Partitioning :visited links:

What, like it's hard?

Sept 25, 2024

Kyra Seevers

bit.ly/visited-links

Agenda

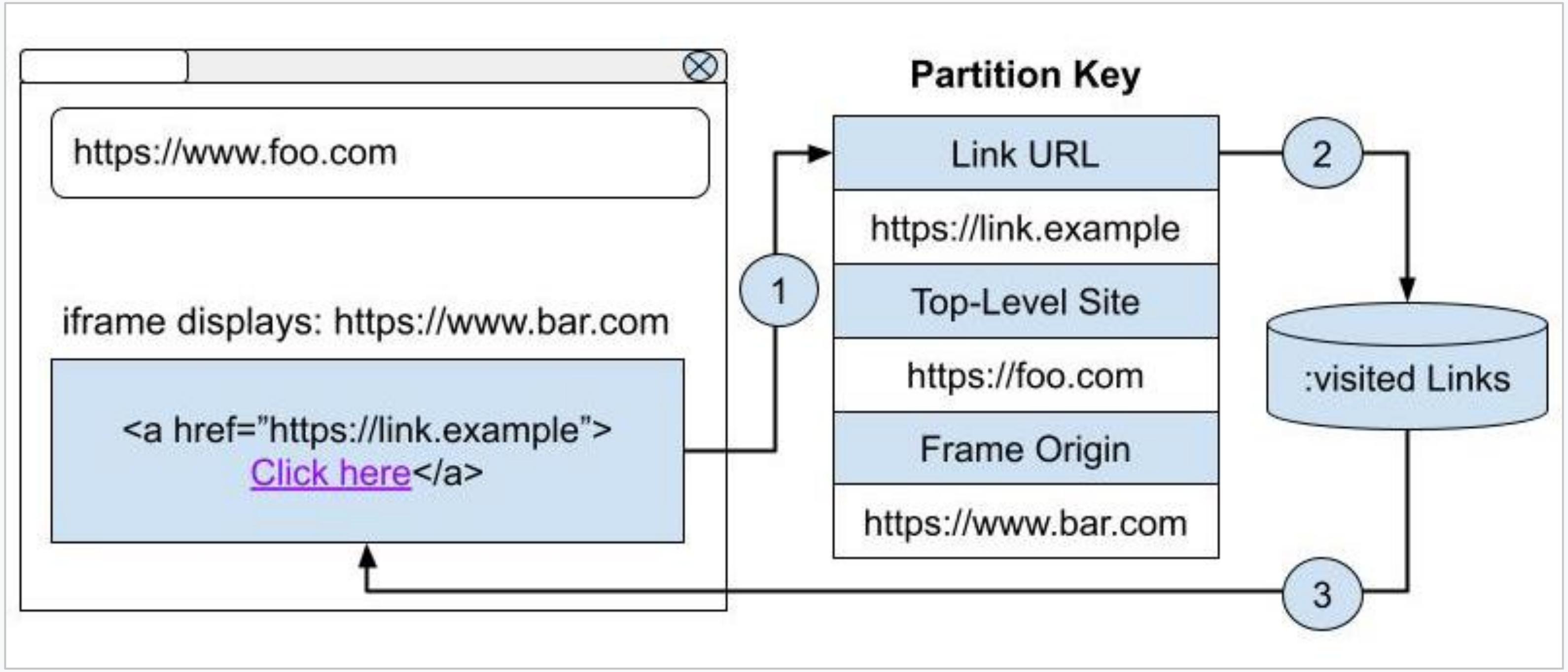
01

Background

Improve user privacy by eliminating :visited links history leaks.

Our Proposal:

The renderer styles a link as visited, if and only if we have visited that link from this top-level site and frame-origin previously.



02

Current Status

Google

Phase 1

- **Storing triple-key partitioned state in the new VisitedLinkDatabase**
  - Enabled by Default since Chrome 121

Phase 1

- **Storing triple-key partitioned state in the new VisitedLinkDatabase**
  - Enabled by Default since Chrome 121

Phase 2

- **User-facing partitioned :visited links**
  - Stable 1% Experiment Completed in Chrome 128
    - Android: 4.0% Partitioned vs. 5.5% Unpartitioned
    - Desktop: 6.4% Partitioned vs. 9.4% Unpartitioned
    - Great Performance Metrics on Both Platforms
  - Multi-armed Experiment with Self-Links in Chrome 130

Phase 1

- **Storing triple-key partitioned state in the new VisitedLinkDatabase**
  - Enabled by Default since Chrome 121

Phase 2

- **User-facing partitioned :visited links**
  - Stable 1% Experiment Completed in Chrome 128
    - Android: 4.0% Partitioned vs. 5.5% Unpartitioned
    - Desktop: 6.4% Partitioned vs. 9.4% Unpartitioned
    - Great Performance Metrics on Both Platforms
  - Multi-armed Experiment with Self-Links in Chrome 130

Phase 3

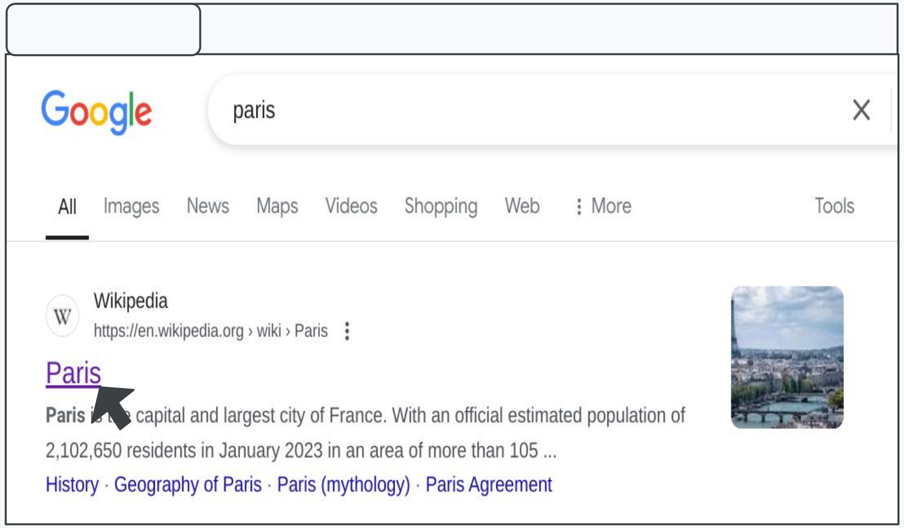
- **Incremental improvements post launch**

03

Implementation and Challenges

“But what about self-links?”

Chrome Case Study: Self-Links


paris

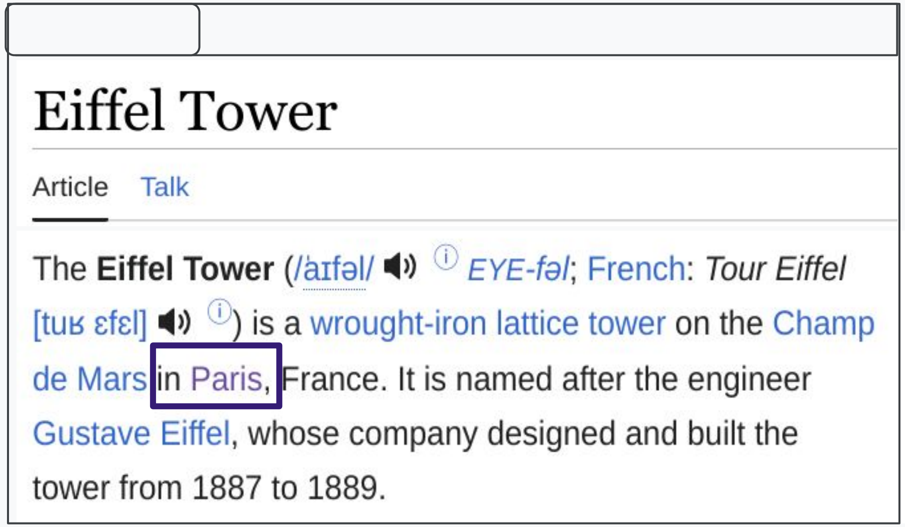
All Images News Maps Videos Shopping Web More Tools

Wikipedia
https://en.wikipedia.org › wiki › Paris

Paris

Paris is the capital and largest city of France. With an official estimated population of 2,102,650 residents in January 2023 in an area of more than 105 ...

History · Geography of Paris · Paris (mythology) · Paris Agreement


Eiffel Tower

Article Talk

The **Eiffel Tower** (/ˈaɪfəl/ EYE-fəl; French: *Tour Eiffel* [tuʁ ɛfɛl]) is a wrought-iron lattice tower on the Champ de Mars in Paris, France. It is named after the engineer Gustave Eiffel, whose company designed and built the tower from 1887 to 1889.

Link URL	wikipedia.org/paris
Top Level	google.com
Frame Origin	google.com

Link URL	wikipedia.org/paris
Top Level	wikipedia.org
Frame Origin	wikipedia.org

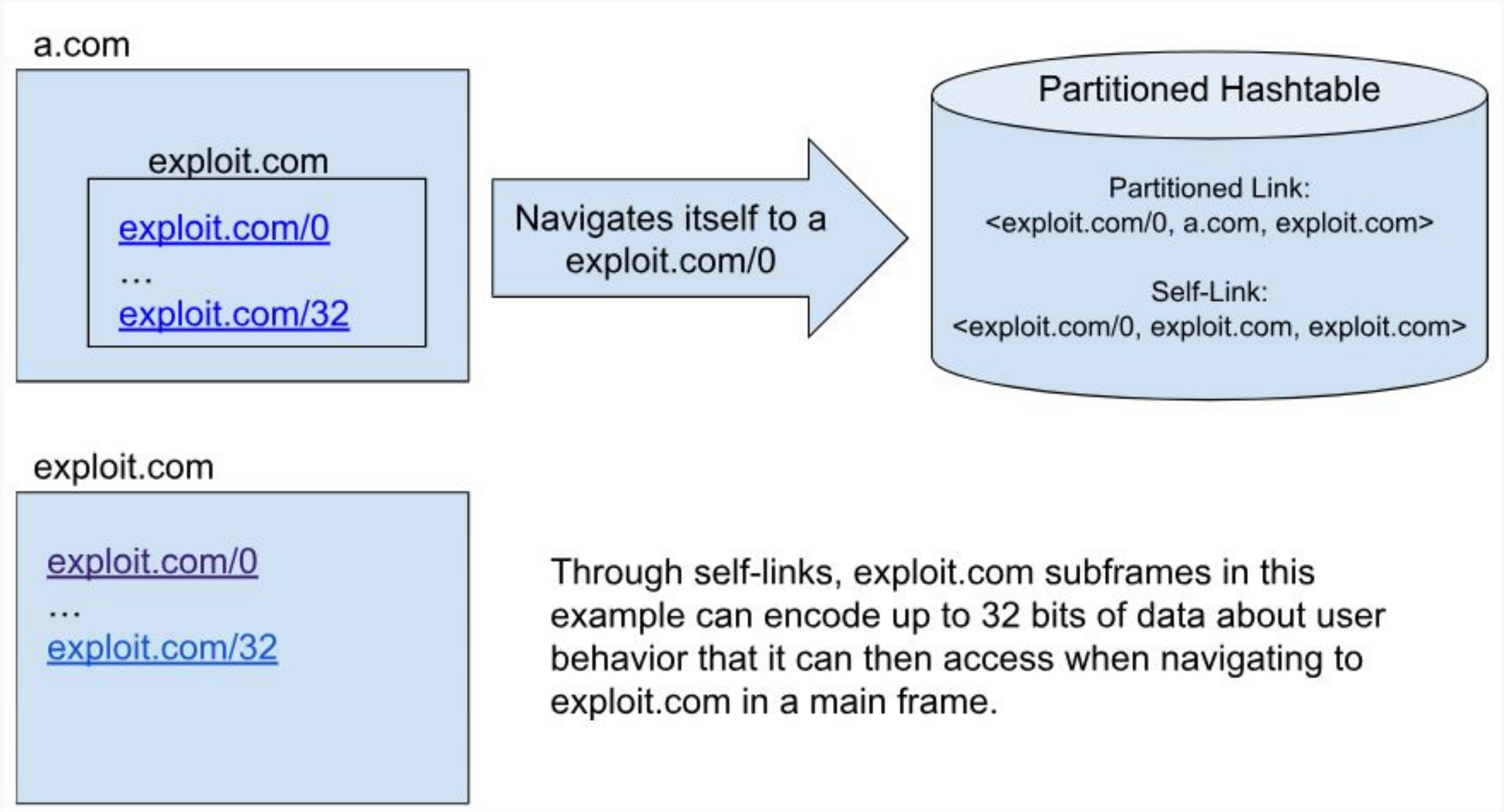
Chrome Case Study: Self-Links

However, this conflicts with our proposal of styling links as :visited if and only if we have visited them from this context before.

But we had a lot of feedback from external stakeholders that “self-links” can be valuable.

So the question became: **“How do we implement self-links without compromising our privacy and security boundary?”**

Chrome Case Study: Self-Links

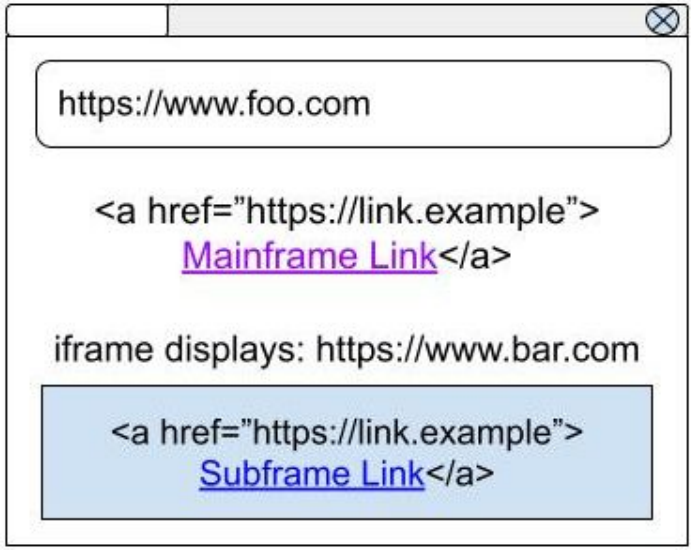


Through self-links, exploit.com subframes in this example can encode up to 32 bits of data about user behavior that it can then access when navigating to exploit.com in a main frame.

Chrome Case Study: Self-Links

Solution:

We only support **self-links for top-level frames** and **same-origin subframes.**


https://www.foo.com
<a href="https://link.example"> Mainframe Link</a>
iframe displays: https://www.bar.com
<a href="https://link.example"> Subframe Link</a>

Mainframe Key

Link URL
https://link.example
Top-Level Site
https://foo.com
Frame Origin
https://www.foo.com

Subframe Key

Link URL
https://link.example
Top-Level Site
https://foo.com
Frame Origin
https://www.bar.com

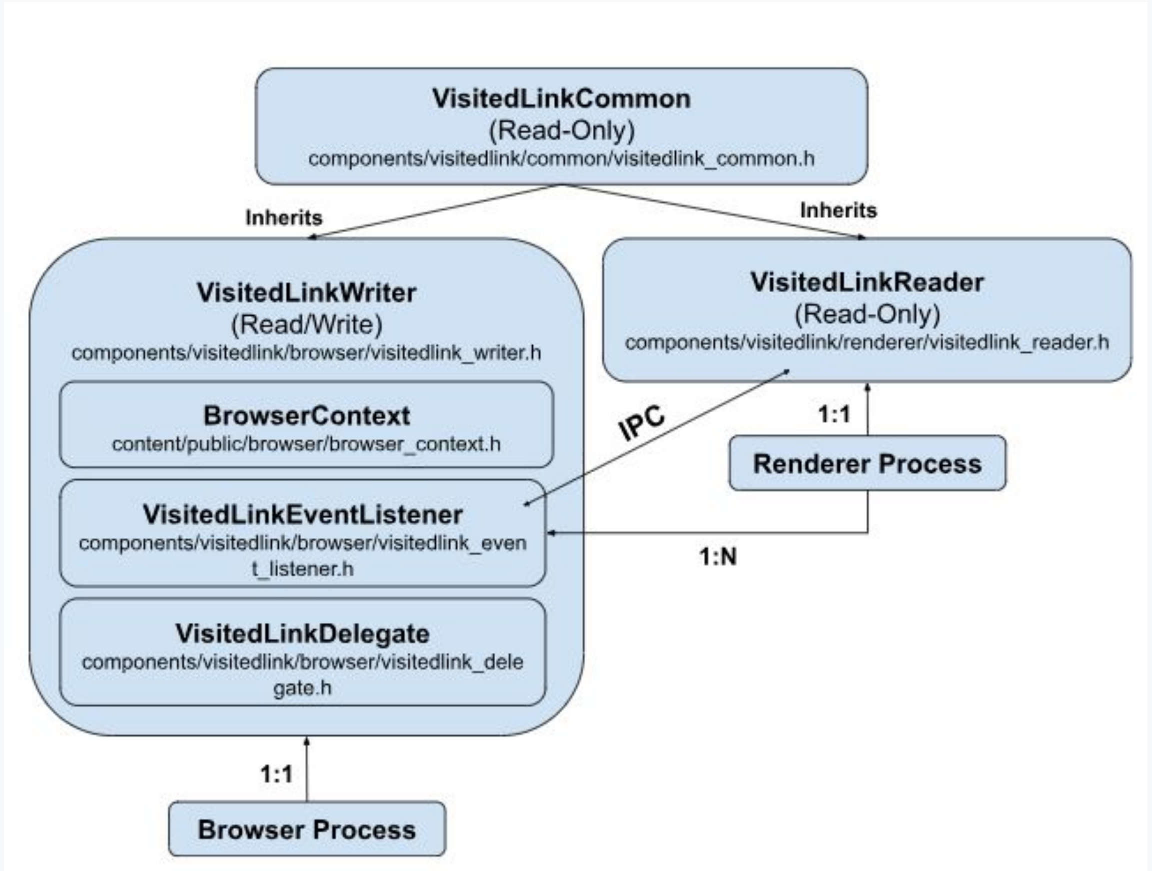
Self-Link Key

Link URL
https://link.example
Top-Level Site
https://link.example
Frame Origin
https://link.example

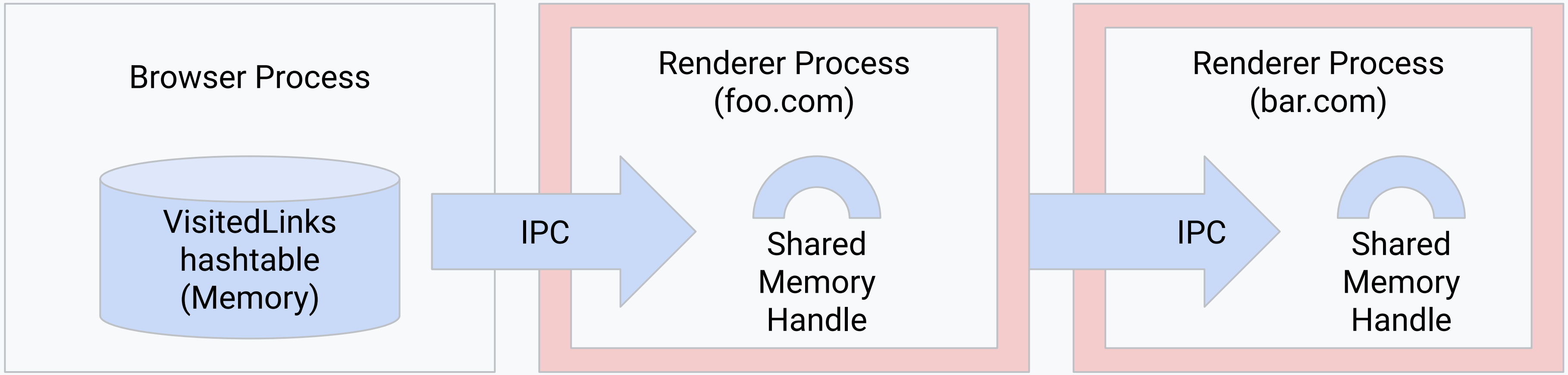
Google

“But what about renderer compromises?”

Chrome Case Study: Renderer Compromises


VisitedLinkCommon
(Read-Only)
components/visitedlink/common/visitedlink_common.h
Inherits
Inherits
VisitedLinkWriter
(Read/Write)
components/visitedlink/browser/visitedlink_writer.h
BrowserContext
content/public/browser/browser_context.h
VisitedLinkEventListener
components/visitedlink/browser/visitedlink_event_listener.h
VisitedLinkDelegate
components/visitedlink/browser/visitedlink_delegate.h
VisitedLinkReader
(Read-Only)
components/visitedlink/renderer/visitedlink_reader.h
IPC
1:1
Renderer Process
1:N
1:1
Browser Process

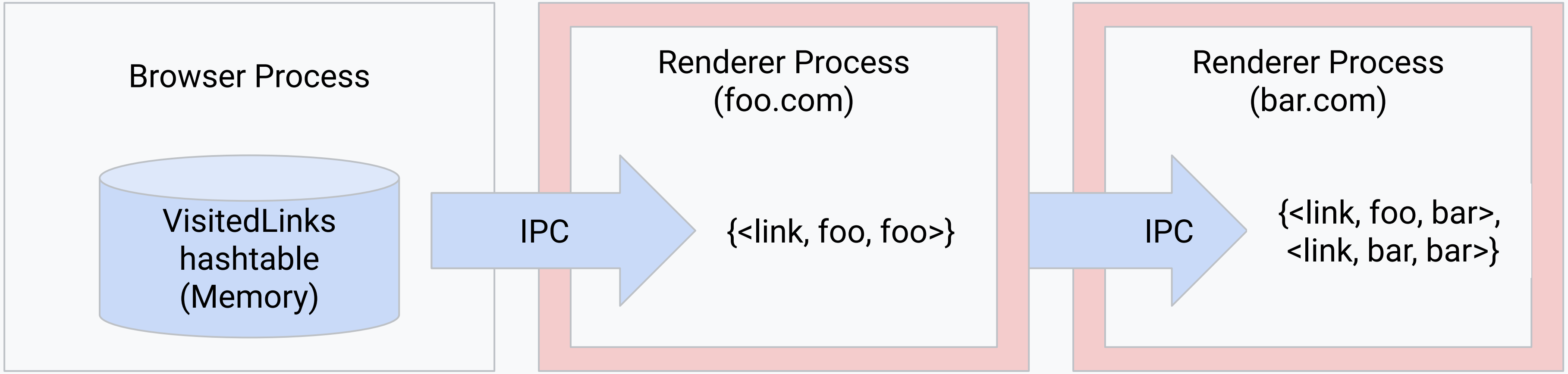
Chrome Case Study: Renderer Compromises


Browser Process
VisitedLinks hashtable (Memory)
IPC
Renderer Process (foo.com)
Shared Memory Handle
IPC
Renderer Process (bar.com)
Shared Memory Handle

Chrome’s :visited links are stored in a hashtable in memory.

A shared memory handle is sent via IPC to each Renderer Process.

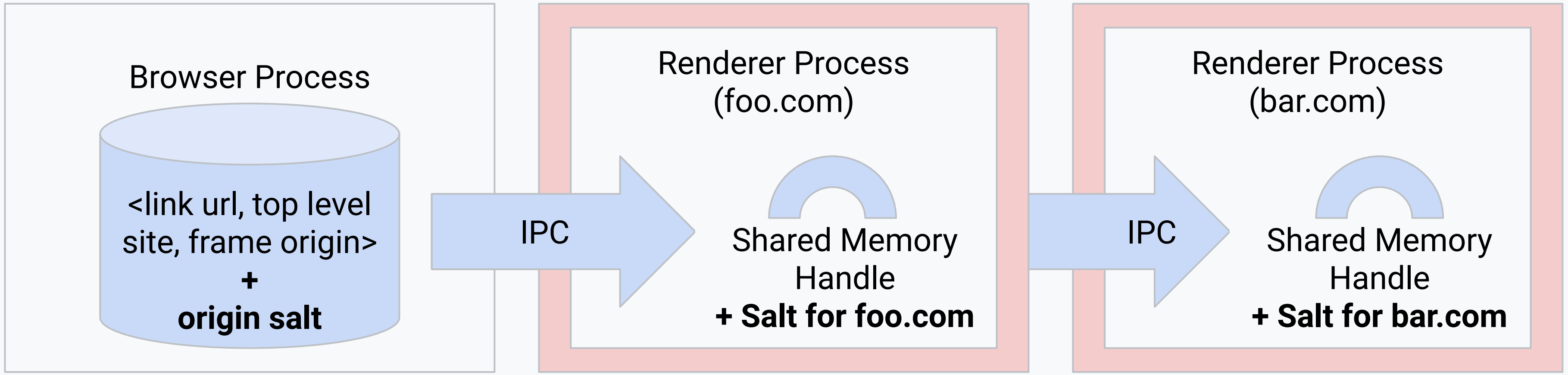
Chrome Case Study: Renderer Compromises


Browser Process
VisitedLinks hashtable (Memory)
IPC
Renderer Process (foo.com)
{<link, foo, foo>}
IPC
Renderer Process (bar.com)
{<link, foo, bar>, <link, bar, bar>}

We pitched several mitigations including "pre-filtering" or only sending each navigation request the links which matched its own triple-key.

Unfortunately, these were all too inefficient for Chrome.

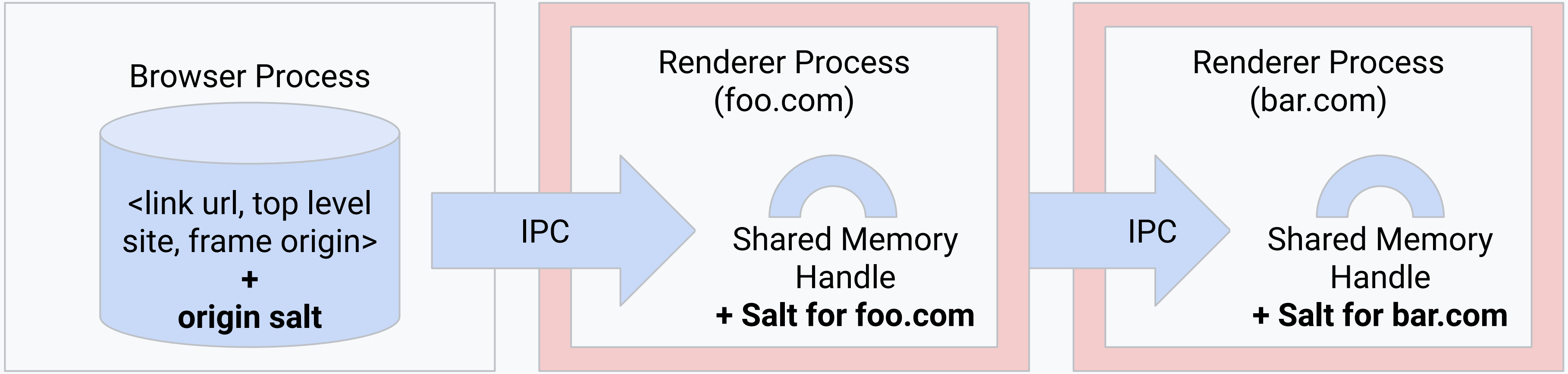
Chrome Case Study: Renderer Compromises



Our solution is "per-origin salts" where each triple-key gets hashed with an additional salt corresponding to its frame origin.

Each Render Process receives the salt corresponding to its origin prior to load, so it has the ability to "read" only its own origin's hashes.

Chrome Case Study: Renderer Compromises



To avoid **race conditions**, we do not determine or send per-origin salts during hashtable build.

Once build completes, we **query every RenderProcessHost** for its origin (or pending cross-document origin commits) and IPC its per-origin salt.

04

Frames, Frames, and More Frames

“But what about other types of frames?”

Chrome Case Study: Special Frames

Frame Type	Click #1: Iframe		Click #2: Credentialless		Click #3: Sandboxed		Click #4: Fenced	
iframe	click	visited		visited		unvisited		unvisited
credentialless		visited	click	visited		unvisited		unvisited
sandboxed		unvisited		unvisited	click	unvisited		unvisited
fenced		visited		visited		unvisited	click	unvisited

The experiment contains four frames
that **all share the same triple-partition key**.

Chrome Case Study: Special Frames

Partitioned Visited Links can be understood in 2 parts:

(1) What we **store**
(2) What we **style**

	Iframe	Credentialless	Sandbox	Fenced
Store	Yes	Currently, Yes Plans to make No	No	No
Style	Yes	Currently, Yes Plans to make No	No	Currently, Yes Plans to Make No

05

The Future

The Future

Phase 3 - Still open for suggestions!

- **Incremental improvements post launch**
  - Potentially integrating blink::StorageKey for reliable nonce calculations and maybe capturing even more “state lost”
  - Potentially improving corner cases with BFCache + restores
  - Potentially shipping on iOS

06

Call for Input

The Future

CSS Selectors Level 4:

- “Since 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 treat all links as unvisited links or implement other measures to preserve the user’s privacy while rendering visited and unvisited links differently.”
- <link url, top-level site, frame origin> vs. blink::StorageKey
- Any other implementation questions, concerns, struggles that other browsers have come across?

What did we miss?

We would love your feedback, thoughts, questions or concerns!

Special Thanks To:

Artur Janc, Mike Taylor, WebAppSec WG, and Legally Blonde

Where To Give Feedback:

File an issue on the explainer