requestStorageAccessFor

Chris Fredrickson
Aaron Selya

https://pad.w3.org/p/tpac-rsafor

The problem rSAFor is solving

- Similarities to the use case for Storage Access API (SAA)
  - Embedded sites have no ability to control what site they are embedded in
  - To enable user functionality that uses shared cross-site information, the embed must be granted access to unpartitioned cookies.

- Problem that rSAFor specifically addresses
  - SAA only works if you have at least one visible iframe that can execute JS
    - Not all subresource and pages that need cookies can execute JS

Current State of the Art

Browser vendors are working on this problem:

- Mozilla proposed Top Level Storage Access API
- Google Chrome proposed requestStorageAccessFor
  - Currently limited to sites in Related Website Sets (RWS)
    - Limited to 5 sites
    - Requires the developer to adopt and maintain a related website set

Why limit rSAFor to within RWS in Chrome?

- A prompt could be interpreted to mean that the embedder endorses the embedded site.
  - => Potential for a reputation attack
- Prompt spam

Open Question: To enable rSAFor w/o RWS, could we show a prompt?

What a solution should address

- Websites should be able to allow showing a prompt on specific other sites
  - Without browser involvement
  - No reliance on an outside dependency (such as RWS)
- A low barrier for adoption
- Cross-browser compatible

Solution 1: Well-known file

Sites can publish a well-known file that contains a list of the sites that they would allow for a prompt to be displayed when requestStorageAccessFor is called on them

- Pros
  - This allows sites to control where prompts are shown
  - Owners of other sites will know ahead of time if their requests will result in prompting or be automatically rejected
  - Lack of a file present, will default to no permission ensuring that existing behavior is preserved
- Cons
  - It’s public which might not be ideal for all sites
  - Would require the caller to ingest and process the entire list which could be expensive
  - Static, no capacity for dynamic decisions

Solution 2: API endpoint

Sites can create an endpoint that responds if requestStorageAccessFor should be automatically rejected

- Pros
  - This allows sites to control where prompts are shown
  - Callers won't have to ingest the whole list
  - Lack of an endpoint will default to no permission ensuring that existing behavior is preserved
  - Sites can make dynamic decisions
- Cons
  - Other sites won't know ahead of time if their requests might be granted
  - It's not private
  - Higher adoption cost than a plain text file

Other solutions considered

Lightweight FedCM

- It can provide the same cross-site cookie access as rSAFor
    - It offers a different set of tradeoffs and requirements which may not be preferable for a given site

Storage Access Headers

- Still requires at least one iframe w/ JS execution, at some point