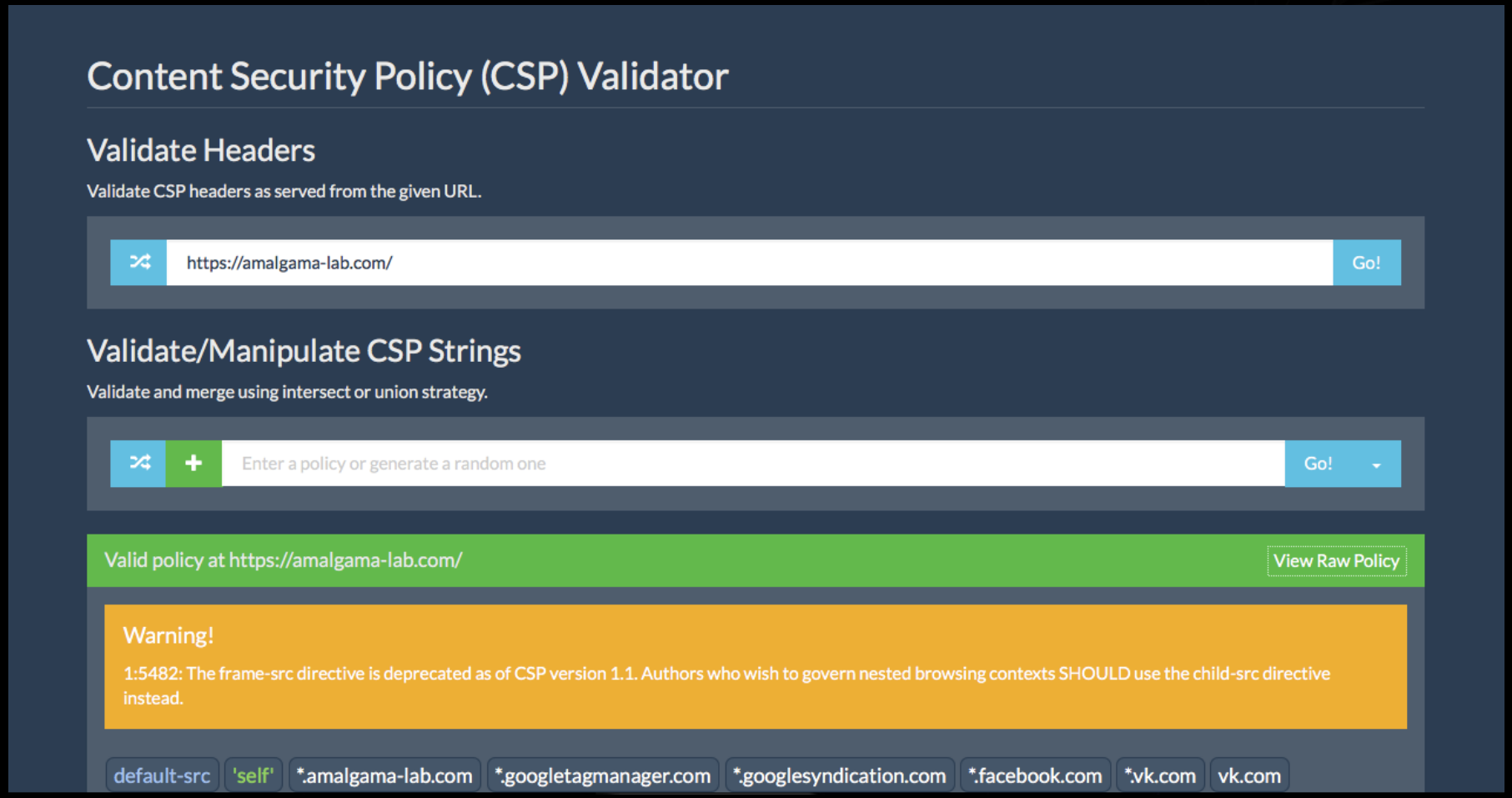
Bad: **default-src * ‘unsafe-inline’ + ‘unsafe-eval’** + allows **data:** for image

ZERO NIGHTS 2015

CSP Examples

ZERO NIGHTS 2015

CSP Examples



CSP Examples

https://apis.google.com *.google.com *.google.ru *.google.com.ua *.google.ca *.twitter.com *.youtube.com https://www.youtube.com
*.odnoklassniki.ru *.rutarget.ru *.adriver.ru *.doubleclick.net *.bidswitch.net *.yandex.ru https://accounts.google.com
https://googleads.g.doubleclick.net https://s-static.ak.facebook.com https://www.facebook.com https://login.vk.com ;
script-src 'self' 'unsafe-inline' 'unsafe-eval' *.amalgama-lab.com *.google.ru http://*.google.ru http://cse.google.ru *.google.ru:*
https://*.google.ru:* *.google.com http://*.google.com *.google.com.ua *.google.com:* https://*.google.com:* *.google.ca
https://apis.google.com ajax.googleapis.com https://*.googleapis.com *.googleapis.com *.mixmarket.biz 4294953203.kt.mixmarket.biz
*.revsci.net *.republer.com http://yastatic.net https://*.republer.com *.gstatic.com https://*.gstatic.com *.yandex.net
https://*.yandex.net *.yandex.st *.yandex.ru https://*.yandex.ru yastatic.net *.yastatic.net *.yastatic.net:* https://*.yastatic.net
mc.yandex.ru https://mc.yandex.ru an.yandex.ru https://an.yandex.ru *.yandex.ua https://*.googlesyndication.com
http://*.googlesyndication.com https://googleads.g.doubleclick.net https://*.doubleclick.net *.doubleclick.net *.luxup.ru luxup.ru
*.luxadv.com *.luxcdn.com vk.com *.twitter.com connect.facebook.net graph.facebook.com connect.mail.ru *.pinterest.com
*.advertur.ru advertur.ru *.betweendigital.com counter.rambler.ru www.googletagservices.com www.googletagmanager.com
*.google-analytics.com *.googleadservices.com https://*.googleadservices.com yadro.ru https://yadro.ru *.yadro.ru https://*.yadro.ru
go.youlamedia.com *.adlabs.ru *.adriver.ru *.kavanga.ru https://*.kavanga.ru *.ok.ru ok.ru *.sociomantic.com *.odnoklassniki.ru
.googleusercontent.com https://.google-analytics.com https://google-analytics.com *.imrkcrv.net *.imrk.net *.infostatsvc.com
*.avocet.io ;
img-src 'self' data: *.amalgama-lab.com *.mixmarket.biz *.rutarget.ru *.smartyads.com https://*.republer.com sync.republer.com
*.revsci.net yastatic.net *.yastatic.net *.adhigh.net *.datamind.ru counter.yadro.ru *.ytimg.com *.yandex.net *.yandex.st *.yandex.ru
*.yandex.ua *.google.am *.google.it *.google.jo *.google.cz *.google.ch *.google.no *.google.ae *.google.at *.google.az *.google.bg

ZERO NIGHTS 2015

CSP Examples

*.google.me *.google.hr *.google.co.id *.google.co.il *.google.dk *.google.fr *.google.ge *.google.nl *.google.rs *.google.gr
*.google.ba *.google.ro *.google.dz *.google.se *.google.sk *.google.md *.google.ee *.google.lv *.google.pl *.google.tn *.google.es
*.google.si *.google.com.kw *.google.com.ng *.google.lu *.google.ca *.google.lt *.google.com.tr *.google.de *.google.com.cy
*.google.com.au *.google.com.sg *.google.com.eg *.google.co.uk *.google.co.jp *.google.co.in *.google.co.ve *.google.com.kh
*.google.com.mt *.google.com.np *.google.com.pe *.google.com.ph *.google.fi *.google.hu *.google.mn *.google.pt *.google.com.mx
*.google.com.tj *.google.co.kr *.google.co.th *.google.com.tw *.pinterest.com vk.com *.googleusercontent.com
.googlesyndication.com https://.googlesyndication.com https://pagead2.googlesyndication.com www.google-analytics.com *.google.com
*.google.ru *.google.by *.google.com.ua *.google.be *.google.ie *.google.com.ar *.google.kz *.gstatic.com https://*.gstatic.com
counter.rambler.ru *.betweendigital.com *.luxup.ru *.luxup.ru:* *.imrk.net *.adlabs-retail.ru go.youlamedia.com *.gravatar.com
front.facetz.net ad.dumedia.ru rtb.rtcdn.ru g.mp.luxcdn.com *.rtb-media.ru *.whisla.com *.adlabs.ru *.recreativ.ru *.mail.ru
*.tns-counter.ru x.bidswitch.net *.luxadv.com *.sociomantic.com *.adriver.ru *.kavanga.ru b.kavanga.ru https://*.kavanga.ru
*.hubrus.com *.doubleclick.net https://stats.g.doubleclick.net https://*.doubleclick.net https://googleads.g.doubleclick.net *.2mdn.net
.googleapis.com https://.googleapis.com https://*.2mdn.net *.madnet.ru *.spotxchange.com https://*.spotxchange.com
https://*.adform.net *.republer.com *.intencysrv.com *.googleadservices.com *.yahoo.com *.admitad.com 109.206.167.217
109.206.167.109 mc.yandex.ru mc.yandex.ru:* https://mc.yandex.ru https://*.yandex.ru *.google-analytics.com google-analytics.com
https://*.google-analytics.com https://google-analytics.com *.rontar.com *.paradocs.ru *.uptolike.com *.adap.tv *.r24-tech.com
*.adnxs.com *.rubiconproject.com *.btrll.com *.adtech.de *.liverail.com *.adadvisor.net *.openx.com *.upstats.ru *.advertising.com
*.adsniper.ru *.lijit.com *.openx.net *.adframesrc.com *.targetix.net *.trafex.net *.imrkcrv.net *.tovarro.com *.begun.ru *.avocet.io
*.advombat.ru *.smaclick.com *.yadro.ru https://*.yadro.ru *.liveinternet.ru https://*.liveinternet.ru ;

CSP Examples

style-src 'self' 'unsafe-inline' *.amalgama-lab.com *.gstatic.com https://*.gstatic.com https://fonts.googleapis.com fonts.googleapis.com *.google.com *.sociomantic.com yastatic.net *.yastatic.net *.yastatic.net:* https://*.yastatic.net ;
media-src 'self' *.amalgama-lab.com *.youtube.com youtube.com ;
object-src 'self' *.amalgama-lab.com http://imrkcrv.net *.imrkcrv.net *.adriver.ru *.gstatic.com *.googlesyndication.com https://*.googlesyndication.com https://*.gstatic.com *.datariver.ru *.kavanga.ru *.googlevideo.com googlevideo.com *.ytimg.com ytimg.com *.youtube.com youtube.com an.yandex.ru https://an.yandex.ru *.yandex.ru ;
frame-src 'self' *.amalgama-lab.com *.googlesyndication.com https://*.googlesyndication.com *.googleadservices.com https://*.googleadservices.com *.doubleclick.net https://stats.g.doubleclick.net https://*.doubleclick.net https://googleads.g.doubleclick.net *.google.ru *.google.com *.yandex.st *.imrk.net yastatic.net *.yastatic.net *.yastatic.net:* https://*.yastatic.net ;
font-src 'self' data: *.amalgama-lab.com *.gstatic.com *.googleusercontent.com *.googleapis.com https://*.googleapis.com https://*.gstatic.com ;
connect-src 'self' data: *.amalgama-lab.com mc.yandex.ru *.yandex.ru https://mc.yandex.ru www.google-analytics.com www.liveinternet.ru *.google-analytics.com google-analytics.com https://*.google-analytics.com https://google-analytics.com https://*.yandex.ru ;
report-uri http://www.amalgama-lab.com/csp/report0.php

CSP Validator was built by Sergey Shekyan, Michael Ficarra, Lewis Ellis, Ben Vinegar, and the fine folks at Shape Security.

CSP Examples

Worst policy ever?

```
allow 'self'; frame-ancestors 'self'
```

"The 'allow' directive has been replaced with 'default-src'. Please use that directive instead, as 'allow' has no effect."

Didn't make into CSP level 1.

"Unrecognized Content-Security-Policy directive 'frame-ancestors'."

Introduced by CSP level 2, replaces **`frame-src`**

ZERO NIGHTS 2015

Browser Handling of Bad Policies

```
script-src 'self' foo.com/\U0001F4A9
```

```
The value for Content Security Policy directive 'script-src' contains an invalid character: 'http://google.com/Ã£'. Non-whitespace characters outside ASCII 0x21-0x7E must be percent-encoded, as described in RFC 3986, section 2.1: http://tools.ietf.org/html/rfc3986#section-2.1.
```

Parsing error skips the whole directive

```
Content Security Policy: Couldn't parse invalid source http://google.com/Ð°
```

```
Content Security Policy: The page's settings blocked the loading of a resource at self ("script-src 'none'").
```

Parsing error skips only bad source source expression

Some of Security Bugs

```
script-src *.x.y
```

```
<script src='//x.y/file'>
```

Chromium bug 534542 - *.x.y must match host that ends with .x.y, but not x.y

```
script-src *
```

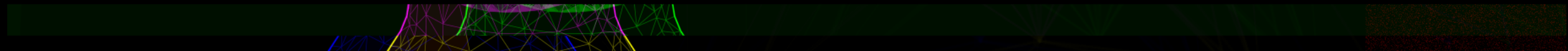
```
<script src='data:applicication/
javascript, alert(1337)'>
```

Chromium bug 534570 - wildcard (*) should not match data:, filesystem:, blob:

CSP Reporting

```
report-uri /csp-report /another-endpoint
```

- sends report to every report URL
- multiple policies do not share reporting context, meaning every policy violation will be reported to it's `report-uri`
- If the origin of report URL is the same as the origin of the protected resource, cookies will be sent too (except Firefox)
- there are as many report formats as there are browsers, so normalize



CSP Reporting

```
Content-Type: application/csp-report
{"csp-report":{
     "document-uri":"http://example.com",
    "referrer":"",
    "violated-directive":"script-src 'self'",
    "effective-directive":"script-src",
    "original-policy":"default-src 'none';script-src 'self' ;img-src 'self'; connect-src
'self';style-src 'self';frame-ancestors 'none';form-action 'self';report-uri /csp-report",
    "blocked-uri":"http://www.evil.com",
     “status-code":200
}}
```

```
Content-Type: application/json
{"csp-report":{
   "blocked-uri":"http://www.evil.com/animate.js",
   “document-uri”:”http://example.com“,
   "original-policy":"default-src 'none'; script-src http://example.com; img-srchttp://
example.com; connect-srchttp://example.com; style-src http://example.com; frame-
ancestors 'none'; form-action http://example.com; report-uri http://example.com/csp-report“,
   "referrer":"",
   "violated-directive":"script-srchttp://example.com",
   "user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:41.0) Gecko/20100101 Firefox/
41.0",
   "level":"info",
   "message":"",
   "timestamp":"2015-10-20T19:54:51.248Z"
}}
```

Reports Can Leak Data

- make report URI query string unique for segmentation and to avoid dealing with poisoned logs. It is harder to spoof `report-uri /csp-report?id=nonce`
- hacked server with big audience might become DDoS orchestrator

Evaluate before deploying

```
Content-Security-Policy-Report-Only:
'self'; report-uri /csp-report
```

- `Content-Securty-Policy-Report-Only` header monitors, reports but doesn't enforce
- Try new policy with `Content-Securty-Policy-Report-Only` along with your current policy delivered by `Content-Security-Policy` to catch problems before it's too late

Salvation

general purpose FOSS Java library supports:

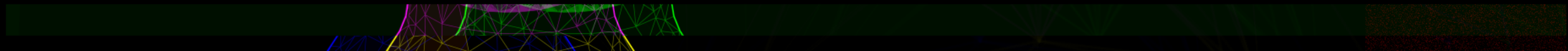
- warnings and syntax errors with position info
- policy optimisation
- merging using union or intersection strategyanswering questions like `allowsConnectTo`, `allowsScriptFromSource`, `allowsStyleWithHash`, `allowsChildFromSource`, etc

is used in production

the only CSP implementation outside of the browser

get it from:

- https://github.com/shapesecurity/salvation
- Maven Central



Merging CSP policies

- ***Union***, which is useful when crafting a policy, starting with a restrictive policy and allowing each resource that is needed.

```
default-src a b
```

U

```
default-src b c
```

==

```
default-src a b c
```

ZERO NIGHTS 2015

Merging CSP policies

- ***Intersection*** combines policies in such a way that the result will behave similar to how browsers enforce each policy individually.

```
default-src a b
```

∩

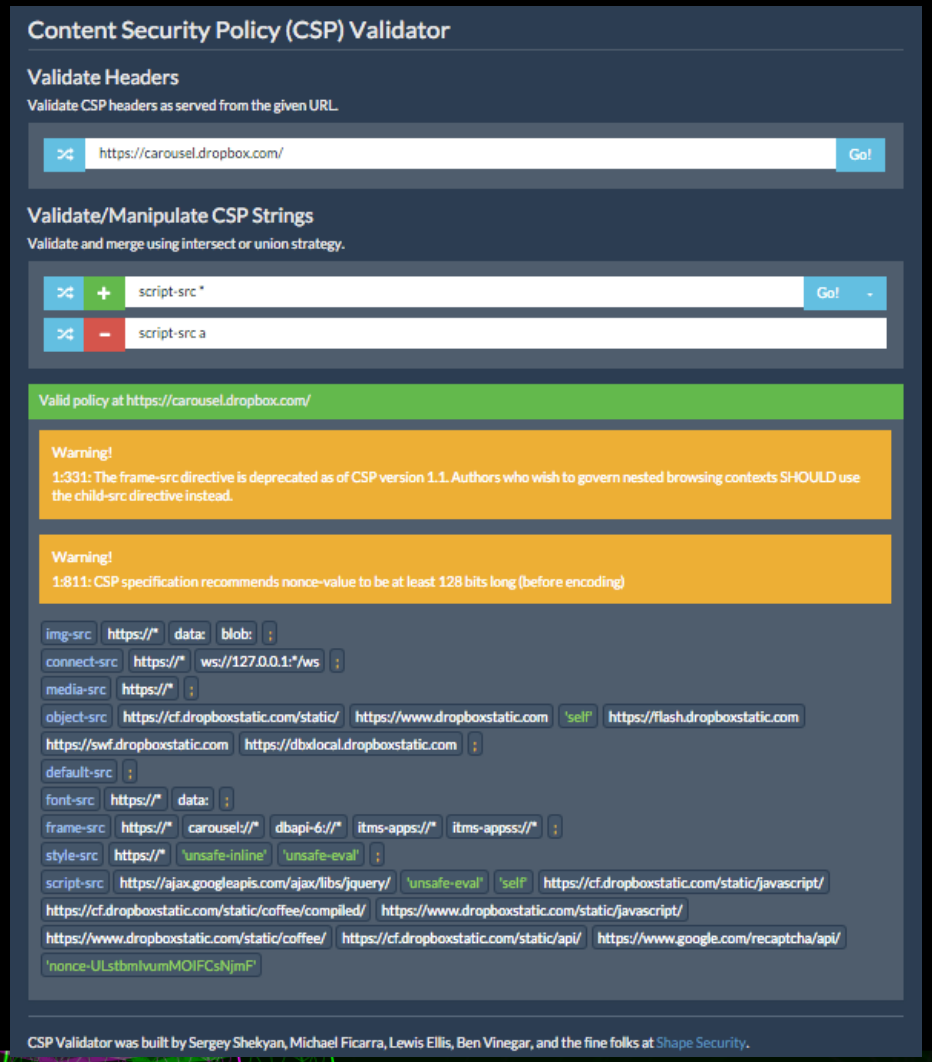
```
default-src; script-src *; style-src c
```

==

```
default-src; script-src a b
```

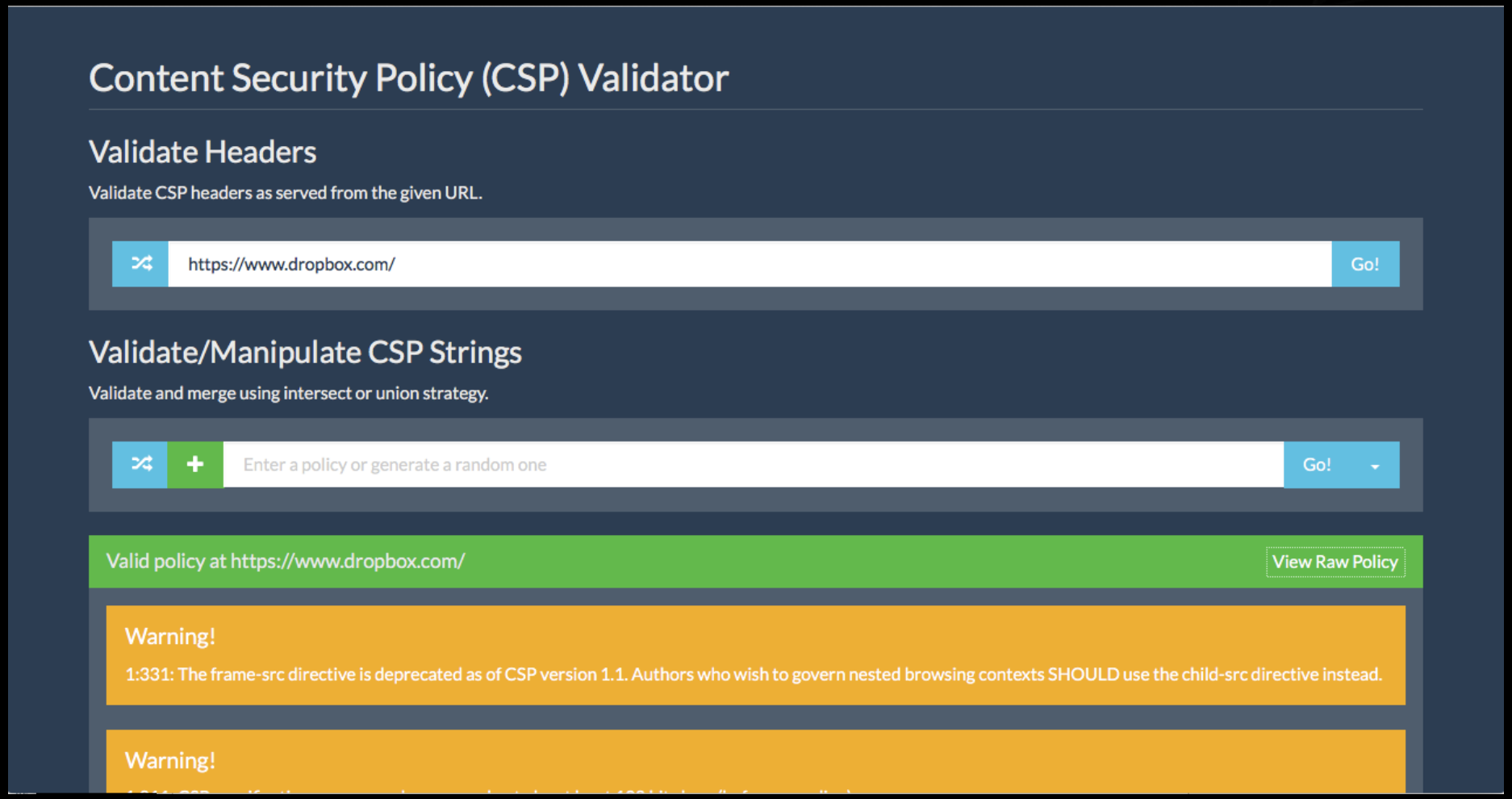
ZERO NIGHTS 2015

https://cspvalidator.org



ZERO NIGHTS 2015

https://cspvalidator.org



https://cspvalidator.org

Warning!

1:811: CSP specification recommends nonce-value to be at least 128 bits long (before encoding)

img-src	https://* data: blob: ;
connect-src	https://* ws://127.0.0.1:*/ws ;
media-src	https://* ;
object-src	https://cf.dropboxstatic.com/static/ https://www.dropboxstatic.com 'self' https://flash.dropboxstatic.com https://swf.dropboxstatic.com https://dbxlocal.dropboxstatic.com ;
default-src	;
font-src	https://* data: ;
frame-src	https://* carousel://* dbapi-6://* itms-apps://* itms-appss://* ;
style-src	https://* 'unsafe-inline' 'unsafe-eval' ;
script-src	https://ajax.googleapis.com/ajax/libs/jquery/ 'unsafe-eval' 'self' https://cf.dropboxstatic.com/static/javascript/ https://cf.dropboxstatic.com/static/coffee/compiled/ https://www.dropboxstatic.com/static/javascript/ https://www.dropboxstatic.com/static/coffee/ https://cf.dropboxstatic.com/static/api/ https://www.google.com/recaptcha/api/ 'nonce-4Kslap8Tije6SugZfxXZ'

CSP Validator was built by Sergey Shekyan, Michael Ficarra, Lewis Ellis, Ben Vinegar, and the fine folks at Shape Security.

Powered by Salvation, a Java library for working with CSP policies.

ZERO NIGHTS 2015

https://cspvalidator.org

- Exposes a subset of Salvation features
- validates, inspects, merges
- try random policy buttons 
- compare raw policy served by web site with optimised one 

- share policies
  - https://cspvalidator.org/#url=https://www.twitter.com/
  - https://cspvalidator.org/#headerValue%5B%5D=script-src+example.com&headerValue%5B%5D=&strategy=union
- Web app takes advantage of browser security features like HPKP, HSTS, CSP, security headers

How does good CSP policy look like?

```
default-src ;
script-src 'self' https://www.google-analytics.com
https://code.jquery.com https://
maxcdn.bootstrapcdn.com ;
img-src 'self' https://www.google-analytics.com ;
font-src 'self' https://fonts.gstatic.com https://
bootswatch.com ;
connect-src 'self' ;
style-src 'self' https://bootswatch.com https://
fonts.googleapis.com ;
frame-ancestors 'none' ;
form-action 'self' ;
report-uri https://cspvalidator.org/csp-report
```

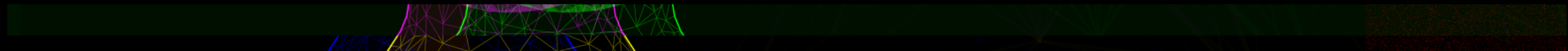
Contributing to CSP

While developing Salvation

- >10 bugs reported against the CSP specification
- >10 bugs reported against Chromium CSP implementation

Use Salvation!

Participate! We haven't looked at other browsers



Useful stuff

- https://github.com/w3c/webappsec-csp/issues
- https://oreoshake.github.io/csp/twitter/2014/07/25/twitters-csp-report-collector-design.html
- https://github.com/yahoo/csptester
- https://github.com/slightlyoff/CriSP

ZERO NIGHTS 2015

Thanks!

@sshekyan @jspedant