Django and Web Security Headers

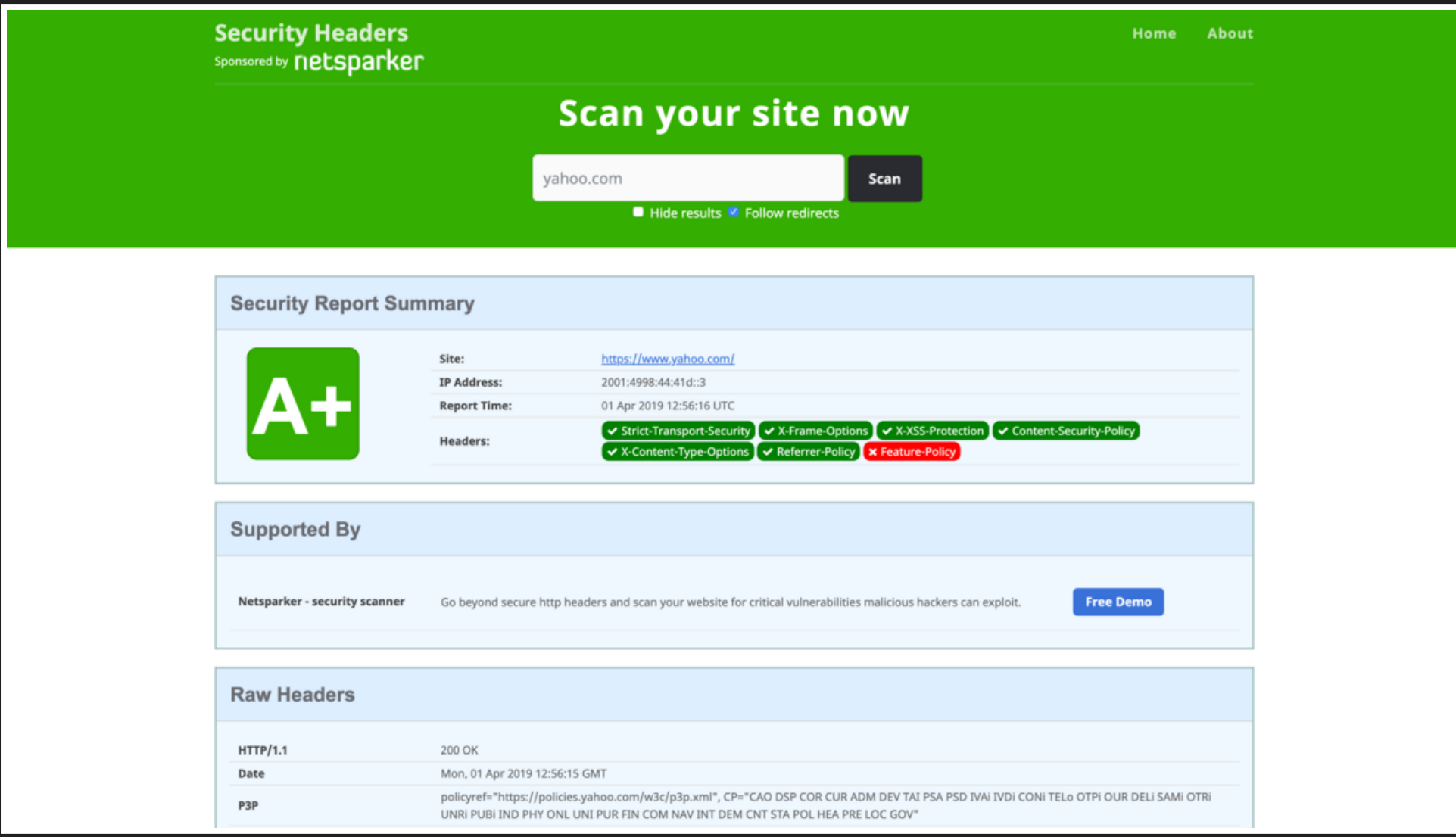
Adam Johnson

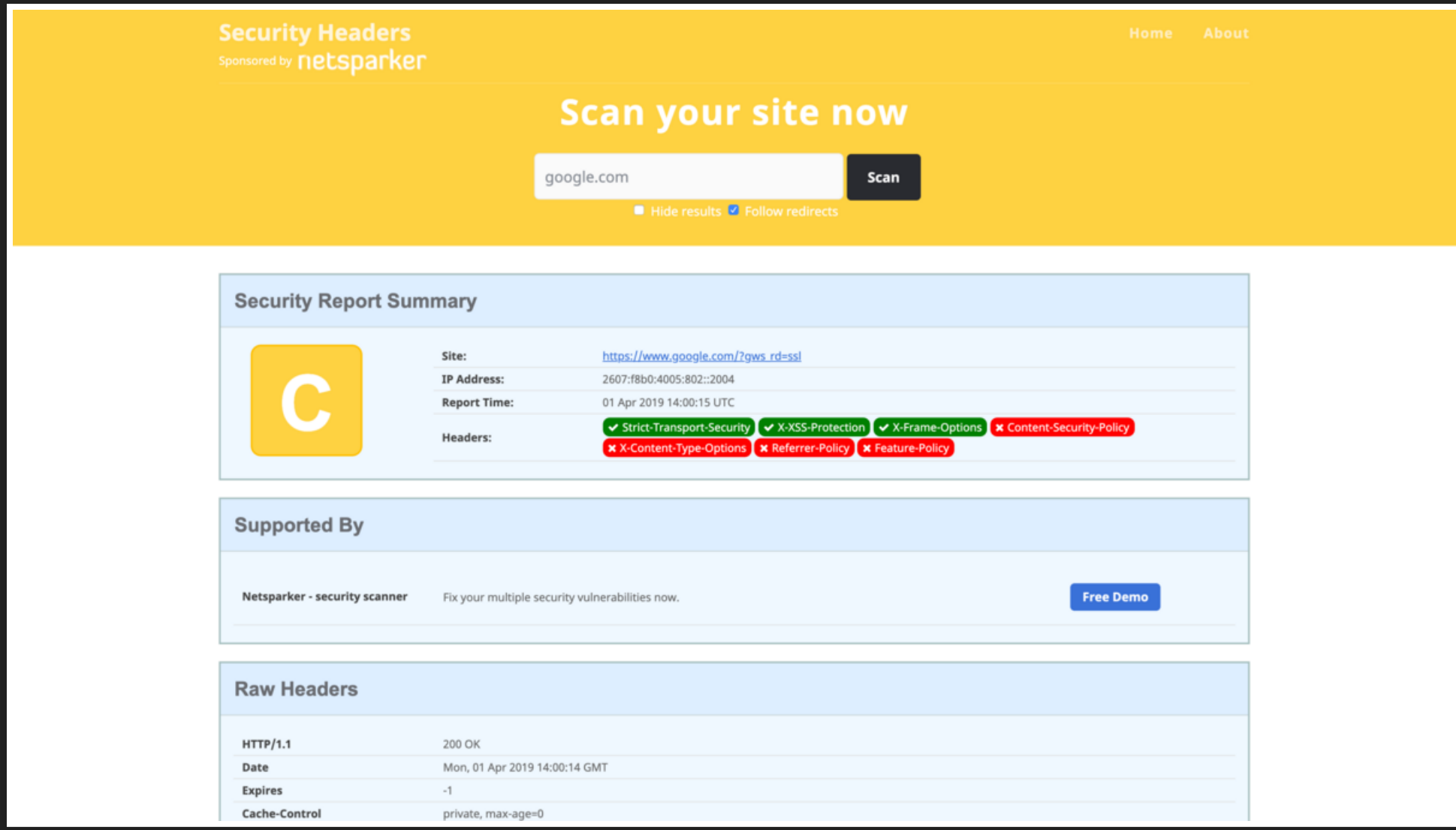
Me

- Adam Johnson
- Django Core Contributor
- @adamchainz on GitHub & Twitter
- me@adamj.eu

Web Security Headers

- Opt-in to more secure behaviour 💪
- Check them with **Securityheaders.com** 🔍


Security Headers
Sponsored by netsparker
Home
About
Scan your site now
yahoo.com
Scan
Hide results
Follow redirects
Security Report Summary
A+
Site:
https://www.yahoo.com/
IP Address:
2001:4998:44:41d::3
Report Time:
01 Apr 2019 12:56:16 UTC
Headers:
Strict-Transport-Security
X-Frame-Options
X-XSS-Protection
Content-Security-Policy
X-Content-Type-Options
Referrer-Policy
Feature-Policy
Supported By
Netsparker - security scanner
Go beyond secure http headers and scan your website for critical vulnerabilities malicious hackers can exploit.
Free Demo
Raw Headers
HTTP/1.1
200 OK
Date
Mon, 01 Apr 2019 12:56:15 GMT
P3P
policyref="https://policies.yahoo.com/w3c/p3p.xml", CP="CAO DSP COR CUR ADM DEV TAI PSA PSD IVAi IVDi CONi TELo OTPi OUR DELi SAMi OTRi UNRi PUBi IND PHY ONL UNI PUR FIN COM NAV INT DEM CNT STA POL HEA PRE LOC GOV"

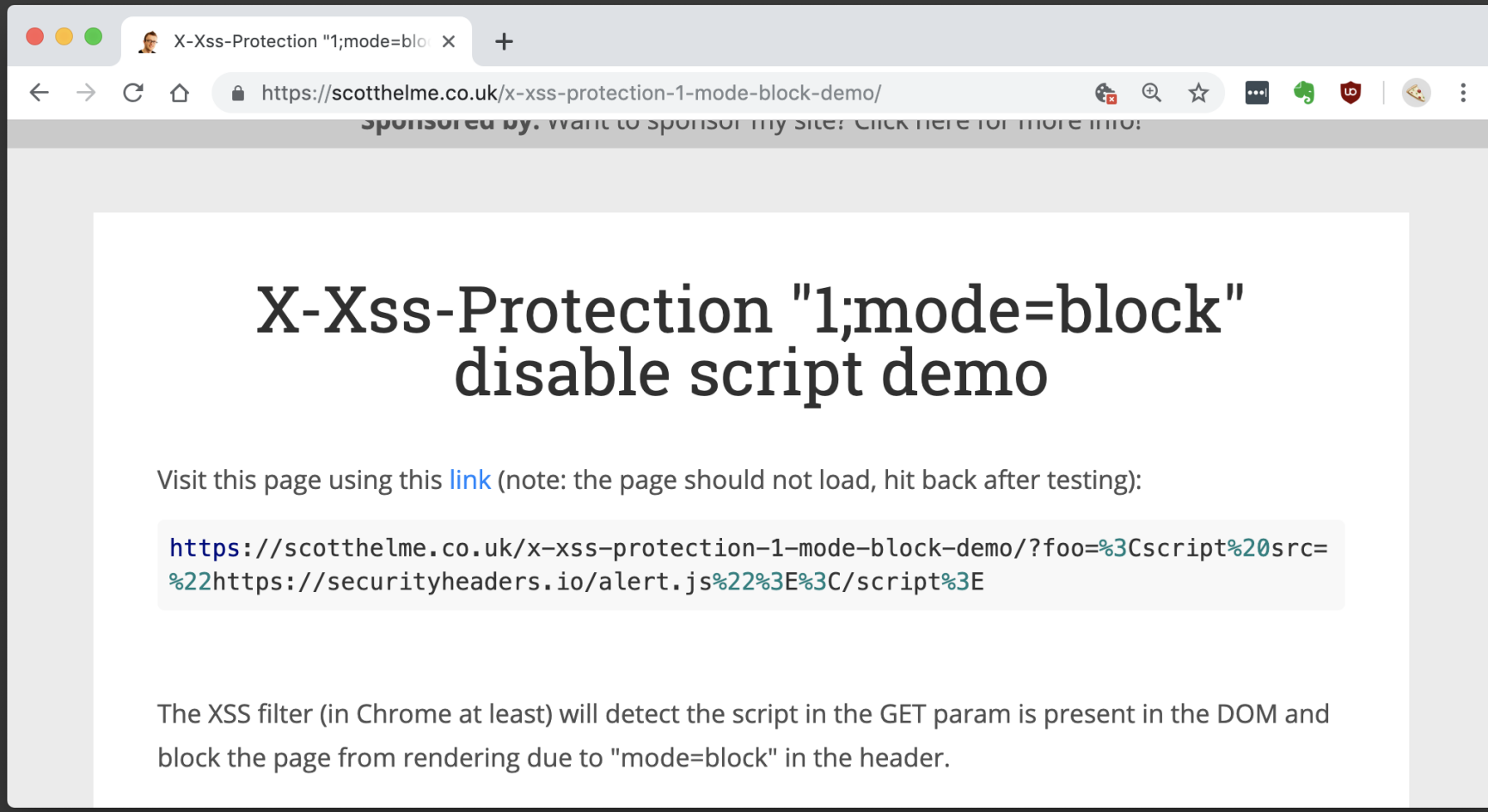

Security Headers
Sponsored by netsparker
Home
About
Scan your site now
google.com
Scan
Hide results
Follow redirects
Security Report Summary
C
Site:
https://www.google.com/?gws_rd=ssl
IP Address:
2607:f8b0:4005:802::2004
Report Time:
01 Apr 2019 14:00:15 UTC
Headers:
Strict-Transport-Security
X-XSS-Protection
X-Frame-Options
Content-Security-Policy
X-Content-Type-Options
Referrer-Policy
Feature-Policy
Supported By
Netsparker - security scanner
Fix your multiple security vulnerabilities now.
Free Demo
Raw Headers
HTTP/1.1
200 OK
Date
Mon, 01 Apr 2019 14:00:14 GMT
Expires
-1
Cache-Control
private, max-age=0

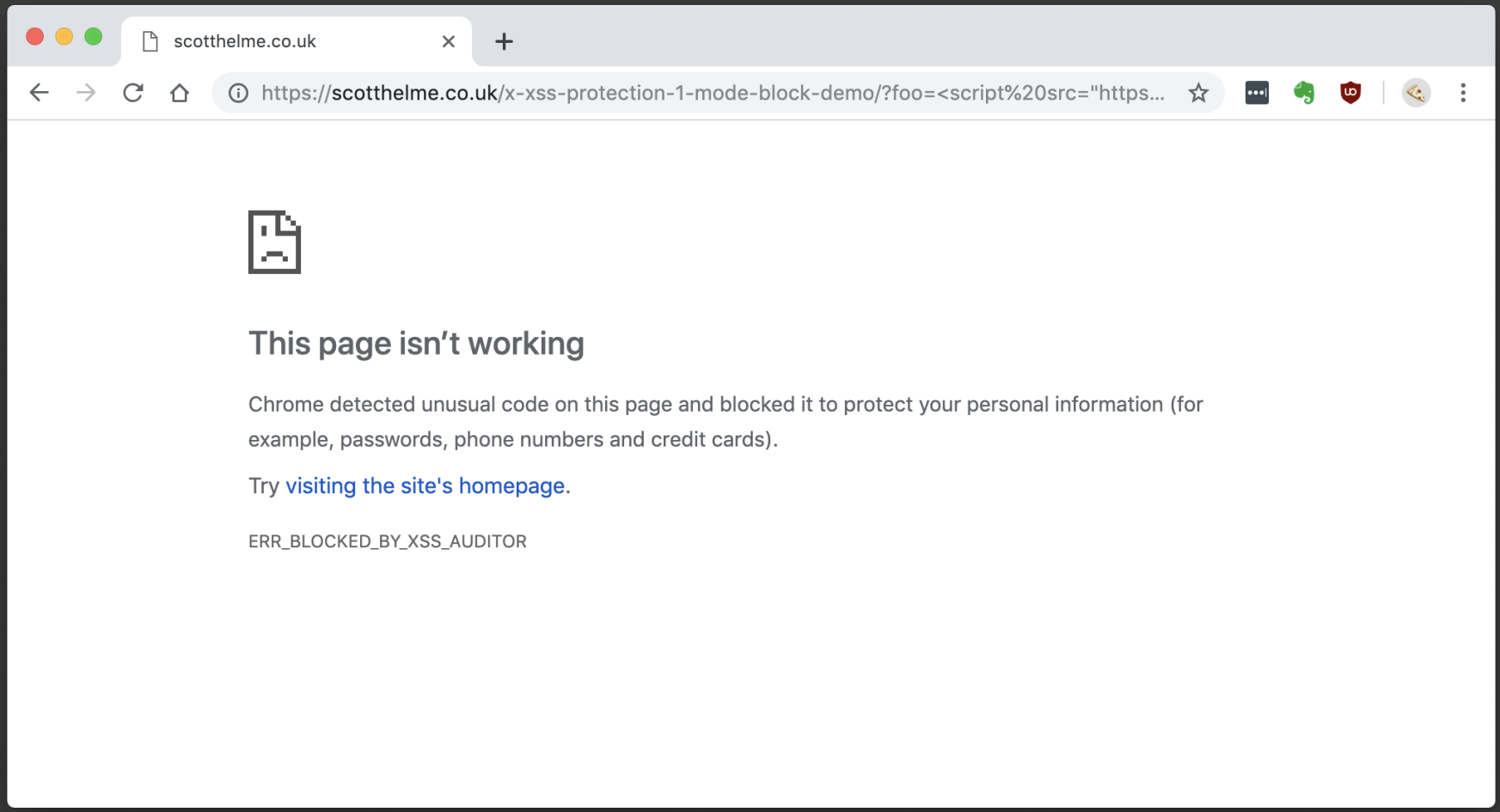
7 Headers for A+

1. X-XSS-Protection
2. Strict-Transport-Security
3. X-Content-Type-Options
4. X-Frame-Options
5. Referrer-Policy
6. Content-Security-Policy
7. Feature-Policy

1. `X-XSS-Protection`

- XSS = Cross-Site Scripting 💉
- Most browsers have XSS Auditors on by default 💂
- `mode=block` to escalate blocking a request to blocking the whole page


X-Xss-Protection "1;mode=blo
https://scotthelme.co.uk/x-xss-protection-1-mode-block-demo/
Sponsored by: Want to sponsor my site? Click here for more info!
X-Xss-Protection "1;mode=block" disable script demo
Visit this page using this link (note: the page should not load, hit back after testing):
https://scotthelme.co.uk/x-xss-protection-1-mode-block-demo/?foo=%3Cscript%20src=%22https://securityheaders.io/alert.js%22%3E%3C/script%3E
The XSS filter (in Chrome at least) will detect the script in the GET param is present in the DOM and block the page from rendering due to "mode=block" in the header.


scotthelme.co.uk
https://scotthelme.co.uk/x-xss-protection-1-mode-block-demo/?foo=<script%20src="https...
This page isn't working
Chrome detected unusual code on this page and blocked it to protect your personal information (for example, passwords, phone numbers and credit cards).
Try visiting the site's homepage.
ERR_BLOCKED_BY_XSS_AUDITOR

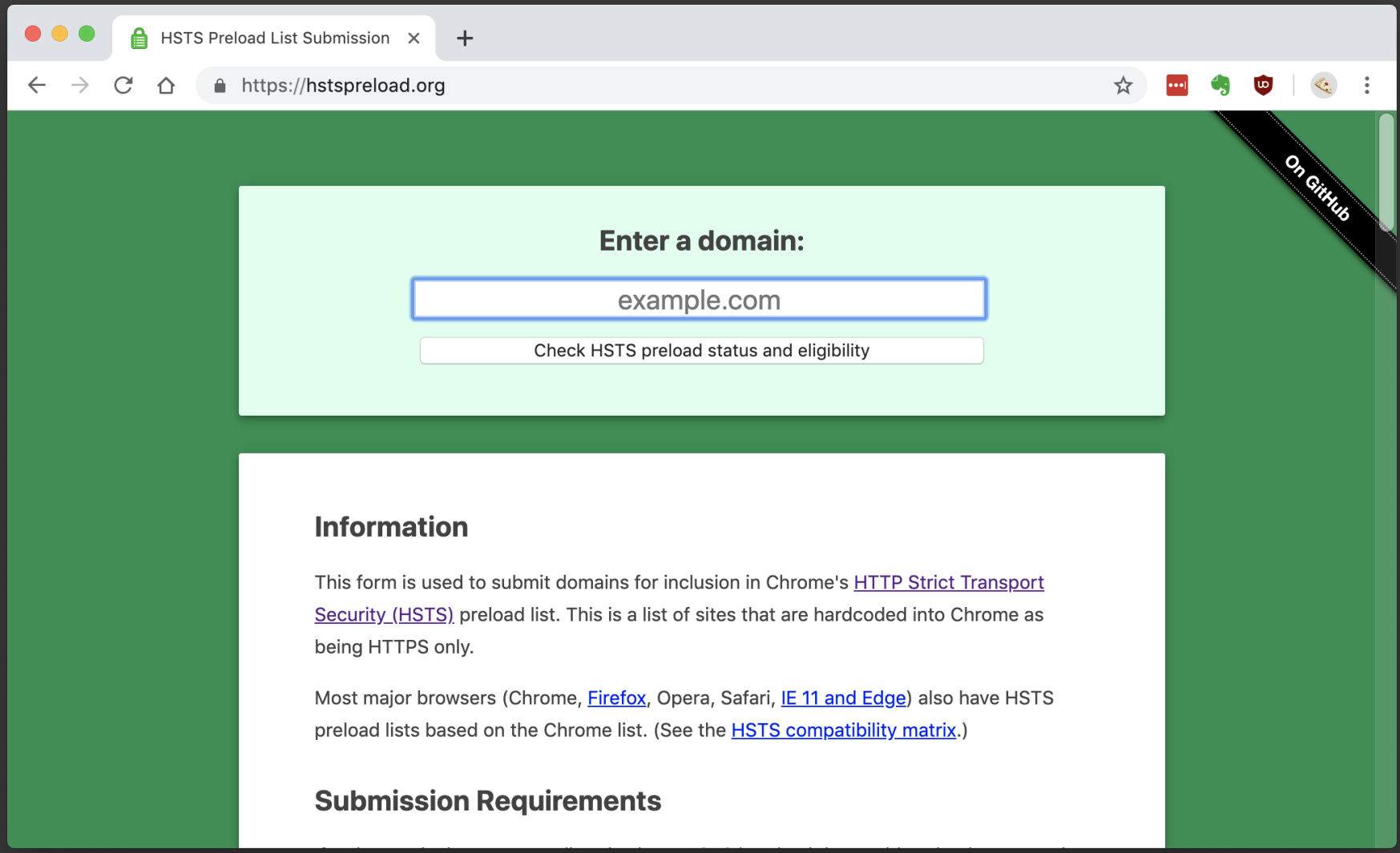
1. `X-XSS-Protection`

Activation in Django:

1. Have `SecurityMiddleware` in `MIDDLEWARE`
2. Set `SECURE_BROWSER_XSS_FILTER = True`

2. `Strict-Transport-Security`

- Your site is HTTPS 🔒
- HTTP ➡ HTTPS redirect is still insecure 🔓
- Header says "never talk to me on HTTP again" 🖐
- Preload database in browsers


HSTS Preload List Submission
https://hstspreload.org
On GitHub
Enter a domain:
example.com
Check HSTS preload status and eligibility
Information
This form is used to submit domains for inclusion in Chrome's HTTP Strict Transport Security (HSTS) preload list. This is a list of sites that are hardcoded into Chrome as being HTTPS only.
Most major browsers (Chrome, Firefox, Opera, Safari, IE 11 and Edge) also have HSTS preload lists based on the Chrome list. (See the HSTS compatibility matrix.)
Submission Requirements

2. `Strict-Transport-Security`

Activation in Django:

1. Have `SecurityMiddleware` in `MIDDLEWARE`
2. Set `SECURE_HSTS_SECONDS`

2. `Strict-Transport-Security`

> *...enabling HSTS carelessly can cause serious, irreversible problems*

- Ramp up seconds gradually
- Enable `includeSubdomains` and `preload` when you're sure

3. `X-Content-Type-Options`

- Browser "MIME Sniffing" guesses content type 👃
- Backfires e.g. user uploaded image interpreted as HTML 😓
- Header set to 'nosniff' *opts out* ☝️

MIME Sniffing Standard

https://mimesniff.spec.whatwg.org

MIME Sniffing

Living Standard — Last Updated 31 March 2019

Participate:

GitHub whatwg/mimesniff (new issue, open issues)
IRC: #whatwg on Freenode

Commits:

GitHub whatwg/mimesniff/commits
Snapshot as of this commit
@mimesniff

Tests:

web-platform-tests mimesniff/ (ongoing work)

Translations (non-normative):

日本語

File an issue about the selected text

3. `X-Content-Type-Options`

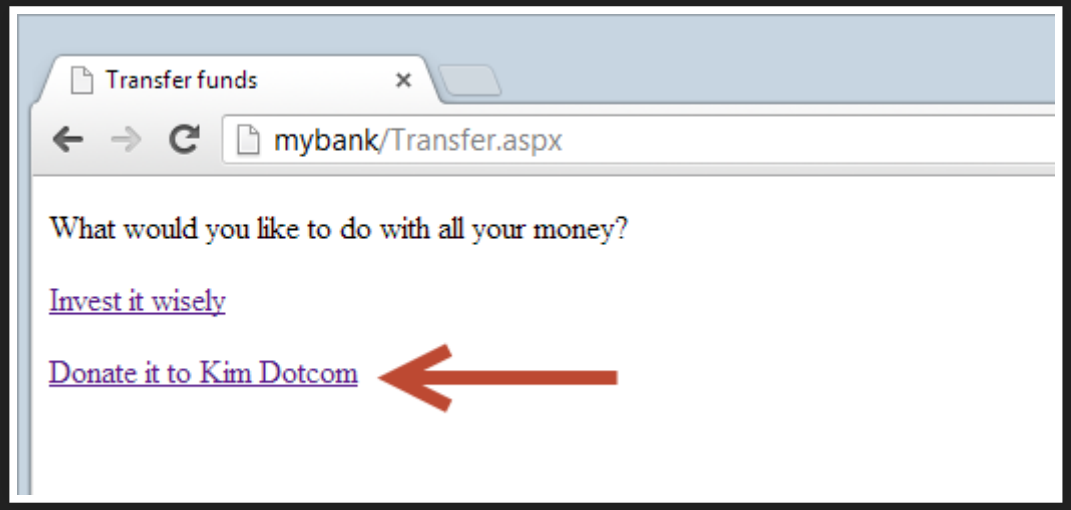
Activation in Django:

- Have `SecurityMiddleware` in `MIDDLEWARE`
- Set `SECURE_CONTENT_TYPE_NOSNIFF = True`

4. `X-Frame-Options`

- Attack: clickjacking 
- Header tells browser which other sites yours can appear on in an `<iframe>`

Troy Hunt's clickjacking demo:


Transfer funds
mybank/Transfer.aspx
What would you like to do with all your money?
Invest it wisely
Donate it to Kim Dotcom


Win an iPad Mini!!!
winipad/
Hey - we're giving away iPad minis!!! Just click the WIN button and it's yours!!!
What would you like to do with all your money?
Invest it wisely
>> WIN <<

4. X-Frame-Options

Activation in Django:

- Have `XFrameOptionsMiddleware` in `MIDDLEWARE`
- Set `X_FRAME_OPTIONS = 'DENY'`

5. `Referrer-Policy`

- Browsers pass `Referer` *(one R)* header on navigation 

- URI's can leak information! 💧
- e.g. `adamj.eu/illuminati-funding/`
- Header *(two R's)* controls who gets `Referer` 👀

5. `Referrer-Policy`

Activation in Django:

- Install James Bennett's `django-referrer-policy`
- Add its `ReferrerPolicyMiddleware` in `MIDDLEWARE`
- Set `REFERRER_POLICY` to desired value, e.g. `'same-origin'`

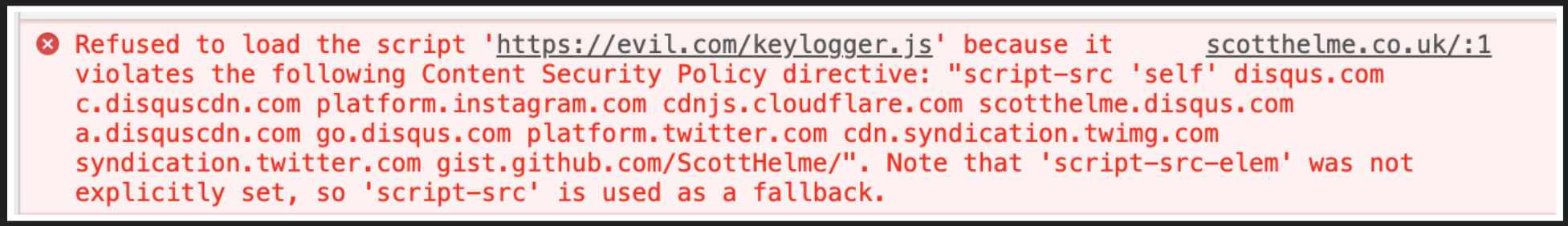
6. Content-Security-Policy

- Stop your site including content from any old other site 🦑
- Strongest way to prevent XSS, clickjacking, and others 🗝️
- Huge topic 🐋

Example CSP

```
default-src 'self';
img-src *;
script-src userscripts.example.com
```

Scott Helme's Demo:



6. `Content-Security-Policy`

Activation in Django:

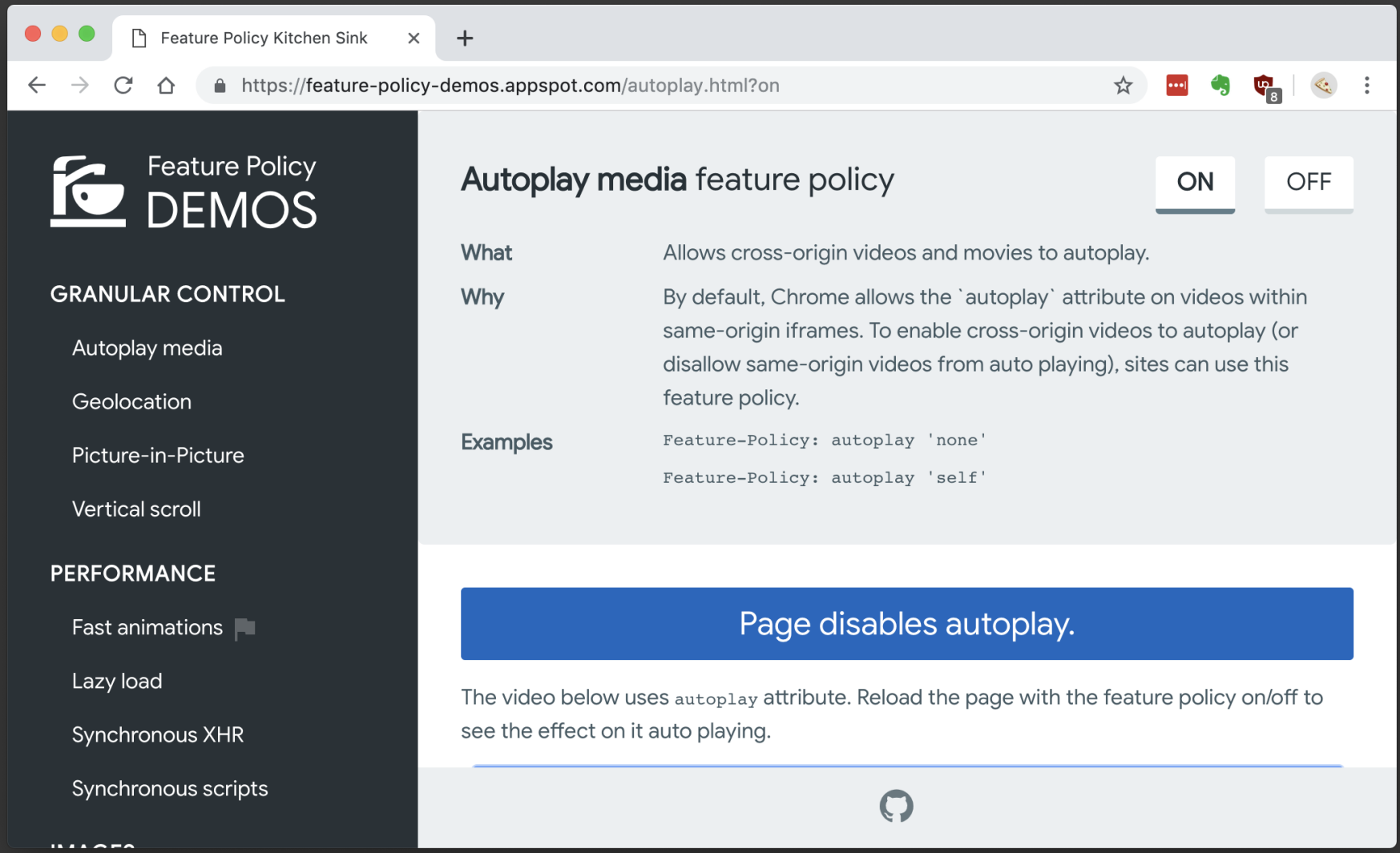
- Install Mozilla's `django-csp`
- Add its `CSPMiddleware` in `MIDDLEWARE`
- Set a bunch of settings e.g. `CSP_DEFAULT_SRC`

6. Content-Security-Policy

- Greenfield: easy 😁
- Existing site: hard 😅
- csp.withgoogle.com/docs/strict-csp.html
- Report-only mode 📋

7. `Feature-Policy`

- Bonus! Not needed for A+. 🎁
- Experimental! 🧪
- Disable browser features your site or its `iframes` shouldn't use, e.g. `autoplay`, `geolocation`, `camera` 🛑


Feature Policy Kitchen Sink
https://feature-policy-demos.appspot.com/autoplay.html?on
Feature Policy
DEMOS
GRANULAR CONTROL
Autoplay media
Geolocation
Picture-in-Picture
Vertical scroll
PERFORMANCE
Fast animations
Lazy load
Synchronous XHR
Synchronous scripts
Autoplay media feature policy
ON
OFF
What
Allows cross-origin videos and movies to autoplay.
Why
By default, Chrome allows the `autoplay` attribute on videos within same-origin iframes. To enable cross-origin videos to autoplay (or disallow same-origin videos from auto playing), sites can use this feature policy.
Examples
Feature-Policy: autoplay 'none'
Feature-Policy: autoplay 'self'
Page disables autoplay.
The video below uses autoplay attribute. Reload the page with the feature policy on/off to see the effect on it auto playing.

7. `Feature-Policy`

Activation in Django:

- Install my `django-feature-policy`
- Add its `FeaturePolicyMiddleware` in `MIDDLEWARE`
- Set `FEATURE_POLICY`

 A+ Rating! 

Thank you! 

- Adam Johnson
- @adamchainz on GitHub & Twitter
- me@adamj.eu
- github.com/adamchainz/talk-django-and-web-security-headers