@yoavweiss - Dec 2019

Public document (converted from internal document)

In a discussion with @davidben, @mikewest, @pauljensen and @aarontag, it was concluded that we want to prefix the Client Hints we ship for the following reasons:

CORS safety

We want the various Client Hints to be CORS safe request headers, because **we need them to avoid preflights**. That means we need to either add them below the CORS checks, or add them to the CORS safe lists.
The latter is something that was rightfully referred to as “poking holes in the same-origin policy”. It would be better to add an exemption once for a class of headers, and be sure that we can consider them safe without a preflight. A well defined prefix will help us there.

Server deployment safety

Another problem with request headers is that they can trigger unintended consequences on unsuspecting servers. Request headers with names that are too generic are likely to be confused by at least some server-side stacks with internal variables and mess up internal server state.
The most prominent example of that was the deployment of the “HTTPS” request header, which caused server-side breakage and was renamed to be “Upgrade-Insecure-Request”.
Adding a prefix can help reduce those odds.

Namespacing

At IETF 105 some folks wanted to be able to tell apart Client Hints from other request headers. Having a clear prefix for all Client Hints would help on that front.
At the same time, there were reservations about creating such a namespace without defining its semantics.

Ensuring that the headers cannot be modified from Javascript

While servers will still need to protect themselves from request headers that contain weird values, it would be a significant improvement if they could trust that credentialed requests from their users cannot arrive with arbitrary values in their Client Hint headers.
Currently for the User-Agent string, its header can be modified by JS and that turned out to be a mistake. It would be good to avoid repeating that same mistake with UA client hints.
Other hints may also have unintended consequences if arbitrarily modified by an attacker (e.g. image servers can perform costly manipulations, DoSing themselves)

Therefore, it seems safer to rule that out, and make it so that hints are not modifiable by script, either by adding a `Sec-` prefix, or by another prefix that has similar semantics.
The counter argument for that, “what if we’d want SW to change CH headers?” seems better addressed in different ways. E.g. the use case where Twitter has their own save-data preference and may want to add that header can be better addressed by a “tell the browser the user wants to save data” API.