wthree.c

A final* update on `:visited` link partitioning

What is this about?

Fixing the 25-year-old leak of browsing history caused by the global nature of `:visited` link styling:

- https://github.com/explainers-by-googlers/Partitioning-visited-links-history

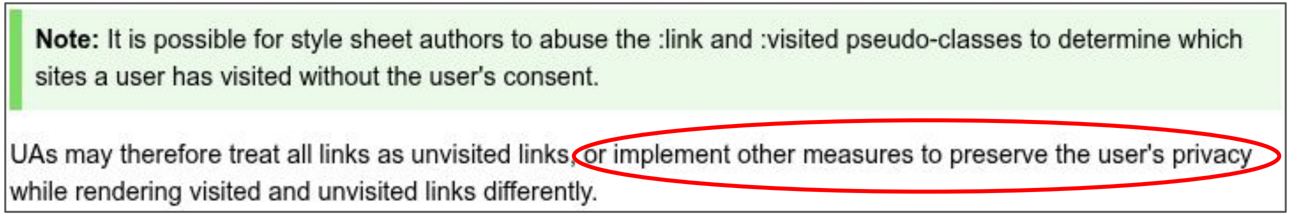
What happened in the last year?

kyraseevers@ launched visited links partitioning in Chrome (specifically: triple-key partitioned `:visited` links with self-links support):

- https://developer.chrome.com/blog/visited-links

Why talk about this today?

Partitioning visited links isn't fully specified because it's only covered by a vague note in the CSS spec, and a non-normative example in Selectors 4. We wanted to share details about the implementation and issues we run into to help other implementations avoid some pitfalls.



Note: It is possible for style sheet authors to abuse the :link and :visited pseudo-classes to determine which sites a user has visited without the user's consent.

UAs may therefore treat all links as unvisited links, or implement other measures to preserve the user's privacy while rendering visited and unvisited links differently.

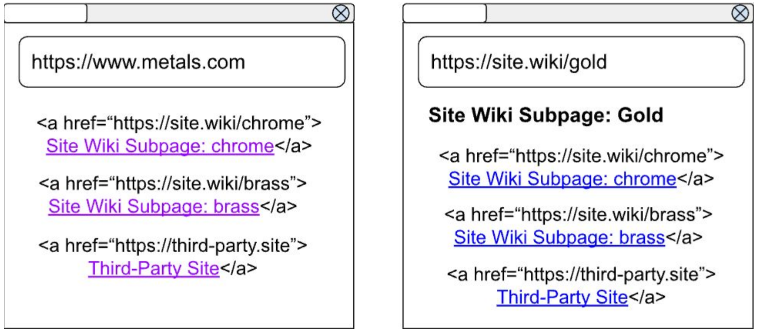
High-level design of :visited link partitioning

aka *Triple-key partitioning with self-links support.*

Triple-key partitioning: Separating state by top-level site (privacy) and current document origin (security), similarly to other storage mechanisms (e.g. local state such as `localStorage`, `IndexedDB`, Service Workers, HTTP cache, …)

Self-links: Visitedness is different from other partitioned mechanisms because each navigation has two parties (source & destination origin) which learn about the navigation. Self-links means we store visitedness in the partition of both the source & destination.

Without self-links



https://www.metals.com

<a href="https://site.wiki/chrome"> Site Wiki Subpage: chrome</a>

<a href="https://site.wiki/brass"> Site Wiki Subpage: brass</a>

<a href="https://third-party.site"> Third-Party Site</a>

https://site.wiki/gold

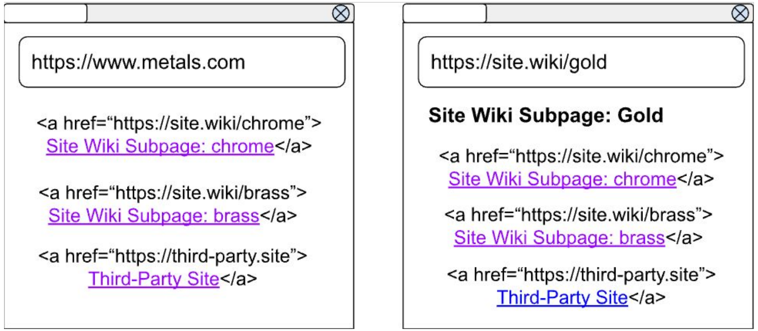
Site Wiki Subpage: Gold

<a href="https://site.wiki/chrome"> Site Wiki Subpage: chrome</a>

<a href="https://site.wiki/brass"> Site Wiki Subpage: brass</a>

<a href="https://third-party.site"> Third-Party Site</a>

With self-links



https://www.metals.com

<a href="https://site.wiki/chrome"> Site Wiki Subpage: chrome</a>

<a href="https://site.wiki/brass"> Site Wiki Subpage: brass</a>

<a href="https://third-party.site"> Third-Party Site</a>

https://site.wiki/gold

Site Wiki Subpage: Gold

<a href="https://site.wiki/chrome"> Site Wiki Subpage: chrome</a>

<a href="https://site.wiki/brass"> Site Wiki Subpage: brass</a>

<a href="https://third-party.site"> Third-Party Site</a>

Things to look out for when implementing #1

- This is *probably* obvious, but users will notice if `:visited` stops working for these:
  - `history.pushState()`
  - `<a rel=”noopener”>`
  - `<a rel=”noreferrer”>`
  - `<a target=_blank>`
  - Web Extensions using `chrome.history.addUrl()`
- Depending on your browser architecture, you might not have everything you want to construct <link URL, top level site, and frame origin>

Things to look out for when implementing #2

- Before making the switch to partitioned `:visited` links, we recommend starting to populate the partitioned database a few releases in advance:
  - This way the browser doesn't lose visitedness state when enabling partitioning.
- Self-links means we double the size of the visited database: each entry is stored in the partition of the source & destination site.
  - We found no observable performance impact, history DBs are small for a large # of users.
- Consider how the implementation interacts with the browser's process model
  - If links are properly partitioned, a compromised renderer shouldn't have access to URLs visited by the user on other sites.

Chrome's odds & ends

- **404 handling**: Chrome did not add 404s to history, which led to another potential :visited exploit. We are currently working to add those error navigations to history and :visited history, which has required a lot of technical solutions to avoid impacting existing clients of Chrome History.

- **Cross-device sync**: We chose not to pursue making :visited-ness eligible for Chrome Sync. Given that anything synced can provide a novel cross-device identifier, we worked with the security team to determine that while sites can reasonably determine who has previously visited on a single device, this is less the case cross-device, so the privacy rationale that supports triple-key partitioning (i.e. sites know what links a user has clicked on in this context before) does not hold up when synced.

- **WebView woes**: Finally, we are not ready to get rid of the mitigations yet because partitioning has not yet launched on Android Webview (for many reasons, one of which being that the infrastructure for how history works is entirely different and would require a new design).
  - Until this is addressed, we can't remove the complex :visitedness mitigations
  - https://developer.mozilla.org/en-US/docs/Web/CSS/Guides/Selectors/Privacy_and_:visited#limits_to_visited_link_styles

Chrome bugs discovered post-launch

List of bugs that we fixed post launch:

- https://issues.chromium.org/u/1/issues/411157351
- https://issues.chromium.org/u/1/issues/404174142
- https://issues.chromium.org/u/1/issues/400254620
- https://issues.chromium.org/u/1/issues/400828290

List of CLs that fixed these bugs:

- https://chromium-review.googlesource.com/c/chromium/src/+/6388415
- https://chromium-review.googlesource.com/c/chromium/src/+/6437866
- https://chromium-review.googlesource.com/c/chromium/src/+/6373930

The end