'humble' (HTTP Headers Analyzer)

https://github.com/rfc-st/humble | v.2024-10-21

[0. Info]

Date : 2024/10/21 - 22:27:18
URL : https://facebook.com
File : humble_https_facebook_com_20241021_222719_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-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/

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:

X-FB-Debug [facebook.com Platform]

Value: 'cZ0/VRwlxA41PF0XAYjLM6lJ6edoOZYg9mMkBaTupiZjCz1WUKnX/gW33psRDhDnPAV97egrB2465YEUijCf4w=='

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

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

Content-Security-Policy (Deprecated Directives)

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

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

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

(*) Origin-Agent-Cluster (No Valid Directives)

The only valid value is '?1'.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Origin-Agent-Cluster

Pragma (Deprecated Header)

This header is deprecated.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Pragma

Report-To (Deprecated Header)

This header is deprecated. Use instead "Reporting-Endpoints".

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Report-To

Strict-Transport-Security (Recommended Values)

Add 'includeSubDomains' and 'max-age' (with 31536000 -one year- as minimum).

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

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

Strict-Transport-Security (Required Values)

'preload' requires 'includeSubDomains' and 'max-age' (with 31536000 -one year- as minimum).

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

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

[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

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

Origin-Agent-Cluster: https://caniuse.com/?search=Origin-Agent-Cluster

Reporting-Endpoints: https://caniuse.com/?search=Reporting-Endpoints

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

Vary: https://caniuse.com/?search=Vary

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

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

```
 X-XSS-Protection: https://caniuse.com/?search=X-XSS-Protection



[6. Analysis Results]

 Done in 0.65 seconds! (changes with respect to the last analysis in parentheses)

 Missing headers:              7 (First Analysis)
 Fingerprint headers:          1 (First Analysis)
 Deprecated/Insecure headers:  11 (First Analysis)
 Empty headers:                0 (First Analysis)

 Findings to review:           19 (First Analysis)

 Analysis Grade:               D (Review 'Deprecated/Insecure headers')

 '(*)' meaning:                Experimental HTTP response header
```