'humble' (HTTP Headers Analyzer)

https://github.com/rfc-st/humble | v.2024-11-22

[0. Info]

Date : 2024/11/22 - 21:30:20

URL : https://samsung.com

File : humble_https_samsung_com_20241122_213022_en.pdf

[1. Enabled HTTP Security Headers]

Content-Type: text/html; charset=UTF-8

Set-Cookie: device_type=pc; path=/; domain=.samsung.com

Strict-Transport-Security: max-age=31536000

X-Frame-Options: SAMEORIGIN

[2. Missing HTTP Security Headers]

Cache-Control

Directives for caching in both requests and responses.

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

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)

Content-Security-Policy

Detect and mitigate Cross Site Scripting (XSS) and data injection attacks, among others.

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

(*) 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-Content-Type-Options

Indicate that MIME types in the "Content-Type" headers should be followed.

Ref: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

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

[3. Fingerprint HTTP Response Headers]

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

X-Akamai-Transformed [Akamai Edge]

Value: '9 - 0 pmb=mRUM,3'

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

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

Content-Type (Incorrect Value: Response body)

The only allowed value is 'text/html; charset=utf-8'

Ref: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta

Etag (Potentially Unsafe Header)

Although unlikely to be exploited, this header should not include inode information.

Ref: https://www.pentestpartners.com/security-blog/vulnerabilities-that-arent-etag-headers/

Server-Timing (Potentially Unsafe Header)

This header should not expose sensitive application or infrastructure information.

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

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

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/

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

[5. Empty HTTP Response Headers Values]

Empty HTTP headers (and are therefore considered disabled):

Nothing to report, all seems OK!

[6. Browser Compatibility for Enabled HTTP Security Headers]

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

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

Server-Timing: https://caniuse.com/?search=Server-Timing

Set-Cookie: https://caniuse.com/?search=Set-Cookie

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

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

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

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

[7. Analysis Results]

```
 Done in 1.96 seconds! (changes with respect to the last analysis in parentheses)

 Missing headers:              11 (First Analysis)
 Fingerprint headers:          1 (First Analysis)
 Deprecated/Insecure headers:  7 (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
 '(*)' ref:                    https://mdn.io/Experimental_deprecated_obsolete
```