Chrome extension manifest v3 proposal #338

New issue

Open **greghuc** opened this issue on Dec 13, 2018 · 143 comments

greghuc commented on Dec 13, 2018 • edited by gwarser

Description

This issue is a heads-up on the proposed Chrome extension manifest version 3, which will have a significant impact on ad-blockers.

There is a tracking bug at: https://bugs.chromium.org/p/chromium/issues/detail?id=896897

In-progress design doc: https://docs.google.com/document/d/1nPu6Wy4LWR66EFLeYInl3NzzhHzc-qnk4w4PX-0XMw8/edit#

Might be worth a review, and giving feedback via the tracking bug.

regarding all dev. on dnr - https://bugs.chromium.org/p/chromium/issues/list?can=2&q=declarative+net+request

"Migrating to Manifest V3" (timeline) - https://developer.chrome.com/extensions/migrating_to_manifest_v3

👀 23

Assignees

No one assigned

Labels

Chromium

something to address

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

42 participants

uBlock-user changed the title ~~Heads-up: Chrome~~

uBlock-user added the discussion label on Dec 13, 2018

and others

uBlock-user commented on Dec 13, 2018 Member

Yes this was announced back in October - https://blog.chromium.org/2018/10/trustworthy-chrome-extensions-by-default.html

Not much for uBO to do though..

👎 1

jspenguin2017 commented on Dec 13, 2018 • edited

I'm not sure how are they planning to change the background process and origin access, it might cause some issues. Beside that, I don't see anything else that can cause problems. As for remotely-hosted code, it really should've been forbidden since the beginning.

uBlock-user added something to address and removed discussion labels on Jan 6, 2019

joey04 commented on Jan 9, 2019

It appears that Google is phasing out the webRequest API in favor of their new declarativeNetRequest.

The current Manifest V3 draft states:

> request, rather than have Chrome forward the request to the extension. Thus, instead of the above flow where Chrome receives the request, asks the extension, and then eventually gets the result, the flow is that the extension tells Chrome how to handle a request and Chrome can handle it synchronously. This allows us to ensure efficiency since a) we have control over the algorithm determining the result and b) we can prevent or disable inefficient rules. This is also better for user privacy, as the details of the network request are never exposed to the extension.

> This API is currently being implemented, and will be available to both the current manifest version and Manifest V3, but will be the primary way to modify network requests in Manifest V3.

Looking at the new API page, it's a rather different way of doing things. I would expect big changes needed to uBO d[illegible] (Plus there's [illegible] from background page to ServiceWorker.)

uBlock-user added the Chromium label on Jan 9, 2019

👍 4

uBlock-user commented on Jan 9, 2019 • edited Member

Another change would be dynamic Content Scripts similar to what Firefox has where extensions can inject contentScript and have it execute before page finishes loading.

gwarser commented on Jan 9, 2019 Member

> declarativeNetRequest

Basic ABP-like syntax, 30000 filters MAX...

uBlock-user commented on Jan 9, 2019 Member

gwarser commented on Jan 9, 2019 Member

I don't see limits to the number of the rules in declarative**Web**Request. declarative**Net**Request have better syntax, but number of rules is limited.

gorhill commented on Jan 9, 2019 Member

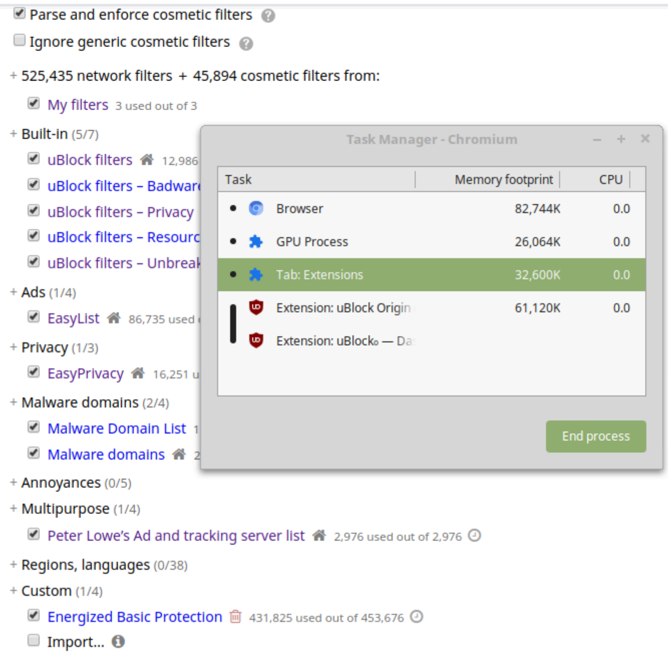
> It appears that Google is phasing out the webRequest API

It is explicitly stated this is meant to phase out `webRequest` API? If so, where? Quickly perusing the page, I don't see such phase out being stated. If there is really such phase out planned, and if the declarativeNetRequest is strictly an implementation of static filtering à la ABP, this would be the death of uBO and uMatrix.

There is no way to transpose either either dynamic filtering, dynamic URL filtering, per-site/per-scope switch logic (let's refer to all these as "dynamic filtering"), into static filters. Dynamic filtering logic requires an arbitrary amount of block/allow rules overriding other block/allow rules based on *specificity*. There is no concept of specificity in static filtering -- and even more, there is no concept of dynamic filtering rules relinquishing filtering to static filters (dynamic filtering's `noop` rules).

> Basic ABP-like syntax, 30000 filters MAX...

And there I was recently testing how uBO handled over half a million network filters with the 3rd-gen HNTrie...



☑ Parse and enforce cosmetic filters
☐ Ignore generic cosmetic filters

+ 525,435 network filters + 45,894 cosmetic filters from:
  ☑ My filters 3 used out of 3
+ Built-in (5/7)
  ☑ uBlock filters 12,986
  ☑ uBlock filters – Badware
  ☑ uBlock filters – Privacy
  ☑ uBlock filters – Resourc
  ☑ uBlock filters – Unbreak
+ Ads (1/4)
  ☑ EasyList 86,735 used
+ Privacy (1/3)
  ☑ EasyPrivacy 16,251 u
+ Malware domains (2/4)
  ☑ Malware Domain List 1
  ☑ Malware domains 2
+ Annoyances (0/5)
+ Multipurpose (1/4)
  ☑ Peter Lowe's Ad and tracking server list 2,976 used out of 2,976
+ Regions, languages (0/38)
+ Custom (1/4)
  ☑ Energized Basic Protection 431,825 used out of 453,676
  ☐ Import...

Task Manager - Chromium

Task	Memory footprint	CPU
Browser	82,744K	0.0
GPU Process	26,064K	0.0
Tab: Extensions	32,600K	0.0
Extension: uBlock Origin	61,120K	0.0
Extension: uBlock₀ — Da		

Issues of performance and privacy lie with web sites, not
uBO -- so I don't feel concerned with the issues of privacy

uBlock-user commented on Jan 9, 2019 • edited

Member

https://docs.google.com/document/d/1nPu6Wy4LWR66EFLeYInl3NzzhHzc-qnk4w4PX-0XMw8/edit#heading=h.ypclvihky0p6

Read the paragraph that's titled DeclarativeNetRequest. They will keep webRequest API but it will not block, modify or redirect anymore, that will be handled by NetRequest API with v3.

They do plan to discourage the use of webRequest API through with its new replacement - https://docs.google.com/document/d/1nPu6Wy4LWR66EFLeYInl3NzzhHzc-qnk4w4PX-0XMw8/edit#heading=h.xe5njuo7voeb

uBlock-user commented on Jan 9, 2019 • edited

Member

this, can you shed some light there ? I don't think it is related to 30K ABP styled filters.

gorhill commented on Jan 9, 2019 Member

Look at "Rule", it's clearly made to implement ABP-compatible filters:

> '|' : Left/right anchor: If used at either end of the pattern, specifies the beginning/end of the url > respectively.
>
> '||' : Domain name anchor: If used at the beginning of the pattern, specifies the start of a (sub-)domain of the URL.
>
> '^' : Separator character: This matches anything except a letter, a digit or one of the following: _ - . %.
> [and so on...]

👍 1

joey04 commented on Jan 9, 2019 • edited

This new API is even worse than being restricted to 30k ABP-style rules -- the rules must also be in a single, bundled json file. Thus any rule changes means a full extension update. (Blockers restricted to this API would be "static" in every way.)

Thinking big picture, it's not that surprising. Last year Google started to bundle a lightweight ad blocker in Chrome, and I recall thinking that something like this would happen. On the bright side, perhaps this could be a boon to Mozilla, assuming they don't foolishly neuter their API in the same way.

👍 4

It is still a draft but is really a big downgrade if it stays this way. Looks to me like Google is doing this to reduce the amount of damage malicious extensions could do using this API by severely limiting the existing one and calling it a privacy and efficiency win....while of course giving more control to Google over what ads/trackers/requests can be blocked. This affects trusted and more advanced extensions like uBO the most.

Hopefully Mozilla and other browser makers don't follow suit on this..

jspenguin2017 commented on Jan 10, 2019 • edited

`declarativeNetRequest` in its current state is really underpowered and isn't going to be enough. However, assuming that they will implement a way to modify the rules and the 30k limit will be raised or removed, it looks like the only thing missing is RegExp rules.

Since we will be able to dynamically modify content scripts, we can still implement scriptlets and cosmetic filtering properly.

uBlock-user commented on Jan 10, 2019 Member

It's still not set in stone, so lets see where it goes.

uBlock-user mentioned this issue on Jan 21, 2019

General filter chit-chat Closed
DandelionSprout/adfilt#7

gorhill commented on Jan 21, 2019 • edited Member

The fact that they are planning to remove a proper blocking webRequest API with no word of an equivalent replacement is a sign of *intent*, that is, reducing the level of user agency in their user agent (aka Google Chrome).

How to do this? Use privacy/performance as Trojan arguments to rationalize reducing user agency over what all bloated web sites throw at people's user agents. That new declarativeNetRequest API seriously reduces what blockers can do, to the point they will become distinguishable only by their UI, not their capabilities. As a user, I personally wouldn't accept browsing the world wild web without the advanced features in uBO, I find this unthinkable.

There are no issue of privacy/performance with uBO, rather the opposite by giving back to users the power of clamping down on what web sites throw at them, so that argument is just plain fallacious as far as uBO is concerned.

Chromium got its webRequest API at a time it was trying to gain market share against Firefox (Sep 2011), where Adblock Plus, Ghostery, Disconnect, NoScript, and other such extensions were the most or among the most popular extensions on Firefox.

I don't expect Firefox to follow suit and also deprecate its own webRequest API.[1] I am confident uBO will still exist on Firefox.[2]

[1] Actually Firefox's own webRequest API is better designed as it's possible to return a Promise, which makes it possible to defer returning an answer to some point in the future.

[2] Which is already better equipped than Chromium's version of uBO -- example, example -- (and also better equipped than the Firefox legacy version).

👍 37 ❤️ 24

> is a sign of intent

Yes, but they're not doing this specifically targeting uBO, other blockers will be affected by this too, won't they ? So I still think it won't end up like this. Like you said, it will be the death of these two extensions, why would they bother doing that after all this time ? If they wanted to kill uBO, they could have done this years ago by deprecating APIs or limiting them.

gorhill commented on Jan 21, 2019 • edited Member

I'm not saying they are targeting uBO specifically, more that they are targeting the *capabilities expressed* in uBO/uMatrix.

What is being said now is that the maximum capabilities content blockers will be allowed come down to a maximum of 30,000 ABP-compatible filters.

Even without dynamic filtering and per-site switches, etc., uBO already enforces over 90,000 filters (which themselves go beyond ABP filter syntax) with just the default filter lists, not counting the regional one which may be activated automatically at first install, and other commonly used ones such as Fanboy Social. When I select only EasyList, uBO reports 42,000 network filters, so even EasyList alone won't be enforceable with the declarativeNetRequest API.

Beside the low maximum of 30,000, ABP-compatible filters have no sense of specificity, hence why dynamic filtering can't be implemented with such approach (and if they did it would be a pain to have to recompile the whole filtering profile when merely adding/removing a rule/switch through a click). Also, this makes it impossible to implement `important` filter option, which purpose is to override exception filters.

regard to manifest v3.

👍 8

This comment was marked as off-topic. Sign in to view

uBlock-user commented on Jan 21, 2019 Member

At this juncture, assuming it will get implemented, can you do something about it like not updating to v3 ? or should Chromium users be ready to abandon ship for good ?

gorhill commented on Jan 21, 2019 • edited Member

> should Chromium users be ready to abandon ship for good ?

I won't tell people what to do. I am pointing out that removing the blocking ability of the webRequest API means the death of uBO, I won't work to make uBO less than what it is now. I quote the document (my emphasis):

> ## WebRequest
>
> **Summary**
>
> In Manifest V3, we will strive to limit the blocking version of webRequest, **potentially removing blocking options from most events (making them observational only)**. Content blockers should instead use declarativeNetRequest (see below). It is unlikely this will account for 100% of use cases (e.g., onAuthRequired), so we will likely need to retain webRequest functionality in some form.

`webRequest.onHeadersReceived` (to implement disabling or restricting JS execution, static filters' `csp=` options, large media filtering, and other filtering capabilities).[1]

With this information on hands, everybody is free to decide for themselves.

[1] There are four listeners in the webRequest API which can be used in blocking mode: onBeforeRequest(uBO, uMatrix), onBeforeSendHeaders(uMatrix), onAuthRequired, onHeadersReceived(uBO, uMatrix). uBO uses two of them, uMatrix uses three of them.

👍 10 👀 1

This comment was marked as off-topic. Sign in to view

jspenguin2017 mentioned this issue on Jan 22, 2019

Chrome Manifest V3 discussion megathread Open
NanoAdblocker/NanoCore#238

Kusresa commented on Jan 22, 2019 • edited

I think the best thing people can really do for now is to get the word out to extension developers and browser developers (especially Google) that the proposed APIs and manifest should not be restricted to such an extent and that users should retain enough freedom and capabilities to easily control what to do with extensions and requests within their browser.

notice a shifting of how and what ads/trackers/requests get blocked and it will be near impossible to rollback the changes as the browser market leader has a low incentive to do so.

I don't want to sound too dramatic but the implementation of the requests API in the manifest v3 proposal as it is right now could be the beginning of something that will have wider implications on the web and users' ability to decide how they can browse it. Due to Google's position of power on the web and influence on websites it will almost certainly affect more than just Chromium/Chrome users.

👍 13

palexande commented on May 31, 2019

I find it a bit dubious that Microsoft Edge starts using Chromium code not too long ago, and now Google wants to prevent ad blocking. Looks like I may be switching back to Firefox, although I like Chromium (Iridium) better.

👍 1

hedenface commented on May 31, 2019

Not gonna downvote **@nsuchy** ..but would like to address some things:

> Chromium is like an operating system, hence it requires dedicated man hours to keep-up. As for backporting changes, I'm developing a system to conduct smart diff analysis that can see which code interacts with the extension and networking area vs others. Changes which do not affect the relevant areas (blocking web request API initially) will be automatically merged without review after official Chromium publishes.

Do you expect to have the ***very many*** developers required for this work paid based on donations from a *POPULAR* projects fork?

> After build passes the modified tests, it'll be published as stable otherwise if the diff analysis fails, manual look will be taken.

A diff analysis probably isn't a smart choice. What happens if you're only diffing on specific functions that rely on other functions? Or what happens when those functions change? Will you be maintaining this CI/CD pipeline that automatically determines which commits to merge and how they affect the specific area of interest?

> Changes will be kept to a minimum until the project is big enough to ensure security/other features backport easily.

As obviated by the responses to your comments here, I don't think the community - and I don't mean the world (only the people who have seen this issue [maybe 5 people? {that's a joke}]) *{- heck for all any of us know*

jitendravyas commented on May 31, 2019

Will the ad blockers stop working on other chromium based browsers too? like Brave, Opera, Vivaldi etc. ?

👍 1

> Not gonna downvote **@nsuchy** ..but would like to address some things:
>
>> I plan to fund using Patreon and/or Github sponsorship (launched a few days ago). I did not want donations but after multiple looks at it, Chromium is like an operating system, hence it requires dedicated man hours to keep-up. As for backporting changes, I'm developing a system to conduct smart diff analysis that can see which code interacts with the extension and networking area vs others. Changes which do not affect the relevant areas (blocking web request API initially) will be automatically merged without review after official Chromium publishes.
>
> Do you expect to have the ***very many*** developers required for this work paid based on donations from a *POPULAR* projects fork?
>
>> After build passes the modified tests, it'll be published as stable otherwise if the diff analysis fails, manual look will be taken.
>
> A diff analysis probably isn't a smart choice. What happens if you're only diffing on specific functions that rely on other functions? Or what happens when those functions change? Will you be maintaining this CI/CD pipeline that automatically determines which commits to merge and how they affect the specific area of interest?
>
>> Changes will be kept to a minimum until the project is big enough to ensure security/other features backport easily.

> [maybe 5 people? {that's a joke}]) *{- heck for all any of us know maybe this project takes off - if we could reliably predict that I don't think many of us would be in this field :) -}* but I suspect unless you have a more specific gameplan with very specific goals and objectives ***and a roadmap*** no one is going to look at this seriously.

That roadmap is going to need to be pretty stellar.

hedenface commented on May 31, 2019

@nsuchy OMG I'm sorry. I may have had a few drinks. Yes, directed specifically at **@w3engine**

😄 5

This comment was marked as off-topic. Sign in to view

This comment was marked as off-topic. Sign in to view

Morthawt commented on May 31, 2019 • edited

I have to use Chrome at work, but at home I refuse to make google chrome my browser. I use Firefox with uBlock Origin, Privacy Badger.

uBlockOrigin locked and limited conversation to collaborators on May 31, 2019

mapx- commented on Jun 12, 2019

uBlock-user commented on Jun 12, 2019 Member

> Additionally, we are currently planning to change the rule limit from maximum of 30k rules per extension to a global maximum of 150k rules.

> The Declarative Net Request API now allows for the registration and removal of dynamic rules - specified at runtime rather than statically in the manifest. We've also added the capability to remove common tracking headers, such as Referer, Cookie, and Set-Cookie.

> The blocking version of the Web Request API remains available for managed extensions because of the deep integrations that enterprises may have between their software suites and Chrome.

gwarser commented on Jun 14, 2019 • edited Member

Good thread: https://groups.google.com/a/chromium.org/d/msg/chromium-extensions/qFNF3KqNd2E/8R9PWdCbBgAJ

gorhill unpinned this issue on Jun 24, 2019

gwarser commented on Sep 3, 2019 Member

Mozilla's Manifest v3 FAQ

uBlock-user commented on Sep 4, 2019 Member

> how they use the APIs in question to help determine how to best support them.

So they're keeping the door open? That's concerning.

gwarser commented on Sep 23, 2019 Member

[meta] Manifest v3 on bugzilla.

gorhill commented on Sep 23, 2019 • edited Member

Being compatible with manifest v3 does not mean going Google Chrome's way. It makes sense to want to keep Firefox's extensions framework *compatible* with Google Chrome one, so that little to no work is required to make an extension work on both platforms.

For instance, deprecating cross-origin requests from content scripts make a lot of sense. The other item, allow extensions to specify CSP for content script also make sense if cross-origin requests from content scripts is deprecated -- this means extensions will have to explicitly ask permissions to make cross-site requests from their content scripts if they really need such a thing. uBO does not need this, the filter lists are pulled for import/update purpose from the background page.

The other item is replacing background page with service workers -- this could break uBO but I provided feedback about this -- and there is an opened issue for this: https://bugzilla.mozilla.org/show_bug.cgi?id=1580254.

okiehsch commented on Nov 1, 2019 • edited

gwarser commented on Oct 14, 2020 Member

ChrEdge will switch to v3 https://blogs.windows.com/msedgedev/2020/10/14/extension-manifest-chromium-edge/

uBlock-user commented on Oct 14, 2020 Member

MS's statement

> After an extensive review of the concerns raised by content blockers and the community, we believe that a majority of those concerns have been resolved or will be resolved before Web Request API is deprecated. If you continue to face issues, we encourage you to share your feedback, where our team can engage to understand and address your feedback.

gorhill commented on Oct 14, 2020 • edited Member

> Prepare to update your extensions from Manifest v2 to v3 [...] we're disallowing extensions from using remotely hosted code. This change makes extensions more secure.

It's merely store policy, you do not need to wait for manifest v3 for this policy to be enforced. It's has been Firefox's AMO policy since WebExtensions came out 3 years ago -- this has nothing to do with manifest v3.

gwarser commented on Nov 11, 2020 • edited Member

Adblocker Dev Summit 2020

- Status of Manifest v3 in Firefox: https://www.youtube.com/watch?v=tpDFS-GUytg
- Status update on Chrome side: https://www.youtube.com/watch?v=zxWG2yUcTFA

gorhill commented on Nov 12, 2020 Member

I don't really see a divergence issue with keeping a blocking webRequest in Firefox since Google has said it would still be available to Google Chrome Enterprise, the only "divergence" would be who has access to it.

mapx- commented on Dec 10, 2020

Manifest V3 Dynamic Content Scripts

https://docs.google.com/document/d/1nRJ6iW-W1MVSpJnQzNrRQFLMsr0RycwsNym06TD5i18/edit#heading=h.itw7kc7egimi

gorhill commented on Dec 10, 2020 Member

I might give a go at trying to find out all the features which can be fitted into MV3, if only to make the un-portable features stand out more. But this is quite an amount of work and working on current uBO will always be a higher priority.

uBlock-user commented on Dec 10, 2020 Member

> Dynamic content scripts will solve this problem by allowing files scripts registered by the API to be synchronously injected into documents during the specified part of their lifecycle.

gorhill commented on Dec 10, 2020 Member

> Could this cause performance issue/s ?

Why? This is how declarative content scripts are currently injected. Firefox already support dynamically registered content scripts, this is even used in uMatrix.

uBlock-user commented on Dec 10, 2020 • edited Member

Because browser vendors actively discourage running anything synchronously.

mapx- commented on Dec 11, 2020 • edited

https://bugs.chromium.org/p/chromium/issues/detail?id=1054624#c9

> Some of the future features are exciting but there are some omissions:
>
> ## No include_globs and exclude_globs.
>
> These are parts of the existing extension API, and in some cases only globs can provide a solution. If you still want to remove them then please use histograms/metrics to confirm the usage is below the removal threshold.
>
> ## Missing RequestContentScript's

> a) The ability to specify `priority` to reorder the scripts without re-registering the entire list. RequestContentScript used the system of filtered events so that was an inherent feature.
>
> b) The much more flexible URL matching, particularly the simplified RE2 regular expression syntax. The absence of both RE2 syntax and globs in the new API, while simplifying the implementation of the API for Chromium developers, will force some of the extension authors to make their scripts run on more pages than necessary and use JS checks for `location.href` inside. This is somewhat wasteful.
>
> c) RCS runs content scripts **before** document_start when DOM doesn't even contain documentElement. For user experience, this is often much better than document_start because the page is still being downloaded in background by Chrome, no DOM work is being done. Whereas when content scripts run at document_start they compete with the page scripts for DOM rendering time, which delays the early stages of page load. It's not that rare for several extensions to delay the initial render by 100ms or even more. If they could run before document_start (say, at document_create) they could initialize their state while the page is still being downloaded.
>
> These features aren't crucial, admittedly, but they worked for almost the entire lifetime of Chrome and solved real problems. RequestContentScript, for example, would be a real hit and is under-used only because it's marked as experimental in documentation due to a couple of edge case problems.

mapx- commented on Jan 13 • edited

having any ad-blocker installed)

Masatoshi Kimura [:emk] |

(In reply to Kershaw Chang [:kershaw] from comment 2)
Do you have an idea about what we can do here?

Migrating to Manifest v3? (deprecate blocking webRequest and implement declarativeNetRequest)