bugs chromium New issue Open issues Search chromium issues... Sign in

☆ Starred by 6 users

Owner:	asamidoi@chromium.org
CC:	falken@chromium.org tommycli@chromium.org wanderview@chromium.org kinuko@chromium.org rdevl...@chromium.org lazyboy@chromium.org creis@chromium.org ghazale@chromium.org dpa...@chromium.org alex...@chromium.org bashi@chromium.org nasko@chromium.org solomonkinard@chromium.org tjudkins@chromium.org wfh@chromium.org ajgo@chromium.org
Status:	Fixed *(Closed)*
Components:	Internals>Sandbox>SiteIsolation UI>Browser>WebUI Platform>Extensions UI>Browser>Navigation Blink>ServiceWorker
Modified:	Sep 15, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Windows, Chrome, Mac
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
stable
Security_Impact-Stable
Security_Severity-High
allpublic
reward-inprocess
reward-15000
CVE_description-submitted
Target-90

Issue 1199198: Security: UAF caused by some WebUIMessageHandlers when OnJavascriptDisallowed() is not called before destruction

Reported by derce...@gmail.com on Wed, Apr 14, 2021, 10:19 PM EDT

Code

VULNERABILITY DETAILS
The ServiceWorkerContextCore class dispatches various service worker notifications within the browser process. The chrome://serviceworker-internals/ WebUI page then observes some of those notifications.

Due to the way in which the notifications are dispatched, it's possible for an observer associated with a chrome://serviceworker-internals/ page to be called after the page has been destroyed, triggering a use-after-free.

As an extension can open a chrome://serviceworker-internals/ page and trigger service worker notifications, it's possible for an extension to trigger the issue.

VERSION
Chrome Version: Tested on 92.0.4478.0 (latest asan build)
Operating System: Windows 10, version 20H2

REPRODUCTION CASE
1. Install the attached extension.
2. Once installed, the extension will register a service worker, then open manifest.json in a new tab and continually refresh it. Doing this will cause a consistent stream of service worker notifications (since the extension's service worker will be invoked for any content fetched in the tab).
3. The extension will then create two windows, both with 50 chrome://serviceworker-internals/ tabs. The windows will then be closed once all the tabs have loaded. This should trigger the UAF, since it's likely that a service worker notification will be dispatched to one of the destroyed pages.

The extension here tries to trigger the issue reliably, but I have also seen this triggered just by opening and closing a chrome://serviceworker-internals/ tab, so an extension might not need to do too much in practice to trigger this.

CREDIT INFORMATION
Reporter credit: David Erceg

asan_output_872603.txt
17.1 KB View Download

background.js
3.1 KB View Download

manifest.json
169 bytes View Download

service_worker.js
431 bytes View Download

Comment 1 by sheriffbot on Wed, Apr 14, 2021, 10:22 PM EDT Project Member

Labels: external_security_report

Comment 2 by est...@chromium.org on Thu, Apr 15, 2021, 5:32 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Owner: falken@chromium.org
Cc: rdevl...@chromium.org bashi@chromium.org kinuko@chromium.org
Labels: Security_Severity-High stable FoundIn-91 OS-Chrome OS-Linux OS-Mac OS-Windows Pri-1
Components: Blink>ServiceWorker Platform>Extensions

I can't repro this on Linux; it seems to be looping indefinitely. Sounds like it's flaky. I'm going to tentatively triage as High severity because it requires an extension install, though if it's really hard to repro due to flakiness, Medium might be more appropriate. Service Worker owners, PTAL. Also, can you please add a Security_Impact based on where you believe this bug was introduced (Stable vs Beta vs ToT)?

Also adding Devlin for extensions. I'm not sure if extensions are supposed to be able to open chrome:// pages, but if they are, maybe it would make sense to have an opt-in allowlist for which pages they can open? It's hard for me to imagine a use case for an extension needing to open chrome://serviceworker-internals.

Comment 3 by derce...@gmail.com on Thu, Apr 15, 2021, 6:02 PM EDT

If the process is taking too long to run (e.g. because you're testing the extension in an asan build or in a VM), you may need to reduce the number of tabs created in each window (see line 68 of background.js) and increase the time between tab reloads (see line 58 of background.js). Essentially, the extension is trying to maximize both the number of service worker notifications generated and the number of chrome://serviceworker-internals/ tabs. Because this is a timing issue, if you reduce the number of notifications generated or create less chrome://serviceworker-internals/ tabs, the process will complete more quickly, but there's the risk that it will be less reliable.

For reference, the process completes in about 10 seconds for me in a normal release build and in about 3.5 minutes in an asan build. I think it tends to be much quicker and more reliable when run in a normal release build, as opposed to being run under asan or in a VM.

Comment 4 by falken@chromium.org on Thu, Apr 15, 2021, 7:37 PM EDT Project Member

Cc: lazyboy@chromium.org ghazale@chromium.org wanderview@chromium.org asamidoi@chromium.org
Labels: Security_Impact-Stable

Haven't looked at this yet but pre-emptively adding more people.

I'm not sure anything's changed recently here so I'm guessing Impact Stable.

Comment 5 by sheriffbot on Fri, Apr 16, 2021, 12:46 PM EDT Project Member

Labels: M-90 Target-90

Setting milestone and target because of Security_Impact=Stable and high severity.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 6 by falken@chromium.org on Mon, Apr 19, 2021, 4:19 AM EDT Project Member

Owner: asamidoi@chromium.org
Cc: -asamidoi@chromium.org falken@chromium.org

asamidoi@ said she can take a look.

Comment 7 by asamidoi@chromium.org on Mon, Apr 19, 2021, 11:57 PM EDT Project Member

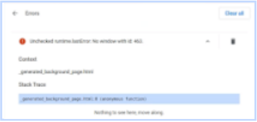
Status: Started (was: Assigned)

Comment 8 by asamidoi@chromium.org on Tue, Apr 20, 2021, 5:16 AM EDT Project Member

Thank you for filing this issue!
I can't reproduce the UAF problem in my environment (92.0.4483.0 (Developer Build) (64-bit) / Linux). I only see the error logs are generated infinitely while enabling the extension for more than 10 mins.
I see the Error, the images attached in this comment, which is saying "unchecked runtime.lastError: No window with id: 463." This error happens without ASAN.

I think it's ok to deprioritize this issue due to the unable/rare reproducibility.

Screen Shot 2021-04-20 at 6.08.40 PM.png
229 KB View Download



Unchecked runtime.lastError: No window with id: 463.
Context
_generated_background_page.html
Stack Trace
_generated_background_page.html:0 (anonymous function)
Nothing to see here, move along.

Screen Shot 2021-04-20 at 6.09.16 PM.png
97.9 KB View Download

Service worker notification UAF 1.0
ID: bhgaffjiimkljglhmjdlbinmomogap

Comment 9 by asamidoi@chromium.org on Thu, Apr 22, 2021, 3:55 AM EDT Project Member

This issue is not reproducible in my environment yet but based on the source code and asan_output_872603.txt in the description, I guess the root cause is a race condition between ServiceWorkerInternalsHandler::RemoveObserverFromStoragePartition() and notifications called from ServiceWorkerContextCore.

We need to call ServiceWorkerInternalsHandler::RemoveObserverFromStoragePartition() and remove a ServiceWorkerInternalsHandler::PartitionObserver from an observer list when chrome://serviceworker-internals/ is deleted.

The code where UAF is triggered is

```
notification.method.Run(observer);
```

and `observer` is already gone.

https://source.chromium.org/chromium/chromium/src/+/master:base/observer_list_threadsafe.h;l=235;drc=44e5a3e4464147c35771e564fb211ac5bfff6a57;

Correct order:
1. Delete RenderFrameHostImpl (= chrome://serviceworker-internals/)
https://source.chromium.org/chromium/chromium/src/+/master:content/browser/webui/web_ui_impl.cc;l=125;drc=44e5a3e4464147c35771e564fb211ac5bfff6a57;bpv=1;bpt=1
2. Remove PartitionObserver from an observer list
https://source.chromium.org/chromium/chromium/src/+/master:content/browser/service_worker/service_worker_internals_ui.cc;l=557;bpv=0;bpt=1
3. Call Notify() but nothing should happen because the observer is already gone.
https://source.chromium.org/chromium/chromium/src/+/master:base/observer_list_threadsafe.h;l=174;drc=44e5a3e4464147c35771e564fb211ac5bfff6a57;bpv=1;bpt=1?q=observer_list_threadsafe.h

Wrong order (UAF case):
The 2 and 3 is reversed so that Notify() is called before the observer is removed.

Comment 10 by asamidoi@chromium.org on Thu, Apr 22, 2021, 6:38 AM EDT Project Member

I created a CL to fix this issue but I can't confirm this CL is actually able to fix this.
https://chromium-review.googlesource.com/c/chromium/src/+/2846223

derceg86@gmail.com:
Can you try to run Chromium in your environment with the CL and see if the issue is still happening? Thank you for your help!

Comment 11 by derce...@gmail.com on Wed, Apr 28, 2021, 12:16 PM EDT

Having spent some time looking into this issue, I believe the reason that it happens is as follows:

The ServiceWorkerInternalsHandler class relies on OnJavascriptDisallowed being called to remove the observers it sets up. That method may be called when a tab is closed, but it may not be.

When a tab is closed, execution can proceed through RenderProcessHostImpl::FastShutdownIfPossible:

https://source.chromium.org/chromium/chromium/src/+/master:content/browser/renderer_host/render_process_host_impl.cc;l=3461;drc=fbdeac6ee19fe7789daaaf2af06cb685c0da1a27

Provided that happens, WebUIImpl::RenderFrameDeleted will be called, which will ultimately result in a call to OnJavascriptDisallowed:

https://source.chromium.org/chromium/chromium/src/+/master:content/browser/webui/web_ui_impl.cc;l=124;drc=aace0cfef2d51e7f038b07d5df26d126e4160f70

In that case, the observers will be removed as the tab is closed and there won't be any issue.

However, if execution doesn't proceed through RenderProcessHostImpl::FastShutdownIfPossible, RenderFrameHostImpl::RenderFrameDeleted will be called, but the web_ui_ instance will have already been destroyed. That means that WebUIImpl::RenderFrameDeleted won't be called and neither will OnJavascriptDisallowed. Which then leads to the UAF being triggered.

There are a few reasons execution may not go through RenderProcessHostImpl::FastShutdownIfPossible. Looking at TabStripModel::CloseWebContentses, it can be seen that RenderProcessHostImpl::FastShutdownIfPossible won't be called if unload handlers need to be run:

https://source.chromium.org/chromium/chromium/src/+/master:chrome/browser/ui/tabs/tab_strip_model.cc;l=1843;drc=5f62968b2788b23a65f4fcb600839010562165b4

While chrome://serviceworker-internals/ doesn't set up any unload handlers itself, that condition is also triggered if the devtools is opened on that tab:

https://source.chromium.org/chromium/chromium/src/+/master:chrome/browser/ui/unload_controller.cc;l=54;drc=16eb8a676d5e584f71f07aa82fbb858be7de423b

Therefore, one trivial way to trigger the issue reliably is to open a chrome://serviceworker-internals/ tab, open the devtools, close the tab, then trigger a service worker notification. This will result in a UAF, since the tab was closed without OnJavascriptDisallowed being called, which means the observers won't have been removed.

RenderProcessHostImpl::FastShutdownIfPossible will also return early if there are other views hosted in the process:

https://source.chromium.org/chromium/chromium/src/+/master:content/browser/renderer_host/render_process_host_impl.cc;l=3465;drc=fbdeac6ee19fe7789daaaf2af06cb685c0da1a27

While WebUI pages typically don't share a renderer, it's fairly easy for an extension to force that situation by creating enough processes that RenderProcessHost::ShouldTryToUseExistingProcessHost returns true:

https://source.chromium.org/chromium/chromium/src/+/master:content/browser/renderer_host/render_process_host_impl.cc;l=4125;drc=fbdeac6ee19fe7789daaaf2af06cb685c0da1a27

On Windows at least, the max renderer process count is 82:

https://source.chromium.org/chromium/chromium/src/+/master:content/browser/renderer_host/render_process_host_impl.cc;l=1434;drc=fbdeac6ee19fe7789daaaf2af06cb685c0da1a27

Therefore, if an extension creates enough frames (hosted by different processes), this limit will be hit and pages will start to share processes.

If two chrome://serviceworker-internals/ tabs share a process and one of the tabs is closed, RenderProcessHostImpl::FastShutdownIfPossible will return early because there's still an active view in the process. Which then means that OnJavascriptDisallowed won't be called. Triggering a service worker notification will then trigger the UAF issue.

Comment 12 by derce...@gmail.com on Wed, Apr 28, 2021, 12:23 PM EDT

I've also looked at some of the other WebUI handlers and it looks like there are at least two more handlers that have the same issue.

The handlers are:

InvalidationsMessageHandler
SignInInternalsHandler

Which are used on the following pages, respectively:

chrome://invalidations/
chrome://signin-internals/

Triggering the issue on those pages is very similar to triggering the issue on chrome://serviceworker-internals/. For example, one way to trigger the issue would be to go through the following steps:

1. Load chrome://invalidations/ in a tab.
2. Open the devtools.
3. Close the tab.
4. Open chrome://invalidations/ again. This will trigger a UAF.

The steps are the same for chrome://signin-internals/.

There's also been at least one other instance where assuming that OnJavascriptDisallowed will always be called has been a problem (see ~~issue 1058769~~).

I can create patches for the two handler classes above, since that's simple enough. But it does seem like this is somewhat of a recurring issue, so perhaps it might be worth trying to prevent it from happening generally.

One simple option might be to update the method comment for OnJavascriptDisallowed to indicate that it's not guaranteed to be called when a tab is closed and shouldn't be relied on to remove observers or free resources before destruction.

Comment 13 by derce...@gmail.com on Wed, Apr 28, 2021, 12:32 PM EDT

Following on from the last two messages, I've created an updated demonstration extension that should reliably trigger the issue. The extension creates enough frames so that the renderer process limit is reached and different chrome://serviceworker-internals/ tabs start being hosted together.

For simplicity, the extension is targeted at Windows, so creates 82 frames, specifically. If you're on another platform and your process limit is higher, you'll need to adjust the value in background.js.

The reproduction steps are:

1. Install the extension.
2. Once installed, the extension will create 82 about:blank tabs (which will be hosted in different processes, at least when the process count is below the limit).
3. The extension will then create two chrome://serviceworker-internals/ tabs.
4. The extension will close one of the chrome://serviceworker-internals/ tabs, then load manifest.json in a tab. Doing that will trigger a service worker notification and cause the UAF to be triggered.

The extension also allows you to test that the same issue affects chrome://invalidations/ and chrome://signin-internals/. You can do that by uncommenting the appropriate sections in background.js.

background.js
5.5 KB View Download

manifest.json
156 bytes View Download

service_worker.js
58 bytes View Download

Comment 14 by rdevl...@chromium.org on Thu, Apr 29, 2021, 5:23 PM EDT Project Member

Thank you for the report, derceg86@, and for the incredibly in-depth analysis in comments 11-13! asamidoi@, does that help with identifying a fix here?

> Also adding Devlin for extensions. I'm not sure if extensions are supposed to be able to open chrome:// pages, but if they are, maybe it would make sense to have an opt-in allowlist for which pages they can open? It's hard for me to imagine a use case for an extension needing to open chrome://serviceworker-internals.

This is WAI. There's a few different use cases of this, including tab/window/session managers, bookmark and history enhancements, etc. A lot of these may need to open up chrome pages. We've generally allowed this as extensions are an extension of the user agent, so we allow them to perform navigations that web pages wouldn't normally be able to (though only through e.g. the tabs or windows API, not just window.open()).

Comment 15 by wanderview@chromium.org on Thu, Apr 29, 2021, 5:25 PM EDT Project Member

FYI, I believe there it is a holiday in TOK this week, so asamidoi may be delayed in responding.

Comment 16 by derce...@gmail.com on Tue, May 4, 2021, 8:58 PM EDT

I've attached a patch for the InvalidationsMessageHandler and SignInInternalsHandler classes. The patch adds a call to OnJavascriptDisallowed in the destructor of both classes, as is also done in PeopleHandler and ProfilePickerHandler.

web_ui_remove_observers.patch
1.4 KB View Download

Comment 17 by derce...@gmail.com on Tue, May 4, 2021, 10:39 PM EDT

Re #c10: Hopefully you can reproduce the issue now, with the information given above, but for what it's worth, it appears that the fix doesn't work, since the .handler_ WeakPtr has already been invalidated when ~PartitionObserver is called. That's because the ServiceWorkerInternalsHandler instance is part-way through destruction at the time of the call. Which then means the observer won't be removed.

A simple option would be to call OnJavascriptDisallowed in the destructor of the ServiceWorkerInternalsHandler class. Though it does still seem like a more general solution or at least convention would be useful, since:

- Some handler classes call OnJavascriptDisallowed in the destructor. Which requires knowing or remembering that OnJavascriptDisallowed won't necessarily be called during destruction.
- Some perform work in both OnJavascriptDisallowed and the destructor, with some code duplicated between the two. Which is prone to breakage if you forget to update one of the locations when adding an observer.
- Some handlers (the ones mentioned in this issue) only implement OnJavascriptDisallowed and don't remove observers at all during destruction. Which leads to the UAFs described above.

Also, because OnJavascriptDisallowed will typically be called when closing a tab, it makes it harder to spot any issues during development.

Comment 18 by Git Watcher on Mon, May 10, 2021, 12:16 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/43dc680a056da697cf31064fa8628b8c557b7010

commit 43dc680a056da697cf31064fa8628b8c557b7010
Author: Asami Doi <asamidoi@chromium.org>
Date: Mon May 10 04:15:30 2021

Remove PartitionObserver from the observer list when it's destructed

This CL fixes the UAF issue of PartitionObserver. This happens when a
chrome://serviceworker-internals/ page is removed and some notifications
are sent to the page before the observer is removed from the
`observer_list_` in ServiceWorkerContextCore. ServiceWorkerContextCore
tries to call Notify() to an already freed observer but the observer is
still registered.

This issue is fixed by calling RemoveObserver() at the destruction of
PartitionObserver.

~~Bug: 1199108~~
Change-Id: Icb921ff05b9fc833155b6b58662d21bd5517d8e9
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2846223
Commit-Queue: Asami Doi <asamidoi@chromium.org>
Reviewed-by: Matt Falkenhagen <falken@chromium.org>
Cr-Commit-Position: refs/heads/master@{#880840}

[modify] https://crrev.com/43dc680a056da697cf31064fa8628b8c557b7010/content/browser/service_worker/service_worker_internals_ui.cc
[modify] https://crrev.com/43dc680a056da697cf31064fa8628b8c557b7010/content/browser/service_worker/service_worker_internals_ui.h

Comment 19 by asamidoi@chromium.org on Mon, May 10, 2021, 1:30 AM EDT Project Member

Status: Fixed (was: Started)

The CL #18 should fix the issue so I'm closing this issue now.
Feel free to reopen if you find this UAF again. Thanks!

Comment 20 by derce...@gmail.com on Mon, May 10, 2021, 1:37 AM EDT

I don't believe the CL in #c18 fixes the issue, for the reason given in #c17. Is there a specific way in which you've tested the patch that shows that it fixes the issue?

Also, there's still the same issue on the other two pages I mentioned - chrome://invalidations/ and chrome://signin-internals/. See #c11, #c12 and #c13. Would you prefer it if I created separate issues for those?

Comment 21 by derce...@gmail.com on Mon, May 10, 2021, 1:40 AM EDT

Also, #c16 contains a proposed patch for InvalidationsMessageHandler and SignInInternalsHandler.

Comment 22 by asamidoi@chromium.org on Mon, May 10, 2021, 3:29 AM EDT Project Member

Status: Started (was: Fixed)

Sorry I totally missed derceg86@'s comments since I saw this page without reloading.
Thank you for proposing the patch!
Do you want to make a CL? I can make a patch in behalf of derceg86@ if you want.

Let me reopen this issue.

Comment 23 by derce...@gmail.com on Tue, May 11, 2021, 12:51 AM EDT

Sure, I can look at creating a CL.

Comment 24 by asamidoi@chromium.org on Tue, May 11, 2021, 2:06 AM EDT Project Member

Thank you so much for your help!

I think the patch #16 covers InvalidationsMessageHandler and SignInInternalsHandler classes. They will be fixed by derceg86@ (thanks again!).
So I'll make a CL for a ServiceWorkerInternalsHandler class.

> Would you prefer it if I created separate issues for those?
I think it's ok to keep track the issue for chrome://invalidations/ and chrome://signin-internals/ in this issue.

Comment 25 by asamidoi@chromium.org on Tue, May 11, 2021, 9:15 AM EDT Project Member

Confirmed the issue still happens with latest Asan build (92.0.4504.0) on Linux thanks to the code at #13.

It's not cleanest solution as commented at #17 but calling OnJavascriptDisallowed in the destructor of ServiceWorkerInternalsHandler fixes the issue and it should be ok. I think fixing the issue is highest priority and we'll refactor them later.

I created a CL that includes comments we found.
https://chromium-review.googlesource.com/c/chromium/src/+/2887386

Comment 26 by falken@chromium.org on Thu, May 13, 2021, 11:43 AM EDT Project Member

Summary: Security: UAF caused by some WebUIMessageHandlers when OnJavascriptDisallowed() is not called before destruction (was: Security: UAF when dispatching service worker notifications)
Cc: creis@chromium.org tommycli@chromium.org
Components: UI>Browser>WebUI

Thanks derceg86@ for the great analysis and fix! I've retitled the crbug to be more clear this is about WebUI. I am not very familiar with the WebUI handler. +creis, tommicli: do you think this is this a bug of WebUiMessageHandler, or is it expected that subclasses should know that OnJavascriptDisallowed() may never be called? It looks like derceg86 has identified 3 subclasses that expected OnJavascriptDisallowed() to be called and depended on it to cleanup state. (It does seem to me they should not have relied on the assumption, for what it's worth.)

Comment 27 by Git Watcher on Thu, May 13, 2021, 11:54 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/a050bc371d8b7937d27620a6ddcbe6a61ae157e5

commit a050bc371d8b7937d27620a6ddcbe6a61ae157e5
Author: Asami Doi <asamidoi@chromium.org>
Date: Fri May 14 03:53:54 2021

Call OnJavascriptDisallowed in the dtor of ServiceWorkerInternalsHandler

OnJavascriptDisallowed() should be called to remove observers from the
observer list when a page is destructed but it may not be called when
1) TabStripModel::ShouldRunUnloadListenerBeforeClosing() returns true
   and FastShutdownIfPossible() is not called.
2) FastShutdownIfPossible() returns early and OnJavascriptDisallowed.

To avoid the above cases, this CL calls OnJavascriptDisallowed()
in the destructor of ServiceWorkerInternalsHandler.

~~Bug: 1199198~~
Change-Id: I2bd8836c292acbbb00cca0789d8fd5c63d29de86
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2887386
Commit-Queue: Asami Doi <asamidoi@chromium.org>
Reviewed-by: Matt Falkenhagen <falken@chromium.org>
Cr-Commit-Position: refs/heads/master@{#882847}

[modify] https://crrev.com/a050bc371d8b7937d27620a6ddcbe6a61ae157e5/content/browser/service_worker/service_worker_internals_ui.cc

Comment 28 by creis@chromium.org on Fri, May 14, 2021, 1:19 AM EDT Project Member

Cc: nasko@chromium.org alex...@chromium.org
Components: UI>Browser>Navigation Internals>Sandbox>SiteIsolation

Adding nasko@ for the WebUiMessageHandler question in comment 26, since I'm less familiar with that.

I am grateful to see the in-depth report, analysis, and proposed fix! I haven't read it all, but one part caught my eye from comment 11:
> While WebUI pages typically don't share a renderer, it's fairly easy for an extension to force that situation by creating enough processes that RenderProcessHost::ShouldTryToUseExistingProcessHost returns true:

I don't think the process limit applies to WebUI pages anymore, since they're now locked to their respective chrome://host sites. That means chrome://foo and chrome://bar shouldn't share a process even when we're over the limit. (I can't find quite where that started, but nasko@ and alexmos@ made some changes related to that in ~~issue 1002276~~ and ~~issue 1071464~~.)

That said, without reading more I'm not sure if that impacts the severity here or not, or if it offers any level of mitigation.

Comment 29 by derce...@gmail.com on Fri, May 14, 2021, 4:08 AM EDT

Regarding the statement about WebUI pages, it probably could be clearer, but it's referring specifically to WebUI pages from the same origin - for example, two chrome://serviceworker-internals/ pages or two chrome://signin-internals/ pages. RenderProcessHostImpl::FastShutdownIfPossible will return early if there's other active views in the renderer process. Having two pages from the same origin hosted in the renderer process fulfills that.

Comment 30 by creis@chromium.org on Fri, May 14, 2021, 12:17 PM EDT Project Member

Comment 29: Ah, thanks! Yes, then the process-sharing-when-over-the-limit description makes sense. (FWIW, we used to make same-site WebUI pages always share a process using process-per-site, which would have made this even easier to attack, but that was removed in r677963 for ~~issue 982266~~. Still, I agree it's not much of a mitigation since it isn't hard to get over the process limit.)

Comment 31 by falken@chromium.org on Mon, May 17, 2021, 10:19 AM EDT Project Member

Labels: Merge-Request-91

Thanks again derceg86 for the analysis and asamidoi for landing the fix for SW. As this affects Stable I think we should start merging the CLs soon to get it in the earliest release we can. It looks like M91 stable cut is... tomorrow. Let me start requesting a merge for the CL that landed, and I'll just go ahead and upload one for the other two Webui handlers.

Comment 32 by sheriffbot on Mon, May 17, 2021, 10:20 AM EDT Project Member

Labels: -Merge-Request-91 Hotlist-Merge-Review Merge-Review-91

This bug requires manual review: We are only 7 days from stable.
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?
5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

Chrome OS Only:
8. Was the change reviewed and approved by the Eng Prod Representative? See Eng Prod ownership by component: http://go/cros-engprodcomponents

Please contact the milestone owner if you have questions.
Owners: benmason@(Android), bindusuvarna@(iOS), marinakz@(ChromeOS), pbommana@(Desktop)

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 33 by adetaylor@google.com on Mon, May 17, 2021, 12:58 PM EDT Project Member

Labels: -Merge-Review-91 Merge-Approved-91

Approving merge to M91, branch 4472, for the fix in #c27.

Comment 34 by falken@chromium.org on Mon, May 17, 2021, 3:59 PM EDT Project Member

- Merge CL for comment 27 under review at https://chromium-review.googlesource.com/c/chromium/src/+/2899866
- CL for the other handlers under review at https://chromium-review.googlesource.com/c/chromium/src/+/2900179

Comment 35 by Git Watcher on Mon, May 17, 2021, 6:35 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd

commit 2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd
Author: Matt Falkenhagen <falken@chromium.org>
Date: Mon May 17 22:34:34 2021

WebUI: Fix dangling observers in two webui handlers.

Original patch from derceg86@gmail.com.

~~Bug: 1199198~~
Change-Id: I90b4ccc3f07b92bbffd0cc4c673ac7cad8650801
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2900179
Reviewed-by: dpapad <dpapad@chromium.org>
Commit-Queue: Matt Falkenhagen <falken@chromium.org>
Cr-Commit-Position: refs/heads/master@{#883674}

[modify] https://crrev.com/2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd/chrome/browser/ui/webui/invalidations/invalidations_message_handler.cc
[modify] https://crrev.com/2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd/chrome/browser/ui/webui/signin_internals_ui.cc

Comment 36 by falken@chromium.org on Mon, May 17, 2021, 6:39 PM EDT Project Member

Status: Fixed (was: Started)
Labels: Merge-Request-91

Marking fixed as the known handlers now have fixes on trunk. Request merge for 91 for c#35 which hopefully won't reset the Merge-Approved-91 label.

Comment 37 by sheriffbot on Mon, May 17, 2021, 6:41 PM EDT Project Member

Labels: -Merge-Request-91 Merge-Review-91

This bug requires manual review: We are only 7 days from stable.
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?
5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

Chrome OS Only:
8. Was the change reviewed and approved by the Eng Prod Representative? See Eng Prod ownership by component: http://go/cros-engprodcomponents

Please contact the milestone owner if you have questions.
Owners: benmason@(Android), bindusuvarna@(iOS), marinakz@(ChromeOS), pbommana@(Desktop)

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot

Comment 38 by Git Watcher on Mon, May 17, 2021, 7:30 PM EDT Project Member

Labels: -merge-approved-91 merge-merged-4472 merge-merged-91

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/9609fc8e7b8ec41e8ac22d20ae97f1e2d2770acc

commit 9609fc8e7b8ec41e8ac22d20ae97f1e2d2770acc
Author: Asami Doi <asamidoi@chromium.org>
Date: Mon May 17 23:29:56 2021

M91: Call OnJavascriptDisallowed in the dtor of ServiceWorkerInternalsHandler

OnJavascriptDisallowed() should be called to remove observers from the
observer list when a page is destructed but it may not be called when
1) TabStripModel::ShouldRunUnloadListenerBeforeClosing() returns true
   and FastShutdownIfPossible() is not called.
2) FastShutdownIfPossible() returns early and OnJavascriptDisallowed.

To avoid the above cases, this CL calls OnJavascriptDisallowed()
in the destructor of ServiceWorkerInternalsHandler.

(cherry picked from commit a050bc371d8b7937d27620a6ddcbe6a61ae157e5)

~~Bug: 1109108~~
Change-Id: I2bd8836c292acbbb00cca0789d8fd5c63d29de86
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2887386
Commit-Queue: Asami Doi <asamidoi@chromium.org>
Reviewed-by: Matt Falkenhagen <falken@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#882847}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2899866
Reviewed-by: Asami Doi <asamidoi@chromium.org>
Commit-Queue: Matt Falkenhagen <falken@chromium.org>
Cr-Commit-Position: refs/branch-heads/4472@{#1123}
Cr-Branched-From: 3d60439cfb36485e76a1c5bb7f513d3721b20da1-refs/heads/master@{#870763}

[modify] https://crrev.com/9609fc8e7b8ec41e8ac22d20ae97f1e2d2770acc/content/browser/service_worker/service_worker_internals_ui.cc

Comment 39 by falken@chromium.org on Mon, May 17, 2021, 7:42 PM EDT Project Member

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines

yes

2. Links to the CLs you are requesting to merge.

https://chromium-review.googlesource.com/c/chromium/src/+/2900179

3. Has the change landed and been verified on ToT?

It landed in ToT and is a simple CL that should fix the UAF.

4. Does this change need to be merged into other active release branches (M-1, M+1)?

No. M90 is also impacted but it's probably too late.

5. Why are these changes required in this milestone after branch?

It's a security bug that was not fixed before the branch.

6. Is this a new feature?

No

7. If it is a new feature, is it behind a flag using finch?

No

Comment 40 by sheriffbot on Tue, May 18, 2021, 12:43 PM EDT Project Member

Labels: reward-topanel

Comment 41 by sheriffbot on Tue, May 18, 2021, 2:02 PM EDT Project Member

Labels: -Restrict-View-SecurityTeam Restrict-View-SecurityNotify

Comment 42 by adetaylor@google.com on Tue, May 18, 2021, 4:18 PM EDT Project Member

Labels: -merge-merged-91 Merge-Approved-91

Approving merge of https://chromium-review.googlesource.com/c/chromium/src/+/2900179 to M91. Please merge to branch 4472.

Initial M91 stable cut is today, so depending on when the merge occurs, this is highly likely to miss the initial stable release. If so that's OK, it will be picked up in the first security refresh.

Comment 43 by adetaylor@google.com on Tue, May 18, 2021, 4:18 PM EDT Project Member

Labels: -Merge-Review-91

Comment 44 by Git Watcher on Wed, May 19, 2021, 3:58 AM EDT Project Member

Labels: -merge-approved-91 merge-merged-91

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/f952a863ff8222037a87ece6a960675ad5eb1373

commit f952a863ff8222037a87ece6a960675ad5eb1373
Author: Matt Falkenhagen <falken@chromium.org>
Date: Wed May 19 07:57:32 2021

M91: WebUI: Fix dangling observers in two webui handlers.

Original patch from derceg86@gmail.com.

(cherry picked from commit 2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd)

~~Bug: 1109108~~
Change-Id: I90b4ccc3f07b92bbffd0cc4c673ac7cad8650801
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2900179
Reviewed-by: dpapad <dpapad@chromium.org>
Commit-Queue: Matt Falkenhagen <falken@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#883674}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2903528
Reviewed-by: Asami Doi <asamidoi@chromium.org>
Bot-Commit: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Cr-Commit-Position: refs/branch-heads/4472@{#1179}
Cr-Branched-From: 3d60439cfb36485e76a1c5bb7f513d3721b20da1-refs/heads/master@{#870763}

[modify] https://crrev.com/f952a863ff8222037a87ece6a960675ad5eb1373/chrome/browser/ui/webui/invalidations/invalidations_message_handler.cc
[modify] https://crrev.com/f952a863ff8222037a87ece6a960675ad5eb1373/chrome/browser/ui/webui/signin_internals_ui.cc

Comment 45 by nasko@chromium.org on Wed, May 19, 2021, 8:23 PM EDT Project Member

Cc: dpa...@chromium.org

To follow up on comment #26, the documentation says that OnJavascriptDisallowed() is not guaranteed to be called and has guidance on how to handle this:

"Because OnJavascriptDisallowed() is not guaranteed to be called before a WebUIMessageHandler's destructor, it is often advisable to use some form of scoped observer that automatically unsubscribes on destruction but can also imperatively unsubscribe in OnJavascriptDisallowed()."

Adding dpapad@ to be aware that an unsafe pattern has crept up in multiple WebUIs, despite documentation.

Comment 46 by falken@chromium.org on Wed, May 19, 2021, 10:20 PM EDT Project Member

Thanks nasko@. Yes dpapad@ pointed me to that documentation in https://chromium-review.googlesource.com/c/chromium/src/+/2900179.

I filed issue 1211190 for follow-up work here.

Comment 47 by dpa...@chromium.org on Thu, May 20, 2021, 3:34 AM EDT Project Member

> Adding dpapad@ to be aware that an unsafe pattern has crept up in multiple WebUIs, despite documentation.

@nasko: I quoted the exact same section in the CL at [1]. Acknowledged that this guidance is not enough to prevent unsafe code from being added to the repo. Open to suggestions. The entire existence of OnJavascriptAllowed/Disallowed is supposedly to make the code safer, and prevent other problems, as explained in the WebUI explainer doc [2].

[1] https://chromium-review.googlesource.com/c/chromium/src/+/2900179/2#message-aea29f0048e3c20569e52502478bea99b57ea241
[2] https://chromium.googlesource.com/chromium/src/+/HEAD/docs/webui_explainer.md#Browser-C_Renderer-JS

Comment 48 by amyressler@chromium.org on Mon, May 24, 2021, 11:06 AM EDT Project Member

Labels: Release-0-M91

Comment 49 by amyressler@google.com on Mon, May 24, 2021, 2:18 PM EDT Project Member

Labels: CVE-2021-30527 CVE_description-missing

Comment 50 by achuith@chromium.org on Thu, May 27, 2021, 3:35 PM EDT Project Member

Labels: LTS-Merge-Request-86 LTS-Security-86

Comment 51 by sheriffbot on Fri, May 28, 2021, 12:21 PM EDT Project Member

Labels: -M-90 M-91 Target-91

Comment 52 by achuith@chromium.org on Tue, Jun 1, 2021, 2:12 PM EDT Project Member

Asami, I'm working on LTS (http://go/lts-merge), and am looking at your CL (https://crrev.com/c/2887386) as a candidate to merge to M86-LTS. There has been a lot of intermediate refactoring. Looks like ServiceWorkerInternalsHandler does not exist in M86:
https://source.chromium.org/chromium/chromium/src/+/refs/tags/86.0.4240.277:content/browser/service_worker/service_worker_internals_ui.cc;l=416;bpv=1

It also looks like the dtor of ServiceWorkerInternalsUI is removing observers with RemoveObserverFromStoragePartition, which is what I think is what the fix is intended to accomplish?

Do you think there's anything additional we need to do for this bug for M-86? Thanks in advance!

Comment 53 by surabhigrover@chromium.org on Tue, Jun 1, 2021, 3:53 PM EDT Project Member

Labels: -LTS-Merge-Request-86 LTS-Merge-Approved-86

Comment 54 by asamidoi@chromium.org on Tue, Jun 1, 2021, 8:42 PM EDT Project Member

> It also looks like the dtor of ServiceWorkerInternalsUI is removing observers with RemoveObserverFromStoragePartition, which is what I think is what the fix is intended to accomplish?

Yes, it looks like it. I think we don't need to merge the fix to M86.

Comment 55 by falken@chromium.org on Wed, Jun 2, 2021, 4:48 AM EDT Project Member

Labels: RegressedIn-90

Hm, I didn't realize that code was so recent. It looks like you're right, the message handler and code relying on OnJavascriptDisallowed() was added in https://chromium-review.googlesource.com/c/chromium/src/+/2697216 which landed in M90.

Comment 56 by achuith@chromium.org on Wed, Jun 2, 2021, 12:29 PM EDT Project Member

Labels: LTS-Security-NotApplicable-86

Comment 57 by amyressler@google.com on Wed, Jun 2, 2021, 3:51 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-15000

*** Boilerplate reminders! ***
Please do NOT publicly disclose details until a fix has been released to all our users. Early public disclosure may cancel the provisional reward. Also, please be considerate about disclosure when the bug affects a core library that may be used by other products. Please do NOT share this information with third parties who are not directly involved in fixing the bug. Doing so may cancel the provisional reward. Please be honest if you have already disclosed anything publicly or to third parties. Lastly, we understand that some of you are not interested in money. We offer the option to donate your reward to an eligible charity. If you prefer this option, let us know and we will also match your donation - subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Please contact security-vrp@chromium.org with any questions.
****

Comment 58 by amyressler@chromium.org on Thu, Jun 3, 2021, 5:52 PM EDT Project Member

And another one! Congratulations, David - the VRP Panel has decided to award you $15,000 for this report. Nice work.

Comment 59 by amyressler@google.com on Fri, Jun 4, 2021, 10:50 AM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 60 by amyressler@google.com on Mon, Jun 7, 2021, 3:27 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 61 by asumaneev@google.com on Tue, Jun 8, 2021, 4:54 AM EDT Project Member

Labels: -LTS-Merge-Approved-86

Comment 62 by asumaneev@google.com on Tue, Jun 8, 2021, 6:43 AM EDT Project Member

Labels: LTS-Security-90 LTS-Merge-Request-90

Comment 63 by gianluca@google.com on Wed, Jun 9, 2021, 9:09 AM EDT Project Member

Labels: -LTS-Merge-Request-90 LTS-Merge-Approved-90

Comment 64 by Git Watcher on Wed, Jun 9, 2021, 12:38 PM EDT Project Member

Labels: merge-merged-4430 merge-merged-90

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/ce050f3c4b04cbaf6718e0028069283d6ffef74c

commit ce050f3c4b04cbaf6718e0028069283d6ffef74c
Author: Matt Falkenhagen <falken@chromium.org>
Date: Wed Jun 09 16:36:58 2021

[M90-LTS] WebUI: Fix dangling observers in two webui handlers.

Original patch from derceg86@gmail.com.

M90 merge conflicts and resolution:
* chrome/browser/ui/webui/signin_internals_ui.cc
SignInInternalsHandler does not exists in M90. Functionality of
OnJavascriptAllowed/Disallowed in SignInInternalsHandler ctor and
dtor is done in SignInInternalsUI ctor and dtor. No additional
changes needed.

(cherry picked from commit 2d48ab7ecc6ccf24b3997321f1e738aa8bfb88dd)

~~Bug: 1199198~~
Change-Id: I90b4ccc3f07b92bbffd0cc4c673ac7cad8650801
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2900179
Reviewed-by: dpapad <dpapad@chromium.org>
Commit-Queue: Matt Falkenhagen <falken@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#883674}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2947087
Commit-Queue: Artem Sumaneev <asumaneev@google.com>
Owners-Override: Artem Sumaneev <asumaneev@google.com>
Reviewed-by: Victor-Gabriel Savu <vsavu@google.com>
Cr-Commit-Position: refs/branch-heads/4430@{#1505}
Cr-Branched-From: e5ce7dc4f7518237b3d9bb93cccca35d25216cbe-refs/heads/master@{#857950}

[modify] https://crrev.com/ce050f3c4b04cbaf6718e0028069283d6ffef74c/chrome/browser/ui/webui/invalidations/invalidations_message_handler.cc

Comment 65 by asumaneev@google.com on Wed, Jun 9, 2021, 12:40 PM EDT Project Member

Labels: -LTS-Merge-Approved-90 LTS-Merged-90

Comment 66 by Git Watcher on Wed, Jun 9, 2021, 12:43 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/bf33e2ad52bfc3b6c2e114457aee65967afce0c9

commit bf33e2ad52bfc3b6c2e114457aee65967afce0c9
Author: Asami Doi <asamidoi@chromium.org>
Date: Wed Jun 09 16:42:08 2021

[M90-LTS] Call OnJavascriptDisallowed in the dtor of ServiceWorkerInternalsHandler

OnJavascriptDisallowed() should be called to remove observers from the
observer list when a page is destructed but it may not be called when
1) TabStripModel::ShouldRunUnloadListenerBeforeClosing() returns true
and FastShutdownIfPossible() is not called.
2) FastShutdownIfPossible() returns early and OnJavascriptDisallowed.

To avoid the above cases, this CL calls OnJavascriptDisallowed()
in the destructor of ServiceWorkerInternalsHandler.

(cherry picked from commit a050bc371d8b7937d27620a6ddcbe6a61ae157e5)

~~Bug: 1199198~~
Change-Id: I2bd8836c292acbbb00cca0789d8fd5c63d29de86
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2887386
Commit-Queue: Asami Doi <asamidoi@chromium.org>
Reviewed-by: Matt Falkenhagen <falken@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#882847}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2945177
Commit-Queue: Artem Sumaneev <asumaneev@google.com>
Owners-Override: Artem Sumaneev <asumaneev@google.com>
Reviewed-by: Victor-Gabriel Savu <vsavu@google.com>
Cr-Commit-Position: refs/branch-heads/4430@{#1506}
Cr-Branched-From: e5ce7dc4f7518237b3d9bb93cccca35d25216cbe-refs/heads/master@{#857950}

[modify] https://crrev.com/bf33e2ad52bfc3b6c2e114457aee65967afce0c9/content/browser/service_worker/service_worker_internals_ui.cc

Comment 67 by sheriffbot on Wed, Sep 15, 2021, 1:31 PM EDT Project Member

Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

For more details visit https://www.chromium.org/issue-tracking/autotriage - Your friendly Sheriffbot