The use of the web for services that were previously limited to mobile or desktop applications has increased. Because of this, web applications have become more and more popular, and web browsers have developed from basic document parsers to functional powerhouses, enhancing user experience and enabling developers to create interactive web pages. Given that a browser has a wide range of access to device functionalities, this raises concerns about user privacy and security. Additionally, a developer could wish to have complete control over their own or third-party content accessing these browser features. Because of this, the W3C established the Feature Policy response headers in April 2019 and a later version of it Permissions Policy in July 2020, to provide developers the choice of enabling or disabling them for documents of the same origin as well as those of different origins [1].

Permissions policy can be used in two ways: as an HTTP response header or inline iframe attribute providing delegated access from the parent site to the iframe. The developer can choose between 27 standardized features, 4 proposed features and 11 experimental features, and some specialized features such as interest_cohort used as an opt-out of Google's FloC web-tracking campaign. In order to disallow unwanted features, the developer has to go out of their way and disallow each existing feature one by one. Assume that Geolocation was disabled in such a way. Then when introducing a new iframe that needs this feature, either the response header or the iframe should be edited to allow this feature using the keyword allow="Geolocation" in the iframe. Keep in mind that in the case that Geolocation is only allowed to same-origin or specific third-party domains access to that feature will not be granted. This process of configuring permissions policy can easily become tedious for website owners that frequently add or remove iframes and interactive elements.

Moreover, W3C states that "Permissions Policy is intended to be used in combination with a sandbox mechanism". However, we are not sure if this is implemented along with the potentially untrusted content in the wild. Furthermore, although there are tools to auto-generate HTTP response headers [2], there are no tools yet for generating allow lists for iframes and third-party content or editing those that are in conflict with the Permission Policy HTTP header. This may discourage developers from using them altogether, as seen on Scott Helme's monthly crawls of the top 1 million websites [3] showing only 4.9 percent use this header and 0.59 continue to use older Feature Policy header as of December 2022. This may result in inconsistent definitions between the response header and the HTML document leading to possible privacy issues, namely unwanted usage of browser features, especially in the presence of an attacker who can embed their own iframes in the parent website.

In similar research Kaleli et al. studied privacy issues of the incorrect use of the feature policy [4] which focused on setting the Feature Policy header rather than each individual iframe arguing it would reduce human error in missing one single iframe. However, this argument only holds when allowing a certain feature to self (cross-origin only) or a specified third-party such as example.com. If one wants to disable a feature for all same and cross-origin in the Permissions Policy response header, an iframe could still make an exception for that feature using the allow keyword to overrule the response header settings. In my research, I plan to take a different route to address the same problem at hand. In a number of steps:

1. Crawl and analyze main and some subpages for iframe feature allowances in contradiction with the Permissions Policy header
2. Perform a case study on Dutch government websites to see if said condition occurs on their iframes.
3. Provide a more accurate and systematic approach for configuring iframe that can help developers avoid said misconfigurations
4. Compare Chrome and Firefox's implementation of the Permissions Policy against the standardization provided by W3C to find if there is a deviation.

References

[1] “Permissions Policy.” Www.w3.org, 7 Dec. 2022, www.w3.org/TR/permissions-policy/. Accessed 20 Dec. 2022.

[2] “Permissions Policy HTTP Header Generator.” www.permissionspolicy.com, www.permissionspolicy.com/. Accessed 20 Dec. 2022.

[3] Helme, Scott. “Top 1 Million Sites Security Analysis.” Crawler.ninja, crawler.ninja/. Accessed 20 Dec. 2022.

[4] Kaleli, B., Egele, M., & Stringhini, G. Studying the Privacy Issues of the Incorrect Use of the Feature Policy.