'humble' (HTTP Headers Analyzer)

https://github.com/rfc-st/humble | v.2024-10-31

[0. Info]

Date : 2024/10/31 - 18:22:14

URL : https://cnn.com

File : humble_https_cnn_com_20241031_182215_en.pdf

[1. Missing HTTP Security Headers]

Clear-Site-Data

Clears browsing data (cookies, storage, cache) associated with the requesting website.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Clear-Site-Data

Cross-Origin-Embedder-Policy

Prevents documents and workers from loading non-same-origin requests unless allowed.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Embedder-Policy

Cross-Origin-Opener-Policy

Prevent other websites from gaining arbitrary window references to a page.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cross-Origin-Opener-Policy

Cross-Origin-Resource-Policy

Protect servers against certain cross-origin or cross-site embedding of the returned source.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cross-Origin_Resource_Policy_(CORP)

(*) NEL

Enables web applications to declare a reporting policy to report errors.

Ref: https://scotthelme.co.uk/network-error-logging-deep-dive/

Permissions-Policy

Previously called "Feature-Policy", allow and deny the use of browser features.

Ref: https://scotthelme.co.uk/goodbye-feature-policy-and-hello-permissions-policy/

Referrer-Policy

Controls how much referrer information should be included with requests.

Ref: https://scotthelme.co.uk/a-new-security-header-referrer-policy/

Strict-Transport-Security

Tell browsers that it should only be accessed using HTTPS, instead of using HTTP.

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

X-Permitted-Cross-Domain-Policies
Limit which data external resources (e.g. Adobe Flash/PDF documents), can access on the domain.
Ref: https://owasp.org/www-project-secure-headers/#div-headers

[2. Fingerprint HTTP Response Headers]

These headers can leak information about software, versions, hostnames or IP addresses:

Via [Generic Proxy server]
Value: '1.1 varnish, 1.1 varnish'

X-Served-By [Generic HTTP Server/Content Delivery Network]
Value: 'cache-iad-kcgs7200105-IAD, cache-iad-kcgs7200105-IAD, cache-mad22049-MAD'

[3. Deprecated HTTP Response Headers/Protocols and Insecure Values]

The following headers/protocols are deprecated or their values may be considered unsafe:

Access-Control-Allow-Origin (Unsafe Values)
Review the values '*' or 'null' regarding your Cross-origin resource sharing requirements.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin

Cache-Control (Recommended Values)
Enable 'no-cache', 'no-store', and 'must-revalidate' if there are sensitive data.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control

Content-Security-Policy (Insecure Schemes)
Do not allow insecure, unencrypted schemes: 'http:'
Ref: https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/
Ref: https://http.dev/wss

Content-Security-Policy (Too Permissive Sources)
Limit these permissive origins: 'data:', '*', 'blob:'
Ref: https://content-security-policy.com/

Content-Security-Policy (Unsafe Values)
'unsafe-inline' and 'unsafe-eval' negate most of the security benefits provided by this header.

Ref: https://csper.io/blog/no-more-unsafe-inline
Ref: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval

Set-Cookie (Insecure Attributes)
Enable 'Secure' and 'HttpOnly': to send it via HTTPS and not be accessed by client APIs.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie

Vary (Potentially Unsafe Header)
The values of this header may expose others, facilitating attacks if user input is accepted.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Vary
Ref: https://www.yeswehack.com/fr/learn-bug-bounty/http-header-exploitation

X-XSS-Protection (Deprecated Header)
This header is deprecated in the three major web browsers.
Instead, use the "Content-Security-Policy" header restrictively.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

X-XSS-Protection (Unsafe Value)
In some cases values other than '0' can create XSS vulnerabilities.
Instead, use the "Content-Security-Policy" header restrictively.
Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

[4. Empty HTTP Response Headers Values]

Empty HTTP headers (and are therefore considered disabled):

Nothing to report, all seems OK!

[5. Browser Compatibility for Enabled HTTP Security Headers]

Cache-Control: https://caniuse.com/?search=Cache-Control
Content-Security-Policy: https://caniuse.com/?search=contentsecuritypolicy2
Content-Type: https://caniuse.com/?search=Content-Type
Set-Cookie: https://caniuse.com/?search=Set-Cookie
Vary: https://caniuse.com/?search=Vary
X-Content-Type-Options: https://caniuse.com/?search=X-Content-Type-Options
X-XSS-Protection: https://caniuse.com/?search=X-XSS-Protection

[6. Analysis Results]

Done in 0.45 seconds! (changes with respect to the last analysis in parentheses)

Missing headers:	9 (First Analysis)
Fingerprint headers:	2 (First Analysis)
Deprecated/Insecure headers:	9 (First Analysis)
Empty headers:	0 (First Analysis)
Findings to review:	20 (First Analysis)
Analysis Grade:	D (Review 'Deprecated/Insecure headers')
'(*)' meaning:	Experimental HTTP response header