


☆ Starred by 4 users


Owner:

jonr...@chromium.org

CC:



 samans@chromium.org  
adetaylor@chromium.org  
japhet@chromium.org  
boliu@chromium.org  
creis@chromium.org  
kylec...@chromium.org  
jonr...@chromium.org  



 collinbaker@chromium.org  
alex...@chromium.org  
arthu...@chromium.org  
dfried@chromium.org  
wfh@chromium.org  
ajgo@chromium.org

Status:

Fixed (Closed)

Components:

Internals>Sandbox>SiteIsolation  
UI>Browser>Navigation  
Internals>Compositing  
Blink>Loader

Modified:

Jun 10, 2021

Backlog-Rank:

----

Editors:

----

EstimatedDays:

----

NextAction:

----

OS:

Android

Pri:

1

Type:

Bug-Security

Hotlist-Merge-Review  
reward-3000  
Security\_Impact-Stable  
Security\_Severity-Medium  
allpublic  
reward-inprocess  
CVE\_description-submitted  
M-89  
Target-89  
Merge-Rejected-87  
Merge-Rejected-88  
LTS-Security-86  
LTS-Security-Failed-86  
Release-Merge

**Issue 1152894: Security: WebView and Chromium based browser Omnibar Spoofing with Race Condition**

Reported by [susah...@gmail.com](#) on Wed, Nov 25, 2020, 5:01 PM EST

 Code

**VULNERABILITY DETAILS**

It is possible to spoof address bar with secure padlock. It involve race condition in navigation, if "https://www.google.com/csi" (w/ header 204 No Content) loaded first, the address bar will show the spoofed address with attacker controlled content.

I think it's similar to [issue-672644](#) and [issue-407688](#) which involve race condition and unresponsive renderer. I think the PoC html can be improved further to increase the win probability.

Currently I can't reproduce this on official Google Chrome and official chromium browser, because it won't stop the navigation even it won the race.

From my current testing, it is affect browser based on Android WebView and Chromium:

- Microsoft Edge (Android)
- Samsung Internet Beta (Android)
- NAVER Whale (Windows, Android, Linux)
- Firefox Lite (Android)
- Via Browser (Android)
- and more...

**VERSION**

Android System WebView Version: 86.0.4240.198 (Updated on Nov 11, 2020)

**CREDIT INFORMATION**

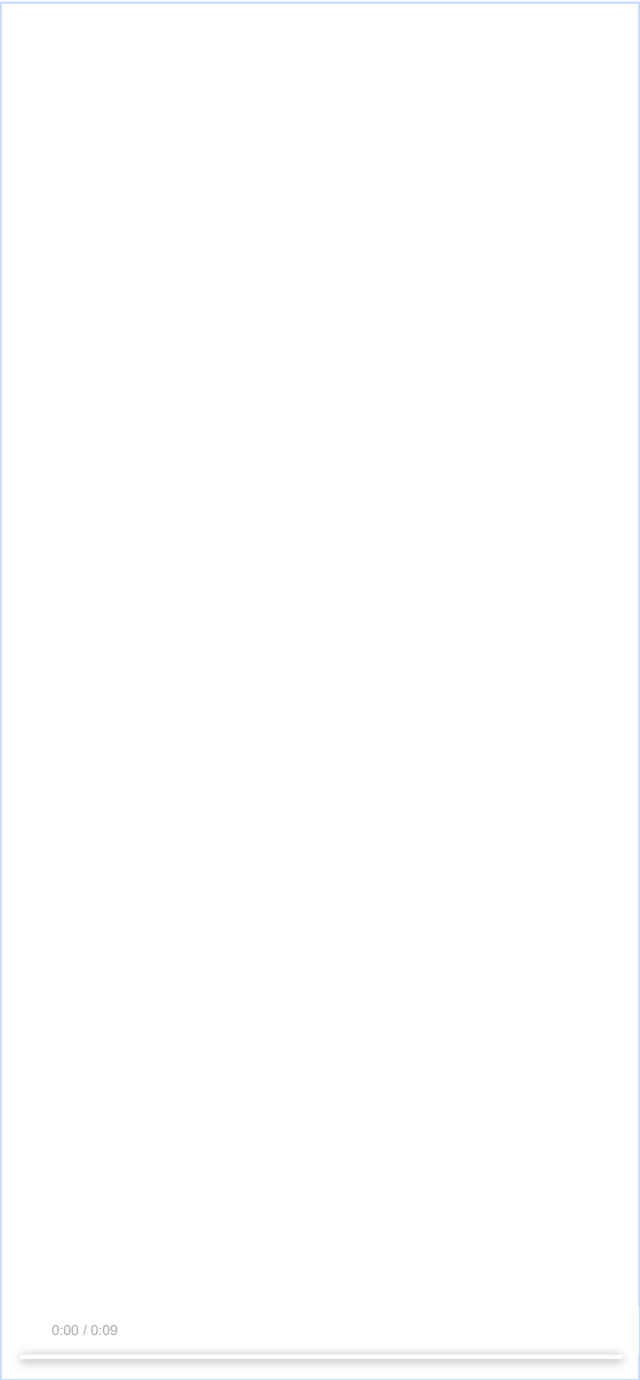
Reporter credit: Irvan Kurniawan (sourc7)

**spoof.1.html**

718 bytes [View](#) [Download](#)

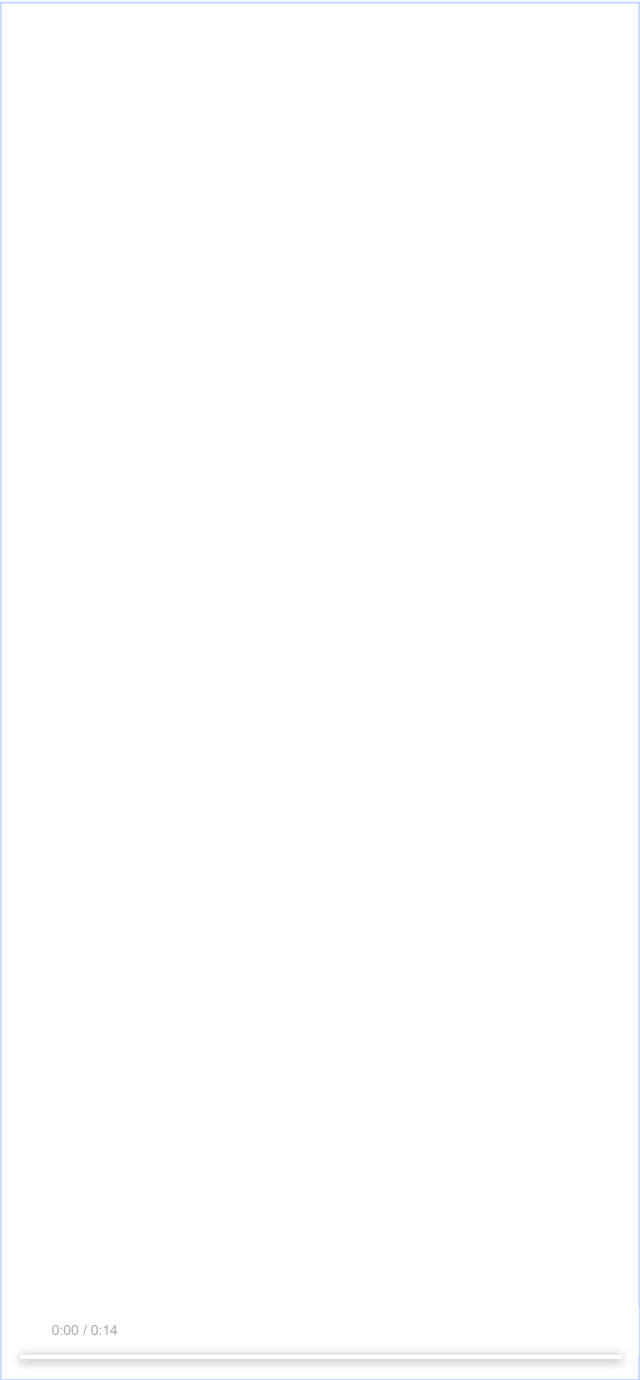
**microsoft-edge-demonstration.mp4**

359 KB [View](#) [Download](#)



0:00 / 0:09

firefox-lite-trying-to-win-race.mp4  
595 KB [View](#) [Download](#)



0:00 / 0:14

firefox-lite-win-race-straight.mp4  
149 KB [View](#) [Download](#)



Comment 1 by [sheriffbot](#) on Wed, Nov 25, 2020, 5:07 PM EST Project Member  
**Labels:** reward-potential

Comment 2 by [cthomp@chromium.org](#) on Wed, Nov 25, 2020, 6:33 PM EST Project Member  
**Status:** Assigned (was: Unconfirmed)  
**Owner:** [creis@chromium.org](#)  
**Labels:** Security\_Severity-Medium Security\_Impact-Stable M-87 OS-Android Pri-1  
**Components:** UI>Browser>Navigation

I can reproduce this on 89.0.4335.0 on Android, although it's a bit inconsistent (about half the time, [Amazon.com](#) ends up loading instead of the "spooft" text). It think it's a bit easier to repro in Incognito. Also interesting is that when changing tabs away and back to the spoofing tab, the top chrome (and the Omnibox) updates significantly faster than the web contents (so the other tab's web contents stick around for a bit, although then the spoof text is lost).

As this allows spoofing the apparent omnibox origin with content controlled by the attacker, but it does not appear to be interactable, I think this is Severity-Medium (it would be High, but lack of interactivity is a strong mitigation) and Impact-Stable. [creis@](#) you've looked into similar bugs before -- could you take a look into this one?

Also of note: the test case hits a DCHECK in ScrollbarLayerImplBase [1], so you need to use a release build without DCHECKs enabled to repro this.

[1] [https://source.chromium.org/chromium/chromium/src/+master.cc/layers/scrollbar\\_layer\\_impl\\_base.cc;l=90;drc=a18be1fcec48ea572192369f55d57bbf8bee3ce9](https://source.chromium.org/chromium/chromium/src/+master.cc/layers/scrollbar_layer_impl_base.cc;l=90;drc=a18be1fcec48ea572192369f55d57bbf8bee3ce9)

Comment 3 by [est...@chromium.org](#) on Tue, Dec 1, 2020, 3:38 PM EST Project Member  
[Issue-1153204](#) has been merged into this issue.

Comment 4 by [est...@chromium.org](#) on Tue, Dec 1, 2020, 3:39 PM EST Project Member  
[Issue-1153204](#) contains a possibly more reliable repro.

Comment 5 by [creis@chromium.org](#) on Wed, Dec 2, 2020, 12:21 AM EST Project Member

I may need to get someone else to help, since I don't have a good Android test environment for this. I can try to offer a few pointers as I dig myself out from OOO, though.

cthomp@: Can you clarify whether you reproduced it in Chrome or another browser? And did the spoof last longer than 4 seconds?

I'm curious why this would only affect other Chromium based browsers. I don't think the defense for this would depend on things outside content/, and I think many of those browsers include the chrome/ layer anyway.

Comment 6 by [susah...@gmail.com](#) on Wed, Dec 2, 2020, 4:47 AM EST

> I may need to get someone else to help, since I don't have a good Android test environment for this.

Today I tried this on WebView Browser Tester 86.0.4240.198 on Android Emulator - Nexus 5 API 30 (w/ Play Store) -> Clear Play Store data -> Update Android System WebView to 86.0.4240.198.

I able to reproduce this (most of the time) on WebView Browser Tester 86.0.4240.198, the address bar and content is spoofed, but the address bar show <https://www.google.com:82> instead of <https://www.amazon.com>

On WebView based browser (Firefox Lite, Via Browser) the address bar show <https://www.amazon.com> (as on video above).

Comment 7 by [susah...@gmail.com](#) on Wed, Dec 2, 2020, 7:25 AM EST

> I'm curious why this would only affect other Chromium based browsers.

After further testing, this is also affect Chrome on Android, I able to reproduce it using more reliable PoC on [issue-1153204](#).

Comment 8 by [creis@chromium.org](#) on Tue, Dec 8, 2020, 2:09 AM EST Project Member

Owner: [jonr...@chromium.org](#)

Cc: [japhet@chromium.org](#) [arthu...@chromium.org](#) [jonr...@chromium.org](#) [collinbaker@chromium.org](#) [alex...@chromium.org](#) [creis@chromium.org](#) [samans@chromium.org](#)

Components: Blink>Loader Internals>Compositing Internals>Sandbox>SiteIsolation UI>Browser>TabStrip

Thanks-- the repro from [issue-1153204](#) does help. I've discovered several issues, and I'm CC'ing some folks who might be familiar with them.

There's a simpler version posted at <http://csreis.github.io/tests/interrupt-popup.html>, without direct explanation describing what behavior it causes. Repro steps:

0) Start Chrome without Site Isolation (e.g., on Android or on desktop with the flags mentioned below).

1) Click "Open Window."

2) Switch back to opener window and click "Add Content."

3) Click "Navigate Window" from opener window, to navigate (and commit) to [example.com](#) and interrupt loading with a 204 HTTP response.

If the popup was occluded during step 3, the stale paint will remain indefinitely. If it was pulled out into a window of its own and was visible at the time, the stale paint goes away after 4 seconds, as we would normally expect. (Note: I have mixed luck repro'ing this on Android, but I've seen it happen there. It's quite consistent on desktop without Site Isolation.)

Some of the issues involved:

1) The issue occurs when Site Isolation is disabled, not just on Android.

You can repro it in a non-official desktop build with `--disable-site-isolation-trials` (but be careful on official builds, which get affected by enterprise policy and may force Site Isolation enabled).

I thought it might have worked even with Site Isolation if the victim and the 204 URL were same-site and shared a process, but that doesn't appear to be the case. It appears the attacker and victim might need to share a process, though I haven't confirmed that yet. (If so, that would be an additional mitigating factor.)

2) For some reason, the same-process case allows the 204 URL to prevent the victim page from painting even though the victim page has committed.

There's no ongoing\_navigation\_request in Navigator::OnBeginNavigation, so there must be something else that prevents the committed page from making further progress. Perhaps that's something the Loading team would know about? It would be great to let the committed page continue loading despite the 204 URL's "failure" if we can. CC'ing [japhet@](#) and [arthuronzogni@](#) in case they have any ideas, but others should feel free to chime in.

3) When the victim page commits, we start the `RenderWidgetHostImpl::new_content_rendering_timeout_paint` timer so that the paint will be cleared after 4 seconds if nothing paints from the new page. However, even though we "do" get to `ForceFirstFrameAfterNavigationTimeout` and call `ClearDisplayedGraphics`, we "don't" actually clear the graphics (in the case that the victim tab was not visible). This appears to be due to [r598124](#) in [issue-878373](#) from [samans@](#) (who has left the team). That CL changed `DelegatedFrameHostAndroid::ResetFallbackToFirstNavigationSurface()` to return early here, breaking the security invariant of `ClearDisplayedGraphics` and causing us to leave an attacker-controlled paint visible in the tab. We need to fix this. [jonross@](#): Can you help point us to someone who can help with this?

4) When the 204 URL commits with no content (prior to the paint timer issue above), we get to `OnRequestFailedInternal` and notify `WebContentsObservers` about `DidFinishNavigation`. At least on desktop, this is triggering `ThumbnailReadinessTracker::DidFinishNavigation` and `ScopedThumbnailCapture::ScopedThumbnailCapture`, which calls `UpdateVisibilityAndNotifyPageAndView` and eventually `ForceFirstFrameAfterNavigationTimeout()`. It appears we end up taking a tab thumbnail capture of the old attacker-displayed content to use for the newly committed URL in the tab strip. I suspect we do not ever want to use a paint from the previous commit, but maybe solving the paint timer issue above will make this go away by clearing the stale paint before tab capture. CC'ing [collinbaker@](#) just in case. (Does this code apply on Android? I don't have an Android build set up to see how we get to `ForceFirstFrameAfterNavigationTimeout()` on that platform; maybe [alexmos@](#) can help with that part of the repro.)

I suspect the paint timer fix will be the quickest disruption to this bug, so I'll assign to [jonross@](#) to help find an owner for that part. We can continue looking at the other issues in parallel. Thanks!

Comment 9 by [collinbaker@chromium.org](#) on Tue, Dec 8, 2020, 3:26 PM EST Project Member

re (4): this code isn't used on Android, just on desktop. Either way, I think displaying a thumbnail from a previous commit or from a failed commit is a bug.

I'm not sure how to handle this though. Upon a 204/205, should the old thumbnail be discarded and a new one captured? Even though in the normal case, this will be the same page?

Forgive me if the answer is obvious, I don't understand this exploit.

Comment 10 by [jonr...@chromium.org](#) on Tue, Dec 8, 2020, 5:36 PM EST Project Member

Can I be added to [issue-1153204](#) so that I can see the additional context / repro steps?

Comment 11 by [creis@chromium.org](#) on Tue, Dec 8, 2020, 8:41 PM EST Project Member

Comment 10: Sure, just CC'd you. Sorry for the delay there.

Comment 9: In general, it seems like we shouldn't create a thumbnail of a tab after a `DidFinishNavigation` unless the last committed URL has produced a paint. In this exploit, an about:blank tab is created, content is injected into it, URL A commits, then URL B returns an HTTP 204 response (no content).

If URL A had painted, it would be fine to create a thumbnail of it after its own `DidFinishNavigation`. Similarly, I wouldn't be concerned if we updated the thumbnail of URL A at later points in time, such as when URL B "aborts" via a 204 response (since URL B never commits a new document). However, we never get a paint from URL A's document at all in this bug, so it seems wrong to create a thumbnail after either URL A's `DidFinishNavigation` or URL B's `DidFinishNavigation`. Here, the thumbnail includes the stale injected content from the about:blank document before URL A, which is wrong.

I suspect that fixing `ForceFirstFrameAfterNavigationTimeout` to actually clear the stale paint (while creating a thumbnail) would make this less important, since the thumbnail would be blank in that case rather than a stale paint. Still, I agree with you that it may be safer if you can check whether there has been a paint since the URL committed, before trying to grab a thumbnail.

Comment 12 by [jonr...@chromium.org](#) on Wed, Dec 9, 2020, 5:35 PM EST Project Member

Cc: [kylec...@chromium.org](#)

+kylechar@ another Viz owner whom I'll need for reviews.

I'm taking a look at the Viz-side changes that are needed.

A few things I've noted while debugging this:

- 1) For Desktop, BrowserNavigator::Nagivate is not being triggered, which would normally lead to advancing the Surfaces. Though RenderWidgetHostViewAura::OnDidNavigateMainFrameToNewPage is notified when we begin these navigations.
- 2) Android is a bit trickier. While it is similarly notified of the navigation, it contains a mix of hidden-navigation edge cases and optimizations. With the goal of allocating the new surface once we start to become visible.
- 3) Our Surface Eviction paths don't like being invoked when not visible
- 4) SurfaceLayer with either invalid surfaces, or not-yet-produced surfaces, seems to be insufficient for invalidating the thumbnail imagery.

I'm not familiar with the thumbnail taking, but will take a look to see how we can cancel when there is no valid surface.

[Comment 13](#) by [jonr...@chromium.org](#) on Thu, Dec 10, 2020, 6:04 PM EST Project Member

Cc: [dfried@chromium.org](mailto:dfried@chromium.org)

+[dfried@](mailto:dfried@chromium.org) an OWNER of [chrome/browser/ui/thumbnails](#)

Hi,  
While looking into this issue I've ran into an edge case with ThumbnailReadinessTracker.

I'm dealing with a navigation that never completed the document load. However when ThumbnailReadinessTracker::DidStartNavigation is called it correctly identified that NavigationShouldInvalidateThumbnail.

However I am not familiar enough to know how/if we clear ThumbnailTabHelper::thumbnail\_.

In this particular bug I'm looking at fixing how RenderWidgetHostImpl handles these interrupted navigations, by evicting the old/pre-navigation surfaces. This allows us to not present the old content on a tab change.

A side effect of this would be that the copy requests never return until a new valid surface is presented. So ThumbnailTabHelper::StoreThumbnail wouldn't be called. In the interim mousing over the tab seems to show the previous thumbnail image.

I'd appreciate any insight here on how we could clear the thumbnail in this case.

[Comment 14](#) by [collinbaker@chromium.org](#) on Thu, Dec 10, 2020, 6:40 PM EST Project Member

Huh, looks like the thumbnail doesn't get cleared. I think the best place to do this would be [https://source.chromium.org/chromium/chromium/src/+master:chrome/browser/ui/thumbnails/thumbnail\\_tab\\_helper.cc;l=186;drc=0348a1421914f3609d287cb5645a856f5eb2542c](https://source.chromium.org/chromium/chromium/src/+master:chrome/browser/ui/thumbnails/thumbnail_tab_helper.cc;l=186;drc=0348a1421914f3609d287cb5645a856f5eb2542c) when transitioning to PageReadiness::kNotReady

I think ThumbnailImage should have an additional method ClearThumbnail(). Seems the only way to do it currently would be to call AssignSkBitmap with a blank or empty bitmap. The implementation should be as simple as 'data\_>data.clear()'.

You're welcome to make this change if you're already working in that area. If not I can do it as well.

[Comment 15](#) by [jonr...@chromium.org](#) on Thu, Dec 10, 2020, 7:45 PM EST Project Member

I can make it a part of the change. Thanks for advice, that does allow for us to clear the thumbnail. Then the evicted surface prevents getting a new one while we don't have content to display.

[Comment 16](#) by [jonr...@chromium.org](#) on Fri, Dec 11, 2020, 1:50 PM EST Project Member

Potential CL is in flight: <https://chromium-review.googlesource.com/c/chromium/src/+2585790>

[Comment 17](#) by [jonr...@chromium.org](#) on Mon, Dec 14, 2020, 2:14 PM EST Project Member

Cc: [khush...@chromium.org](mailto:khush...@chromium.org)

+[khushalsagar@](mailto:khushalsagar@chromium.org) an owner of [ui/android](#) for context on this as it pertains to the review.

[Comment 18](#) by [jonr...@chromium.org](#) on Tue, Dec 15, 2020, 3:46 PM EST Project Member

Cc: [boliu@chromium.org](mailto:boliu@chromium.org)

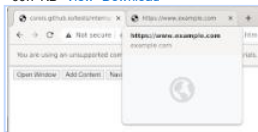
+[boliu@](mailto:boliu@chromium.org) an owner of [ui/android](#) for context on this as it pertains to the review.

[Comment 19](#) by [jonr...@chromium.org](#) on Wed, Dec 16, 2020, 10:48 AM EST Project Member

The behaviour introduced in the code review leaves a grey region where the thumbnail normally displays. With a globe favicon in the middle of the region. (This seems to be the default favicon as the page that fails to load does not have its own custom one.)

See attached image of the new behaviour

**cleared\_data\_thumbnail.png**  
60.7 KB [View](#) [Download](#)



[Comment 20](#) by [jonr...@chromium.org](#) on Wed, Dec 16, 2020, 10:53 AM EST Project Member

Regarding the new failing test:

`browser_tests DeferAllScriptBrowserTest.DeferAllScriptRestoredPreviewWithBackForwardCache`

This test attempts to take a screenshot during teardown:

```
#6 0x7f3eaa5534 content::DelegatedFrameHost::CopyFromCompositingSurfaceInternal()
#7 0x7f3eaa52f3 content::DelegatedFrameHost::CopyFromCompositingSurface()
#8 0x7f3ea2e0986 content::RenderWidgetHostViewBase::CopyMainAndPopupFromSurface()
#9 0x7f3ea2d2b31 content::RenderWidgetHostViewAura::CopyFromSurface()
#10 0x55a296959ca1 ThumbnailTabHelper::CaptureThumbnailOnTabHidden()
#11 0x55a29695c22e ThumbnailTabHelper::TabStateTracker::OnVisibilityChanged()
#12 0x7f3ea7862b6 content::WebContentsImpl::SetVisibilityAndNotifyObservers():$.40::operator()()
#13 0x7f3ea754200 content::WebContentsImpl::WebContentsObserverList::ForEachObserver<>()
#14 0x7f3ea747497 content::WebContentsImpl::SetVisibilityAndNotifyObservers()
#15 0x7f3ea73933a content::WebContentsImpl::UpdateVisibilityAndNotifyPageAndView()
#16 0x7f3ea780a1a content::WebContentsImpl::UpdateWebContentsVisibility()
#17 0x7f3ea7c30eb content::WebContentsViewAura::UpdateWebContentsVisibility()
#18 0x7f3ea7c48a8 content::WebContentsViewAura::OnWindowOcclusionChanged()
#19 0x7f3deb4db9 aura::Window::SetOcclusionInfo()
```

```

#20 0x7ff3debc96ce aura::DefaultWindowOcclusionChangeBuilder::~DefaultWindowOcclusionChangeBuilder()
#21 0x7ff3debc9739 aura::DefaultWindowOcclusionChangeBuilder::~DefaultWindowOcclusionChangeBuilder()
#22 0x7ff3debd8cbc std::_Cr::default_delete<::operator()()
#23 0x7ff3debd8c4a std::_Cr::unique_ptr<::reset()
#24 0x7ff3debd3b19 std::_Cr::unique_ptr<::unique_ptr()
#25 0x7ff3debcf18f aura::WindowOcclusionTracker::MaybeComputeOcclusion()
#26 0x7ff3debd0f25 aura::WindowOcclusionTracker::Unpause()
#27 0x7ff3deb97131 aura::Env::UnpauseWindowOcclusionTracking()
#28 0x7ff3debc429 aura::WindowOcclusionTracker::ScopedPause::~ScopedPause()
#29 0x7ff3deba908e aura::Window::SetVisible()
#30 0x7ff3deba90b7 aura::Window::Hide()
#31 0x7ff3efb80520 views::DesktopNativeWidgetAura::Hide()
#32 0x7ff3efb1e2f1 views::Widget::Hide()
#33 0x55a296cd93a BrowserView::OnWindowCloseRequested()
#34 0x7ff3efb3f5b0 views::NonClientView::OnWindowCloseRequested()
#35 0x7ff3efb1da14 views::Widget::CloseWithReason()
#36 0x7ff3efb1dcb7 views::Widget::Close()
#37 0x55a296cd551c BrowserView::Close()
#38 0x55a2932d109c BrowserCloseManager::CloseBrowsers()
#39 0x55a2932d1466 BrowserCloseManager::CheckForDownloadsInProgress()
#40 0x55a2932d1312 BrowserCloseManager::TryToCloseBrowsers()
#41 0x55a2932d0f27 BrowserCloseManager::StartClosingBrowsers()
#42 0x55a292babbf2 chrome::CloseAllBrowsers()
#43 0x55a2929dcae9 BrowserProcessPlatformPartBase::AttemptExit()
#44 0x55a292babb42 chrome::AttemptExitInternal()
#45 0x55a292babf25 chrome::AttemptExit()
#46 0x55a28cf43ae7 base::internal::FunctorTraits<::Invoke<>()
#47 0x55a28cf43abd base::internal::InvokeHelper<::MakeItSo<>()
#48 0x55a28cf43a81
_ZN4base8internal7InvokerINS0_9BindStateIPFvEJEEES3_E7RunImplIS4_NS14__Cr5tupleJEEEEJEEEvOT_OT0_NS8_16integer_sequencelmJXspT1_EEEEE
#49 0x55a28cf43a4c base::internal::Invoker<::RunOnce()
#50 0x7ff3f76275f1_ZNO4base12OnceCallbackIFvVvEE3RunEv
#51 0x7ff3f77ef182 base::TaskAnnotator::RunTask()
#52 0x7ff3f783515a base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl()
#53 0x7ff3f7834925 base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
#54 0x7ff3f78353b9 base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
#55 0x7ff3f76e1da1 base::MessagePumpGlib::HandleDispatch()
#56 0x7ff3f76e24f1 base::(anonymous namespace)::WorkSourceDispatch()
#57 0x7ff3be6ffb9b g_main_context_dispatch
#58 0x7ff3be6ffe48 (/usr/lib/x86_64-linux-gnu/libglib-2.0.so.0.6600.1+0x51e47)
#59 0x7ff3be6ffeff g_main_context_iteration
#60 0x7ff3f76e1eb0 base::MessagePumpGlib::Run()
#61 0x7ff3f78359e0 base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run()
#62 0x7ff3f7780f55 base::RunLoop::Run()
#63 0x55a2928ec95a InProcessBrowserTest::RunUntilBrowserProcessQuits()
#64 0x55a2928eddbc InProcessBrowserTest::QuitBrowsers()
#65 0x55a2928edc42 InProcessBrowserTest::PostRunTestOnMainThread()
#66 0x55a293abadbf content::BrowserTestBase::ProxyRunTestOnMainThreadLoop()
.... test launching part of stack....

```

Previously this would generate a viz::CopyOutputRequest, and forward the request to the GPU process. However before the changes in content/ there was still a valid Surface to perform the request on.

It appears that with this test a navigation has begun, but not completed, before the test attempts to tear down the browser.  
[delegated\_frame\_host.cc(467)] OnNavigateToNewPage LocalSurfaceId(7, 1, F8F0...)  
[delegated\_frame\_host.cc(154)] JR CopyFromCompositingSurfaceInternal current LocalSurfaceId(0, 0, 0000...) pre-navigation LocalSurfaceId(7, 1, F8F0...)

With there being no valid surface we are canceling the request in DelegatedFrameHost::CopyFromCompositingSurfaceInternal.

viz::CopyOutputRequests always post-tasks its' callback, and it seems to be doing this on a worker thread instead of the UI thread in this particular case. Whereas the non-shutdown cases seen in resolving the bug occur on the UI thread.

```

[FATAL:thumbnail_tab_helper.cc(280)] Check failed: ::content::BrowserThread::CurrentlyOn(content::BrowserThread::UI). Must be called on Chrome_UIThread; actually called on ThreadPoolForegroundWorker.
#0 0x7f6b8deea95f base::debug::CollectStackTrace()
#1 0x7f6b8dc725ea base::debug::StackTrace::StackTrace()
#2 0x7f6b8dc725a5 base::debug::StackTrace::StackTrace()
#3 0x7f6b8dcbe1f9 logging::LogMessage::~LogMessage()
#4 0x7f6b8dcbe939 logging::LogMessage::~LogMessage()
#5 0x7f6b8dc31e5b logging::CheckError::~CheckError()
#6 0x5635a2af91cd ThumbnailTabHelper::StoreThumbnail()
#7 0x5635a2af90ef ThumbnailTabHelper::StoreThumbnailForTabSwitch()
#8 0x5635a2afc1cf base::internal::FunctorTraits<::Invoke<>()
#9 0x5635a2afc061 base::internal::InvokeHelper<::MakeItSo<>()
#10 0x5635a2afbaf8
_ZN4base8internal7InvokerINS0_9BindStateI18ThumbnailTabHelperFvNS_9TimeTicksERK8SkBitmapEJNS_7WeakPtrIS3_EES4_EEEFvS7_EE7RunImplIS9_NS14__Cr5tupleJUSB_S4_EEEJLm0ELm1EEEEvOT_OT0_NSG_16integer_sequencelmJXspT1_EEEEEES7_
#11 0x5635a2afb2f1 base::internal::Invoker<::RunOnce()
#12 0x7f6b7f483af6_ZNO4base12OnceCallbackIFvRK8SkBitmapEE3RunES3_
#13 0x7f6b810a7c82 content::DelegatedFrameHost::CopyFromCompositingSurface():$ _0: operator()()
#14 0x7f6b810a7c1c base::internal::FunctorTraits<::Invoke<>()
#15 0x7f6b810a7bb7 base::internal::InvokeHelper<::MakeItSo<>()
#16 0x7f6b810a7b57
_ZN4base8internal7InvokerINS0_9BindStateI27content18DelegatedFrameHost26CopyFromCompositingSurfaceERKN3gfx4RectERKNS5_4SizeENS_12OnceCallbackIFvRK8SkBitmapEEEE3S_0USH_EEEFvNS14__Cr10unique_ptrIN3viz16CopyOutputResultENSK_14default_deletelSN_EEEEE7RunImplIS1_NSK_5tupleJUSH_EEEJLm0EEEvOT_OT0_NSK_16integer_sequencelmJXspT1_EEEEOSQ_
#17 0x7f6b810a7af1 base::internal::Invoker<::RunOnce()
#18 0x7f6b7b5e9d66_ZNO4base12OnceCallbackIFvNS14__Cr10unique_ptrIN3viz16CopyOutputResultENS1_14default_deletelS4_EEEEE7RunES7_
#19 0x7f6b7b5e9cd6 base::internal::FunctorTraits<::Invoke<>()
#20 0x7f6b7b5e9bc2 base::internal::InvokeHelper<::MakeItSo<>()
#21 0x7f6b7b5e9b72
_ZN4base8internal7InvokerINS0_9BindStateINS_12OnceCallbackIFvNS14__Cr10unique_ptrIN3viz16CopyOutputResultENS4_14default_deletelS7_EEEEEEEJSA_EEEFvVEE7RunImplISCS_NS4_5tupleJSA_EEEJLm0EEEEvOT_OT0_NS4_16integer_sequencelmJXspT1_EEEEE
#22 0x7f6b7b5e9b1c base::internal::Invoker<::RunOnce()
#23 0x7f6b8dc285f1_ZNO4base12OnceCallbackIFvVEE3RunEv
#24 0x7f6b8ddff0182 base::TaskAnnotator::RunTask()
#25 0x7f6b8de66608 base::internal::TaskTracker::RunSkipOnShutdown()
#26 0x7f6b8de66250 base::internal::TaskTracker::RunTaskWithShutdownBehavior()
#27 0x7f6b8de65d61 base::internal::TaskTracker::RunTask()
#28 0x7f6b8df20a95 base::internal::TaskTrackerPosix::RunTask()
#29 0x7f6b8de6524f base::internal::TaskTracker::RunAndPopNextTask()
#30 0x7f6b8de7fec2 base::internal::WorkerThread::RunWorker()

```

Comment 21 by [bugdroid](#) on Wed, Dec 16, 2020, 5:33 PM EST Project Member

The following revision refers to this bug:

<https://chromium.googlesource.com/chromium/src/+865a12d72b770520f88775d5e37810ead62d82c8>

commit [865a12d72b770520f88775d5e37810ead62d82c8](#)

Author: Jonathan Ross <[jonross@chromium.org](mailto:jonross@chromium.org)>

Date: Wed Dec 16 22:33:00 2020

Evict Surfaces from Before Navigation When Re-using DelegatedFrameHost

When we are not applying site-isolation a DelegatedFrameHost can be re-used when navigating between different pages. This can occur on Android as well as with the un-supported flag --disable-site-isolation-trials.

When RenderWidgetHostImpl::ForceFirstFrameAfterNavigationTimeout is invoked, as either from tab-changing, or from thumbnailing, we set fallback surfaces. This is intended to be the first viz::SurfaceId that was generated by navigation.

However if a navigation fails we don't actually embed a new surface. The DelegatedFrameHost then ends up utilizing outdated surfaces as the fallback.

This change updates DelegatedFrameHost to be notified of when a navigation begins. If we fail to embed a new surface by the timeout, we evict the surface that predates the navigation.

The ThumbnailTabHelper has been updated to also clear its currently cached thumbnail in the case that the page has transitioned back to the unready state. Thus clearing thumbnails that exist from before a navigation.

TEST=NoCompositingRenderWidgetHostViewBrowserTest.

NoFallbackAfterHiddenNavigationFails

[Bug-1162804](#)

Change-Id: [Ia5734abb5201a56c271114a891bc81212a3aa975](#)

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+2585790>

Reviewed-by: Dana Fried <[dfried@chromium.org](mailto:dfried@chromium.org)>

Reviewed-by: Jonathan Ross <[jonross@chromium.org](mailto:jonross@chromium.org)>

Reviewed-by: kylechar <[kylechar@chromium.org](mailto:kylechar@chromium.org)>

Reviewed-by: Collin Baker <[collinbaker@chromium.org](mailto:collinbaker@chromium.org)>

Reviewed-by: Bo <[bolu@chromium.org](mailto:bolu@chromium.org)>

Commit-Queue: Jonathan Ross <[jonross@chromium.org](mailto:jonross@chromium.org)>

Cr-Commit-Position: refs/heads/master@{#837779}

[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail\\_tab\\_helper.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail_tab_helper.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail\\_image.h](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail_image.h)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/delegated\\_frame\\_host.h](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/delegated_frame_host.h)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail\\_tab\\_helper.h](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail_tab_helper.h)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/delegated\\_frame\\_host.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/delegated_frame_host.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/ui/android/delegated\\_frame\\_host\\_android.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/ui/android/delegated_frame_host_android.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/render\\_widget\\_host\\_view\\_base.h](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/render_widget_host_view_base.h)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/ui/android/delegated\\_frame\\_host\\_android.h](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/ui/android/delegated_frame_host_android.h)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/render\\_widget\\_host\\_view\\_android.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/render_widget_host_view_android.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/render\\_widget\\_host\\_view\\_browser\\_test.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/render_widget_host_view_browser_test.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail\\_image.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/chrome/browser/ui/thumbnails/thumbnail_image.cc)  
[modify] [https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer\\_host/render\\_widget\\_host\\_view\\_aura.cc](https://crrev.com/865a12d72b770520f88775d5e37810ead62d82c8/content/browser/renderer_host/render_widget_host_view_aura.cc)

Comment 22 by [creis@chromium.org](mailto:creis@chromium.org) on Wed, Dec 16, 2020, 6:44 PM EST Project Member

Thanks jonross@! Does [r837779](#) take care of points (3) and (4) from [comment 8](#)? That seems like it would take care of most of the security impact, allowing us to look into (2) in a separate issue (i.e., why same-process 204 interrupts loading but cross-process 204 doesn't).

Comment 23 by [jonr...@chromium.org](mailto:jonr...@chromium.org) on Thu, Dec 17, 2020, 9:29 AM EST Project Member

Owner: [collinbaker@chromium.org](mailto:collinbaker@chromium.org)

Correct, [r837779](#) addresses points (3) and (4) from [comment 8](#). This address the behaviours seen on Android, as well as desktop with --disable-site-isolation-trials.

In the review [dfried@](mailto:dfried@) and [collinbaker@](mailto:collinbaker@) identified some follow-up work for the thumbnail portion of the fix. However as the previous thumbnail is cleared by the patch this should be sufficient to alleviate the security concerns.

You are also correct that point (2) can be investigated as a separate issue.

I'm going to assign this to [collinbaker@](mailto:collinbaker@) for the thumbnail follow-up.

Comment 24 by [adetaylor@google.com](mailto:adetaylor@google.com) on Mon, Dec 21, 2020, 10:52 AM EST Project Member

[jonross@](mailto:jonross@) would you mind filing a separate crbug for the follow up work, then marking this crbug as Fixed? That will cause Sheriffbot to initiate the merge process for getting this security bug merged back to the appropriate branches. (As an externally reported medium severity bug, I think sheriffbot will choose to merge this to beta, but only after the VRP panel have confirmed the severity).

Comment 25 by [danakj@chromium.org](mailto:danakj@chromium.org) on Tue, Dec 22, 2020, 11:08 AM EST Project Member

I think the CL in [#21](#) is causing crashes: <https://bugs.chromium.org/p/chromium/issues/detail?id=1160146#c10>

[collin](mailto:collin) would you be able to have a look as [jonross](mailto:jonross) is away?

Comment 26 by [sheriffbot](#) on Fri, Dec 25, 2020, 12:21 PM EST Project Member

[collinbaker](mailto:collinbaker): Uh oh! This issue still open and hasn't been updated in the last 14 days. This is a serious vulnerability, and we want to ensure that there's progress. Could you please leave an update with the current status and any potential blockers?

If you're not the right owner for this issue, could you please remove yourself as soon as possible or help us find the right one?

If the issue is fixed or you can't reproduce it, please close the bug. If you've started working on a fix, please set the status to Started.

Thanks for your time! To disable nags, add the Disable-Nags label.

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

Comment 27 by [sheriffbot](#) on Mon, Jan 4, 2021, 11:15 AM EST Project Member



This issue hasn't been updated in the last 30 days - please update it or consider lowering its priority.

Thanks for your time! To disable nags, add the Disable-Nags label.

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

**Comment 28** by [jonr...@chromium.org](#) on Tue, Jan 5, 2021, 10:27 AM EST Project Member  
**Blocking:** 1163121

**Comment 29** by [jonr...@chromium.org](#) on Tue, Jan 5, 2021, 10:33 AM EST Project Member  
**Owner:** jonr...@chromium.org  
**Blockedon:** 1160146

~~Issue 1163121~~ filed to track the follow-up work so that merging work for [r837779](#) can be tracked here. Per [comment 35](#)

Per [comment 25](#) it's suspected that this has introduced a crash in issue 1160146. I am not marking this as fixed until we've addressed the crash. I don't want to merge back crashes.

**Comment 30** by [khush...@chromium.org](#) on Tue, Jan 5, 2021, 5:34 PM EST Project Member  
**Cc:** -khush...@chromium.org

**Comment 31** by [jonr...@chromium.org](#) on Fri, Jan 8, 2021, 5:25 PM EST Project Member  
**Labels:** Merge-Request-87 Merge-Request-88

A fix to the blocking crash in issue 1160146 was landed by <https://chromium-review.googlesource.com/c/chromium/src/+2611174>

I'm going to request a merge for both that as well as [r837779](#)

**Comment 32** by [sheriffbot](#) on Fri, Jan 8, 2021, 5:26 PM EST Project Member  
**Labels:** -Merge-Request-88 Merge-Review-88 Hotlist-Merge-Review

This bug requires manual review: We are only 10 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](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: govind@ (Android), bindusuvama@ (iOS), marinakz@ (ChromeOS), srinivassista @ (Desktop)

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

**Comment 33** by [jonr...@chromium.org](#) on Fri, Jan 8, 2021, 6:21 PM EST Project Member  
1. M-88 yes  
M-87 no, as it is after stable release  
2. <https://chromium-review.googlesource.com/c/chromium/src/+2585790>  
<https://chromium-review.googlesource.com/c/chromium/src/+2611174>  
3. Yes  
4. Yes, was requested for M-87. Should be at least merged into M-88  
5. Security bug found in Stable  
6. No  
7. N/A

**Comment 34** by [gov...@chromium.org](#) on Fri, Jan 8, 2021, 6:22 PM EST Project Member  
**Cc:** adetaylor@chromium.org  
+adetaylor@ (Security TPM) for M88 merge review. Thank you.

**Comment 35** by [adetaylor@chromium.org](#) on Fri, Jan 8, 2021, 6:51 PM EST Project Member  
**Labels:** -Merge-Request-87 -Merge-Review-88 Merge-Rejected-88 Merge-Rejected-87

As this is a medium severity bug, and there's already been some hints of instability, I think we should let this organically release in M89 instead of merging to M88 (which doesn't have too long till we release).

**Comment 36** by [jonr...@chromium.org](#) on Mon, Jan 11, 2021, 12:15 PM EST Project Member  
**Status:** Fixed (was: Assigned)

With the merge review leading to rejections, there is no more work to be done here. All follow up work for Thumbnails is tracked in [issue 1163124](#).

I'll close this as fixed now. Thank you everyone for the help in debugging the issue, and getting a fix landed!

**Comment 37** by [sheriffbot](#) on Mon, Jan 11, 2021, 12:42 PM EST Project Member  
**Labels:** reward-topanel

**Comment 38** by [sheriffbot](#) on Mon, Jan 11, 2021, 1:56 PM EST Project Member  
**Labels:** -Restrict-View-SecurityTeam Restrict-View-SecurityNotify

**Comment 39** by [sheriffbot](#) on Thu, Jan 14, 2021, 4:22 PM EST Project Member  
**Labels:** external\_security\_report

**Comment 40** by [adetaylor@google.com](#) on Wed, Jan 20, 2021, 7:01 PM EST Project Member  
**Labels:** -reward-potential

**Comment 41** by [amyressler@google.com](#) on Wed, Jan 20, 2021, 7:10 PM EST Project Member  
**Labels:** -reward-topanel reward-unpaid reward-3000

\*\*\* 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](mailto:security-vrp@chromium.org) with any questions.  
\*\*\*\*\*

**Comment 42** by [amyressler@google.com](mailto:amyressler@google.com) on Wed, Jan 20, 2021, 7:32 PM EST Project Member

Congratulations, Ivan! The VRP Panel has decided to award you \$3,000 for this report. A member of the finance will soon be in touch to arrange payment. Nice work and thank you for your submission!

**Comment 43** by [amyressler@google.com](mailto:amyressler@google.com) on Thu, Jan 21, 2021, 1:52 PM EST Project Member

**Labels:** -reward-unpaid reward-inprocess

**Comment 44** by [susah...@gmail.com](mailto:susah...@gmail.com) on Sat, Jan 23, 2021, 1:06 PM EST

Thank you very much for the reward!

**Comment 45** by [adetaylor@google.com](mailto:adetaylor@google.com) on Fri, Feb 26, 2021, 1:08 PM EST Project Member

**Labels:** Release-0-M89

**Comment 46** by [adetaylor@google.com](mailto:adetaylor@google.com) on Mon, Mar 1, 2021, 7:27 PM EST Project Member

**Labels:** CVE-2021-21171 CVE\_description-missing

**Comment 47** by [vsavu@google.com](mailto:vsavu@google.com) on Wed, Mar 3, 2021, 5:47 AM EST Project Member

**Labels:** LTS-Merge-Request-86

**Comment 48** by [vsavu@google.com](mailto:vsavu@google.com) on Wed, Mar 3, 2021, 6:01 AM EST Project Member

**Labels:** LTS-Security-86

**Comment 49** by [gianluca@google.com](mailto:gianluca@google.com) on Wed, Mar 3, 2021, 10:36 AM EST Project Member

**Labels:** LTS-Merge-Approved-86

**Comment 50** by [sheriffbot](#) on Wed, Mar 3, 2021, 12:22 PM EST Project Member

**Labels:** -M-87 Target-89 M-89

**Comment 51** by [vsavu@google.com](mailto:vsavu@google.com) on Thu, Mar 4, 2021, 8:36 AM EST Project Member

**Labels:** -LTS-Merge-Approved-86 -LTS-Merge-Request-86 LTS-Security-Failed-86

**Comment 52** by [amyressler@google.com](mailto:amyressler@google.com) on Tue, Mar 9, 2021, 12:58 PM EST Project Member

**Labels:** -CVE\_description-missing CVE\_description-submitted

**Comment 53** by [sheriffbot](#) on Mon, Apr 19, 2021, 1:49 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

**Comment 54** by [alexe...@gmail.com](mailto:alexe...@gmail.com) on Wed, Jun 9, 2021, 3:45 PM EDT

For me this issue is still reproduced on Chrome if the internet connection is slow. May happen if the mobile signal is weak or on the restricted connection.

To reproduce it:

1. Restrict internet connection for the device to 64 kbit/s;
2. Use original issue steps.

Is such case of low speed connection an issue needs to be fixed?

**Comment 55** by [jonr...@chromium.org](mailto:jonr...@chromium.org) on Wed, Jun 9, 2021, 4:51 PM EDT Project Member

Could you file a new issue for this? Feel free to list it as related to this one.

Please include steps to reproduce.

- Including if you were using the test page (<http://csreis.github.io/tests/interrupt-popup.html>) or if you were using other sites.
- As well as what was the error looked like.

**Comment 56** by [alexe...@gmail.com](mailto:alexe...@gmail.com) on Thu, Jun 10, 2021, 7:37 AM EDT

Thanks, I asked in the new issue: <https://bugs.chromium.org/p/chromium/issues/detail?id=1218366>.