

Humble HTTP headers analyzer
(<https://github.com/rfc-st/humble>)

[0. Info]

Date : 2022/08/13 - 20:18:41
Domain : <https://facebook.com>

[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-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_\(CORP\)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Cross-Origin_Resource_Policy_(CORP))

Expect-CT

Prevents the use of misissued certificates from going unnoticed.
Ref: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Expect-CT>

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/>

[2. Fingerprint HTTP Response Headers]

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

Nothing to report, all seems OK!

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

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

Humble HTTP headers analyzer
(<https://github.com/rfc-st/humble>)

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

Content-Security-Policy (Deprecated Directives)

Avoid using deprecated directives: block-all-mixed-content, report-uri.

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

Strict-Transport-Security (Recommended Values)

Add 'includeSubDomains' and set 'max-age' to at least 31536000 (one year).

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

Ref: <https://https.cio.gov/hsts/>

[4. Empty HTTP Response Headers Values]

The following headers have no value (could be equivalent to as if they were not enabled):

Nothing to report, all seems OK!

[5. Browser Compatibility for Enabled HTTP Security Headers]

Cache-Control: <https://caniuse.com/?search=Cache-Control>

Content-Type: <https://caniuse.com/?search=Content-Type>

Content-Security-Policy: https://caniuse.com/?search=mdn-http_headers_csp_content-security-policy

Cross-Origin-Opener-Policy: <https://caniuse.com/?search=Cross-Origin-Opener-Policy>

Pragma: <https://caniuse.com/?search=Pragma>

Strict-Transport-Security: <https://caniuse.com/?search=Strict-Transport-Security>

X-Content-Type-Options: <https://caniuse.com/?search=X-Content-Type-Options>

X-Frame-Options: <https://caniuse.com/?search=X-Frame-Options>

∴

Analysis done in 0.76 seconds!.

Advice: check the deprecated headers/insecure and then the missing headers.

∴