Minutes: Private Cross-Origin/Site Prefetch (TPAC 2025)

Session chair: Robert Liu (elburrito@ [chromium.org | google.com])
Slides: [Public, chromium.org] Privacy-Preserving Prefetch (TPAC 2025)

Goal of talk: quick presentation, speculation rules, what the issue is, time for discussion
Not being recorded, but earlier version of this presentation in the Web Perf working group

Goal of today—the spec for prefetch and ip anonymization is underspec, Mostly implementation defined. Need more in privacy and what needs to be anonymized. Currently only one implementation of a proxy that does this. If more people want to implement, let Robert know

(Agenda Slide)

(Intro Slide)

Prefetch & Prerender

These are the two types of speculative loads defined in API

Prefetch—the browser is downloading just the resource and no subresources
Prerender—resource is downloaded, as are subresources, scripts may or may not be executed. Page is rendered in the background

Prefetch is less expensive in terms of network usage and doesn't use renderer

Rule types

URL list rules
JSON document in script type. Specify type of speculation and list URLs

Document rules
Instead of specifying actual URL, use matching for documents

Header-based speculation
Can be defined in JSON in a header instead of HTML

Where we're concerned with for the rest of this presentation, cross-origin prefetches

Google Chrome uses IP anonymization proxy in some usecases.

```
anonymous-client-ip-when-cross-origin
```

`referrer_policy`

Mostly used by Google Search, but 3P referrers can use this if the client has enabled the extended preloader setting in Chrome

A few blog posts about this.

12% of navigations in Chrome and 12% of origins use speculation rules (up and to the right)

Etsy, Cloudlfare, WordPress, Shopify all use speculation rules and like them

(Demo, speculation rules dev tools)

In DevTools panel, go to application panel, in background services, see speculative loads. See what rules are defined on a page and current status of any speculations. We go in and make search request, we can se that after a little bit that there are a few speculation rules and a few of them are ready on this page and if you click them (this happens very fast in the devtools panel) goes from ready to success right before the speculation is used. To demonstrate that it happened, in network panel, loading time looks instant

Does the network panel tell you that this was used?

I haven't looked, I have a slide further back about identifying prefetched and prerendered pages Can do it through JS or do it through the header we're sending

Cross-origin/site prefetch

Prerendering out of scope for this talk

Don't need to focus on same site, not crossing security boundary

It depends on the security boundary, the referrer knows some information about the client, because it's a speculation, not a load triggered by navigation, not a reflection of user intent, so some information about the user should be hidden when performing the speculation to the origin server

Referrer policy is one way to prevent information leakage. Origin won't know which referrer even is sending the speculation. Can hide other information like client IP address should use a proxy to do that. Just saying that again, client IP address is a pretty good and stable identifier for user location sometimes even user identity if it's stable enough. Pretty commonly used for cross-site tracking as I understand.

MDN defines a few kinds of referrer policies

Limits of prefetch spec

Spec doesn't specify how to achieve IP annon. If there are ways to do it without a proxy that'd be interesting, but future work to be done here. Would like to get more specific on that.

What can be standardized?
Write it down and see if someone objects
Does there need to be a second implementer?
That'd help
Can write down before a 2nd implementer and get feedback

A bit about what Google Chrome is doing

Operating Privacy Preserving proxy for corss-site. Search is the main user, 3P sites can use it if a certain setting is enabled in Chrome (extended preload). Pretty good at reducing load times

We talked a little about this in Web Perf. Why does this require an extra user setting? If the referrer isn't Google, may not want to send traffic to Google. Isn't it a 2 Hop Proxy?

No it's not. I'll talk about 1-hop 2-hop. Google not currently IP blinding like it would for a 2-hop proxy. Because 3P sites are using this not just Search, the proxy learns more information about the referrer the user is navigating to, and client IP address. More infor about browsing history

So previous history about other partners, they're just doing same-site not prefetch

To clarify, one implementation of IP annon proxy for prefetches

Can you talk more about who logically operates the proxy? Google operates it for Search in Chrome. Is the Proxy a Chrome proxy or a Search proxy? Important proxy. Is seeing search results when you get down to it. In one model, Search responisbility

This is related to browser-provided vs referrer provided proxies. As it exists, it's both, but I see it as a browser provided proxy. Search engineers are not maintaing it

That's just Conway's law. Why is it a browser responsibility?

That's a reflection of current state

Thinking about this from Mozilla's perspective (which we cannot afford to do), we would then recieve a slew of information about IP addresses to websites to a first approximation at the moment that's Search. We would have information about people's Search history. That seems like new information we wouldn't otherwise have. If Search operates it, Search is already seeing

results and IP addresses, don't learn anything new. From my perspective, I thought this would be website provided

It's a more natural alignment from a privacy perspective and cost perspective to be a referrer provided proxy.

*Who obtains the benifit is interesting in this as well. User benifit in *snap* instant load, but pay for that in extra bandwidth, work your browser is doing, that could ultimately do nothing for you, waste effectively. Calibrating that cost against instantaneous load is interesting. Arguably Search experience is superior*

Want to push back on information you have already

It's the user's browser, not ours

You did process that search for BF cache

Difference between browser we provide to users and Mozilla operating a service to do this. Deliberate difference

Seaprate cost from privacy perspective. If we want the browser to operate a proxy, a 2-hop the browser vendor woldn't gain this knowledge, whereas if you let the referrer, a 1-hop proxy is fine. A separte conversation from if a browser vendor can operate 2-hop versus referrer 1-hop. Some origins are deeply uninterested in you leaving

I want to separate the question of 2-hop or not. We're starting from the assumption that someone's going from without a proxy and accelerate the navigation to a new website

With 2-hop proxies in the mix, none of this is relevant (who's learning what)

Could I ask two simple questions? One, if I do a Google search myfavoritewebsite.com and the main resources when logged in is different than logged out, do I get the not logged in resource when fetched

No in spec, but we do not use prefetches for results with a cookie or service worker because they'll be different from the annon version

So if I prefetch something with no cookie and does use proxy, if I navigate to that site, are the subresources loaded with or without the proxy?

My understanding is they're loaded without the proxy

What we require in the spec

Coming to the time where we're implementing speculation rules. Now's a good time to workshop how this should be implemented

Want tov verbally highlight. A future Safari implementation may use something along the lines of iCloud Private Relay, Mozilla may use their VPN product

I like the fact Mike's willing to own up to the behavior of other people (joke)

Threat Model

Main privacy principle, none of the parties involved can learn anything new about the user as a result of prefetching a website

Specific things

- IP Address
- Cookies
- Service Workers
- Visited Link Status
- Browser History leakage

They seem to also learn the time of the search, go to search, wait 10 minutes, click, now they learn aggragate search results, how long it takes to click, all new

Yah, those are all good considerations

Depending on the first one, may change based on browser configration, like how eager the speculation is, when it appears in the viewport, but good considerations to throw in

Clarification question. If I clikc a link in Search today, does the origin know the exact search result?

No, most search engines today strip that

Search result may be unique per person

What's interesting about this one, think about threat model, there's a difference between the information the linking site might choose to share with the linked site anyway. Some of that information may choose to share by activating this feature, useful to know, but from a user perspective this is all the information the search engine already has

Who knows what and when

- **UA** - Client IP and prefetched URL
- **Referrer** - Same

- **Proxy**
    - One-hop, CLient IP and prefetched URL
    - Two-hop, first hop only knows client IP, second hop knows only URL

Does it get the URL or are you tunneling

In the tunnel, the client will send the full URL.

- **Origin** - ideally nothing, but the spec only covers IP annon

Origin learns the URL

Does the referrer know the prefetched URLs, know which ones are prefetched or what could prefitched

Think it's the later

I think otherwise it'd be a privacy issue, if you don't prefetch cookies, referrer may learn what sites aren't supported

Good question for the spec

Information leaks from direct prefetches

This is what x.com is doing for cross-origin prerendering. So instead if you're using an HTTP2 proxy you're not leaking IP address. Other stuff we should talk about more.

Other considerations from proxy operation

- Anti-abuse and fraud. Don't want open proxy for internet. Various checks in place to ensure people aren't able to abuse access to Chrome's proxy
- DoS prevention, some amplifaction concerns turning into multiple speculations. Don't want to overwhelm sites directing Search firehose at them
- Traffic control. Sites can control fraction of prefetched traffic using `.well-known/traffic-advice`

I think there's nothing from stoping you following a redirect as long as you maintain the proxy. The question about cookies and how much you pull in all the way through

In the ad ecosystem, client-side redirects which are harder to do predictible if not using meta refresh header, parsing the content/response

That gets you into prerender

- Geolocaiton, important consideration for getting users right content for their locale. Currently configurated for country level geo, Lets site owners know what countries they're recieving prefetches from. Another mechanism for sites to know they're getting prefetched Search traffic

At this point, not assmuing anything the target website does to enable this feature. Besides traffic control, any other controls can be put in place?

I have seen that consideration searching Chrome prefetch proxy, my understanding is yes. May not be in scope for the proxy, but sounds like internal `geoip` question.

Let's say I prefetch and 15s later the entire site changes then I click. Not just caching question, obtained via different path. What extent is this a problem? Do you check with the website? Here's the e-tag, should I update?

I don't have a stasifying asnwer to that question myself. I think it's how our implemetnation is doing this caching

Target sites know this is coming from a prefetch request.

Regarding the freshness of content, if cookies is changed on taget sites, changes, and prefetches times out

Can see it being relatively constrained

Is it down to cache control?

Cache of prefetch results

Subject to cache control lifetime

A lot of HTML documents don't use cache control. Presumably you don't want those used by prefetch. If it's not cachable at all can you ever use it?

(laughing)

Future directions

Incomplete summary of Monday's discussion

- Use HTTP Connect to make sure proxies don't tamper with resources
- Spec shouldn't require browsers to ship a proxy, incentive & cost aligned for referrers that want advantages to pay for it themsleves
- How do UAs know t trust a proxy?

- For a browser, it's their browser
- For referrer, the proxy is same site as referrer

But that would leak if the browser has cookies or proxies

Don't see the request it's tunneled

Still see how names

The decoy prefetch problem

Trying to avoid a browsing history leak. If we do not prefetch requests that the UA has cookies or a Service Worker for, proxies know that information about the UA. In that case, we do connect to hostname, no those results are not usable. Thrown out

Nice (said in italics IRL)

Question about referrer proxy. Lots of things can go wrong. Do we envision it would be reasonable for CDNs like Cloudflare to offer proxies like this?

Yes, this is a CDN opportunity for sure. Bread and butter for these guys

If we're asking each web provider to roll their own seems like a recipe for disaster. Turnkey solution seems more tractable for adoption

Wanna do a startup with me? (joke)

Part of the tricky bit moving to 2-hops, natural is 1-hop is browser, 2-hop for referrer, but don't get great anon if both are run by, say, Google

Would CDNs need 2-hop or contractual and treat as 1-hop?

This is why I think 2-hop is irrelevant, there are some questions about what to do, but don't need the special stuff. Most interesting question is avoiding open relay. Referrer provides it, get token to connect through good for maybe even single resource. Proxy won't be able to tell which resource you're requesting, but have ability to understand what's being requested and bound to particular host. Can imagine search results page here's your prefetch rules, here's the proxy, here's how to authenticate. Totally workable, oeprationally challenging to a degree, but can be bound in client

As much as the referrer is willing to send those tokens is equal to charge

Can do it as much as they want. 10 links on every webpage? Good. Control exposure on both sides

Open Questions

First two carried over from Monday. Last two

How much should the spec say about proxy implementation, and how wants to make one?!

First one, as much as possible, second is ask the CDNs

Seems like CDNs, may not need connect proxy

Thought about OHTTP, not great as a message oriented proxy because not dierctly connecting to origin server potential for tampering wwith resources, CDN to CDN maybe

Need a gateway though

But the gateway needs ot be affiliated with the website and can rewrite them

If the CDN is hosting it, yeah

I think the premise here is that the target website doesn't chagne. If the target website is willing to change, great number of things to change

One to may requires keys for each gateway

Consistency checks on the keys

Had another question, currently, it's H2 only. Don't know anough protocols, but can it support H3?

Doens't matter, just connect, whether it's old school, H2, H3, doens't matter

Could all be done with masks over H3

As I understand it right now it's only H2. Some discussion about using this for preconnect, but loading more through H2 tunnel but doesn't do as much multiplexing

Multiplexing is mostly OK for this doesn't really come into play

If you use thi with preconnect, but don't use it to load page, not useful preconnect

Some multiplexing, is this a cache, is it not

Next Steps

Spec updates

- Specify proxies must annon prefetch requests and remove as much identifying info as possible
- Specifiy about browser-provided vs referrer provided (browser may provide)
- Include in spec the `.well-known/traffic-advice` profile. Can IP block bad proxies that don't do that
- Add support for referrer provided in spec. Referrer mus be same site as proxy. Probably some other stuff too

Which group owns the spec updates, if yo know, shout!

All WHATWG and HTML, speculation rules in HTML, fetch does the busy work

Well, if anyone wants to join me in making the next steps happen, I MADE A LOT OF BUSINESS CARDS, I thought they'd be a bigger part of my life as a kid, please come get one and make my childhood dreams come true

Is the scoping of .well-known right here or something more granular?

Don't know if other options

It's per origin

What's the usecase? Different considerations for different URLs

Could invent syntax to do it

Does Chromium currently support cross-origin prefetch without this?

Yes

Enabled by default

I don't love exposing annon IP as part of the spec network level requirements, but can talk about that

Wrap Up