Security Questions to Consider

1. Does this specification deal with high-value data?
    a. *Explanation:* Data which isn’t personal can still be quite valuable. Sign-in credentials (like username/password pairs, or OAuth refresh tokens) can be extremely powerful in the wrong hands, as can financial instruments like credit card data. Making this data available to JavaScript, for instance, could expose it to XSS attacks and active network attackers who could inject code to read and exfiltrate the data. For instance:
    b. *Example:* Credential Management allows sites to request a user’s credentials from a user agent’s password manager in order to sign the user in quickly and easily. This opens the door for abuse, as a single XSS could expose user data trivially to JavaScript. They mitigate the risk by only offering the username and password as an opaque FormData object which cannot be directly read by JavaScript, and strongly suggest that authors use Content Security Policy (CSP) with reasonable connect-src and form-action values to further mitigate the risk of exfiltration.
2. Does this specification introduce new state for an origin that persists across browsing sessions?
    a. Any new state that persists across browsing sessions can be used to track users.
    b. Example: Cookies, ETag, Last Modified, Local Storage, Indexed DB, etc. all allow an origin to store information about a user, and retrieve it later, directly or indirectly. User agents mitigate the risk that these kinds of storage mechanisms will form a persistent identifier by offering users the ability to wipe out the data contained in these types of storage.
    c.
3. Does this specification create additional information that may expose a user to fingerprinting?
    a. While fingerprinting is difficult (if not impossible) to completely eliminate, specifications should attempt to minimize the creation of new sources of fingerprinting data when possible
    b. Example: The GL_RENDERER string exposed by some WebGL implementations improves performance in some kinds of applications, but does so at the cost of adding persistent state to a user’s fingerprint. These kinds of device-level details should be carefully weighed to ensure that the costs are outweighed by the benefits.
4. Does this specification expose any other data to an origin that it doesn’t currently have access to?
    a. Explanation: Cross origin access can allow attackers to inject malicious code
    b. Example: Cross Site Scripting (XSS) – the injection of malicious scripts into an otherwise benign webpage - is a widespread problem on the modern web.
5. Does this specification enable new script execution/loading mechanisms?
    a. Explanation: While scripting can be useful, when enabled in formats or standards where it is unexpected, attackers can leverage it to perform attacks.
    b. Example: When designing a new image format, care should be taken to ensure said format cannot run arbitrary commands on the user’s client.

6. Does this specification allow an origin access to aspects of a user's local computing environment which might aid in fingerprinting?
    a. Explanation: Standards makers should be cognizant of the risk of fingerprinting. Any network facing information which can change from user to user and is relatively stable can be used to uniquely identify users.
    b. Example: For example, a standard that leaks any of the following could aid fingerprinting: screen sizes, installed fonts, installed plugins, bluetooth or network interface identifiers
7. Does this specification allow an origin access to other devices?
    a. Explanation: Specifically, it's interesting whether or not this specification allows access to devices on a user's local network that would be otherwise inaccessible to a web origin. In particular, connection via Bluetooth and USB should be carefully evaluated to avoid exposing devices to the web that aren't created with the web in mind; doing so has security implications, as these devices may not be hardened against malicious input as well as they should be.
    b. Example: Does this standard allow network access to a user's webcam?
8. Does this specification expose temporary identifiers to the web?
    a. Explanation: "Temporary" identifiers, if they persist for long enough, can be used to fingerprint users.
    b. Example: TLS features like Channel ID, session identifiers/tickets, etc)
9. Does this specification distinguish between behavior in first-party and third-party contexts?
    a. Actions taken in the first party context can have larger implications when applied in the 3rd party context
    b. For example, tracking users visiting one's website is much less invasive than tracking users who visit any website that has widgets allowing that web page to be shared on your social media platform.
10. Does this specification persist data to a user's local device?
    a. Explanation: If so, can the user easily remove this data, and does removing said information impact usability of the standard, or the web in general?
    b. How should user agent's "Clear browsing data" functionality work with this data? Will it remove the data? Will removing the data harm usability? (For example, requiring a user to clear all browser data to revoke permission for a specific site to activate the user's webcam)
11. Does this specification have a "Security Considerations" section?
    a. Interesting features added to the web platform generally have security and/or privacy impacts. Documenting the various concerns and potential abuses in "Security Considerations" sections of a document is a good way to help implementers and web developers understand the risks that a feature presents, and to ensure that adequate mitigations are in place.If it seems like a feature does not have security impacts, then say so inline in the spec section for that feature: "There are no known security or privacy impacts of this feature."Saying so explicitly in the specification serves several purposes:

i. Shows that a spec author/editor has explicitly considered security when designing a feature.

ii. Provides some sense of confidence that there are no such impacts.

iii. Challenges security and minded individuals to think of and find such instances (as well as the mere potential for such impacts.)

iv. Demonstrates the spec author/editor’s receptivity to feedback about such impacts.

12. Does this specification allow downgrading default security characteristics? If so, is this done in a safe manner?
    a. Explanation: Many protocols such as TLS allow two users to agree upon which cipher they will use to communicate. Care should be taken that a malicious attacker cannot force a user to select an insecure protocol
    b. Example: the FREAK attack allowed malicious actors to select insecure encryption suite.

Privacy Questions to Consider

1. Does this specification have a "Privacy Considerations" section?
   a. Interesting features added to the web platform generally have privacy impacts. Documenting the various concerns and potential abuses in "Privacy Considerations" sections of a document is a good way to help implementers and web developers understand the risks that a feature presents, and to ensure that adequate mitigations are in place. If it seems like a feature does not have privacy impacts, then say so inline in the spec section for that feature: “There are no known privacy impacts of this feature.”
   b. Saying so explicitly in the specification serves several purposes:
      1. Shows that a spec author/editor has explicitly considered privacy when designing a feature.
      2. Provides some sense of confidence that there are no such impacts.
      3. Challenges security and minded individuals to think of and find such instances (as well as the mere potential for such impacts.)
      4. Demonstrates the spec author/editor’s receptivity to feedback about such impacts.

2. Does this specification deal with personally derived data ?
   a. Explanation: Personal data includes a large swath of data which could be used on its own, or in combination with other information, to identify a single person. The exact definition of what’s considered “personal information” varies, but could certainly include things like a home address, an email address, birthdates, usernames, fingerprints, video recordings, audio recordings, geographic location or any other information derived from a person.
   b. If the specification under consideration exposes private information to the web, it is important to consider ways to mitigate the obvious impacts. For instance:
      i. A feature which uses biometric data (fingerprints or retina scans) could refuse to expose the raw data to the web, instead using the raw data only to unlock some origin-specific and ephemeral secret and transmitting that secret instead.
      ii. User mediation could be required, in order to ensure that no data is exposed without a user’s explicit choice (and hopefully understanding).
3. Does this specification allow an origin access to a user’s location, and if so is that information minimized?
   a. Explanation: A user’s location is highly-desirable information for a variety of use cases. It is also, understandably, information which many users are reluctant to share, as it can be both highly identifying, and potentially creepy. New features which make use of geolocation information, or which expose it to the web in new ways should carefully consider the ways in which the risks of unfettered access to a user’s location could be mitigated.
   b. Example: geolocation information can serve many use cases at a much less granular precision than the user agent can offer. For instance, a restaurant recommendation

can be generated by asking for a user’s city-level location rather than a position accurate to the centimeter.

4. How should this specification work in the context of a user agent’s "incognito" mode?
    a. Explanation: Ideally, the feature would work in such a way that the website would not be able to determine that the user was in "incognito". Less ideally, the feature wouldn’t work, but the website still wouldn’t be able to distinguish "incognito" from simply being denied permission to use the feature (for instance). Unideally, the feature wouldn’t exist at all in "incognito", which means that the user wouldn’t be exposing data, but the website can probably tell that the user is in that state.
    b. Example: TBD

5. Is it possible to spoof/fake the data being generated for privacy purposes?
    a. Explanation: Users may have a legitimate need to falsify the data eminating from their machine
    b. For example, a user who is located in an oppressive regime may not wish to provide their exact geographic location, instead choosing to appear to be posting from a nearby country with less draconian laws. (Eg: Chinese user might want to say they’re in Hong Kong)
6. Does the standard utilize data that is personally-derived, i.e. derived from the interaction of a single person, or their device or address?
    a. Explanation: If so, even if anonymized, could it be re-correlated? If the data could be re-correlated, does the data record contain elements that would explicitly enable such re-correlation such as unique identifiers?
    b. Example: Social Security numbers, employee ID numbers, etc
7. If the data could be re-correlated, does the data record contain elements that would enable recorrelation when combined with other datasets through the property of intersection?
    a. Example: One record contains name, DOB, and city of birth, a second contains DOB, city of birth, and medical illness treated
8. Is the user likely to know if information is being collected?
    a. Explanation: Do I get feedback on the patterns that the information could reveal (at any instant, over time) so I can adjust behaviors? Information flows should not be invisible - users should be able to see what information is being collected and adjust behaviors accordingly.
    b. For example, does a camera icon appear on a site while the webcam is being utilized? Does a noise occur when a picture is taken? Does an LED light up when the camera is on?
9. Can the user easily, preferably through an element of the GUI, revoke consent granted to a particular feature?
    a. Consent should not be a one time affair, but an ongoing process.
    b. For example, if a user must clear all cookies and cache to turn off consent granted to their webcam, this is a poor consent model.