☆ Starred by 2 users

Owner:	solomonkinard@chromium.org
CC:	karandeepb@chromium.org solomonkinard@chromium.org pbos@chromium.org tbergquist@chromium.org corising@chromium.org dfried@chromium.org tjudkins@chromium.org
Status:	Fixed *(Closed)*
Components:	Platform>Extensions>API
Modified:	Aug 27, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Windows, Chrome, Mac, Lacros
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
reward-10000
Merge-na
Security_Impact-Stable
Security_Severity-Medium
allpublic
reward-inprocess
CVE_description-submitted
M-90
Target-89
Target-90
LTS-Security-86
LTS-Security-NotApplicable-86
external_security_report
external_security_bug
LTS-Merged-90
LTS-Security-90
Release-0-M91
CVE-2021-30542

Issue 1184954: Security: Heap-use-after-free in TabStrip::GetSizeNeededForViews
Reported by chrom...@gmail.com on Thu, Mar 4, 2021, 7:26 PM EST

Code

Chrome Version: 91.0.4435.0 (Official Build) canary (x86_64)
Operating System: MacOS

REPRODUCTION CASE

similar to ~~issue 1171049~~.

1. Install the extension.
2. Try to detach the opened tab by the colored circle.

```
==2197==ERROR: AddressSanitizer: heap-use-after-free on address 0x61800031edd8 at pc 0x00012b8b10fe bp 0x7fff50a424d0 sp 0x7fff50a424c8
READ of size 4 at 0x61800031edd8 thread T0
   #0 0x12b8b10fd in TabStrip::GetSizeNeededForViews(std::__1::vector<TabSlotView*, std::__1::allocator<TabSlotView*> > const&) size.h:49
   #1 0x12b86e47f in TabDragController::MoveAttached(gfx::Point const&, bool) tab_drag_controller.cc:1488
   #2 0x12b86bcf6 in TabDragController::ContinueDragging(gfx::Point const&) tab_drag_controller.cc:847
   #3 0x12b864b13 in TabDragController::Drag(gfx::Point const&) tab_drag_controller.cc:604
   #4 0x12b8c5697 in TabStrip::TabDragContextImpl::ContinueDrag(views::View*, ui::LocatedEvent const&) tab_strip.cc:442
   #5 0x12b8d1b4b in TabStrip::OnMouseDragged(ui::MouseEvent const&) tab_strip.cc:3666
   #6 0x12a49aa3d in views::View::ProcessMouseDragged(ui::MouseEvent*) view.cc:2990
   #7 0x1223c3b4f in ui::EventHandler::OnEvent(ui::Event*) event_handler.cc
   #8 0x1223c1f39 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) event_dispatcher.cc:191
   #9 0x1223c1714 in ui::EventDispatcherDelegate::DispatchEventToTarget(ui::EventTarget*, ui::Event*) event_dispatcher.cc:84
   #10 0x1223c1450 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) event_dispatcher.cc:56
   #11 0x12a4d275a in views::internal::RootView::OnMouseDragged(ui::MouseEvent const&) root_view.cc:457
   #12 0x12a4ec697 in views::Widget::OnMouseEvent(ui::MouseEvent*) widget.cc:1335
   #13 0x12a528e6b in non-virtual thunk to views::NativeWidgetMacNSWindowHost::OnMouseEvent(std::__1::unique_ptr<ui::Event, std::__1::default_delete<ui::Event> >)
native_widget_mac_ns_window_host.mm:804
   #14 0x1267bf99b in -[BridgedContentView mouseEvent:] bridged_content_view.mm:586
   #15 0x1267bce7d in -[BridgedContentView processCapturedMouseEvent:] bridged_content_view.mm:308
   #16 0x1267ca91b in ___ZN12remote_cocoa17CocoaMouseCapture14ActiveEventTap4InitEv_block_invoke mouse_capture.mm:91
   #17 0x7fff7f5567f9 in _NSSendEventToObservers+0x173 (AppKit:x86_64+0x1c77f9)
   #18 0x7fff7fb4f23e in -[NSApplication(NSEvent) sendEvent:]+0x36 (AppKit:x86_64+0x7c023e)
   #19 0x11ffb67d4 in __34-[BrowserCrApplication sendEvent:]_block_invoke chrome_browser_application_mac.mm:335
   #20 0x11edf0299 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xbae8299)
   #21 0x11ffb5b4e in -[BrowserCrApplication sendEvent:] chrome_browser_application_mac.mm:319
   #22 0x7fff7f3ca3d6 in -[NSApplication run]+0x3e9 (AppKit:x86_64+0x3b3d6)
   #23 0x11ee046fa in base::MessagePumpNSApplication::DoRun(base::MessagePump::Delegate*) message_pump_mac.mm:691
   #24 0x11ee00818 in base::MessagePumpCFRunLoopBase::Run(base::MessagePump::Delegate*) message_pump_mac.mm:149
   #25 0x11ed16a3b in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
thread_controller_with_message_pump_impl.cc:460
   #26 0x11ec5425e in base::RunLoop::Run(base::Location const&) run_loop.cc:133
   #27 0x11f305823 in ChromeBrowserMainParts::MainMessageLoopRun(int*) chrome_browser_main.cc:1732
   #28 0x11795c467 in content::BrowserMainLoop::RunMainMessageLoopParts() browser_main_loop.cc:970
   #29 0x117960b51 in content::BrowserMainRunnerImpl::Run() browser_main_runner_impl.cc:150
```

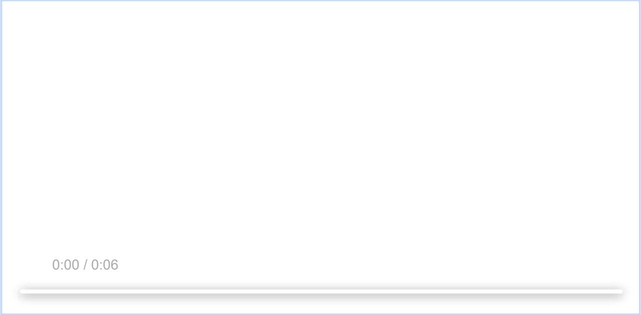
```
    #30 0x1179559fc in content::BrowserMain(content::MainFunctionParams const&) browser_main.cc:47
    #31 0x11ea307c4 in content::ContentMainRunnerImpl::RunBrowser(content::MainFunctionParams&, bool) content_main_runner_impl.cc:581
    #32 0x11ea2faf4 in content::ContentMainRunnerImpl::Run(bool) content_main_runner_impl.cc:944
    #33 0x11ea2cce6 in content::RunContentProcess(content::ContentMainParams const&, content::ContentMainRunner*) content_main.cc:372
    #34 0x11ea2d2fc in content::ContentMain(content::ContentMainParams const&) content_main.cc:398
    #35 0x1133101d5 in ChromeMain chrome_main.cc:141
    #36 0x10f1bc40f in main chrome_exe_main_mac.cc:114
    #37 0x7fff9752a234 in start+0x0 (libdyld.dylib:x86_64+0x5234)

0x61800031edd8 is located 344 bytes inside of 840-byte region [0x61800031ec80,0x61800031efc8)
freed by thread T0 here:
    #0 0x10f3b8fc9  (libclang_rt.asan_osx_dynamic.dylib:x86_64+0x44fc9)
    #1 0x12b894cdb in TabGroupViews::~TabGroupViews() memory:1335
    #2 0x12b8e0d83 in std::__1::__tree<std::__1::__value_type<tab_groups::TabGroupId, std::__1::unique_ptr<TabGroupViews, std::__1::default_delete<TabGroupViews> >
>, std::__1::__map_value_compare<tab_groups::TabGroupId, std::__1::__value_type<tab_groups::TabGroupId, std::__1::unique_ptr<TabGroupViews,
std::__1::default_delete<TabGroupViews> > >, std::__1::less<tab_groups::TabGroupId>, true>, std::__1::allocator<std::__1::__value_type<tab_groups::TabGroupId,
std::__1::unique_ptr<TabGroupViews, std::__1::default_delete<TabGroupViews> > > >
>::erase(std::__1::__tree_const_iterator<std::__1::__value_type<tab_groups::TabGroupId, std::__1::unique_ptr<TabGroupViews, std::__1::default_delete<TabGroupViews> >
>, std::__1::__tree_node<std::__1::__value_type<tab_groups::TabGroupId, std::__1::unique_ptr<TabGroupViews, std::__1::default_delete<TabGroupViews> > >, void*>*,
long>) memory:1596
    #3 0x12b8bd457 in TabStrip::OnGroupClosed(tab_groups::TabGroupId const&) __tree:2445
    #4 0x12b820c6c in BrowserTabStripController::OnTabGroupChanged(TabGroupChange const&) browser_tab_strip_controller.cc:738
    #5 0x12ad3eb10 in TabStripModel::CloseTabGroup(tab_groups::TabGroupId const&) tab_strip_model.cc:1201
    #6 0x12ad2b748 in TabStripModel::UngroupTab(int) tab_strip_model.cc:2189
    #7 0x12ad3ab2a in TabStripModel::GroupTab(int, tab_groups::TabGroupId const&) tab_strip_model.cc:2204
    #8 0x12ad47df3 in TabStripModel::MoveAndSetGroup(int, int, base::Optional<tab_groups::TabGroupId>) tab_strip_model.cc:2153
    #9 0x12ad38a44 in TabStripModel::MoveTabsAndSetGroupImpl(std::__1::vector<int, std::__1::allocator<int> > const&, int, base::Optional<tab_groups::TabGroupId>)
tab_strip_model.cc:2122
    #10 0x12ad37a39 in TabStripModel::AddToNewGroupImpl(std::__1::vector<int, std::__1::allocator<int> > const&, tab_groups::TabGroupId const&) tab_strip_model.cc:2069
    #11 0x12ad37465 in TabStripModel::AddToNewGroup(std::__1::vector<int, std::__1::allocator<int> > const&) tab_strip_model.cc:1057
    #12 0x1297febc7 in extensions::TabsGroupFunction::Run() tabs_api.cc:1844
    #13 0x119c485b7 in ExtensionFunction::RunWithValidation() extension_function.cc:466
    #14 0x119c507aa in extensions::ExtensionFunctionDispatcher::DispatchWithCallbackInternal(ExtensionHostMsg_Request_Params const&, content::RenderFrameHost*,
int, base::RepeatingCallback<void (ExtensionFunction::ResponseType, base::ListValue const&, std::__1::basic_string<char, std::__1::char_traits<char>,
std::__1::allocator<char> > const&)> const&) extension_function_dispatcher.cc:383
    #15 0x119c4f91a in extensions::ExtensionFunctionDispatcher::Dispatch(ExtensionHostMsg_Request_Params const&, content::RenderFrameHost*, int)
extension_function_dispatcher.cc:253
    #16 0x119cc4dd6 in extensions::ExtensionWebContentsObserver::OnMessageReceived(IPC::Message const&, content::RenderFrameHost*)
extension_web_contents_observer.cc:327
    #17 0x129889e9f in extensions::ChromeExtensionWebContentsObserver::OnMessageReceived(IPC::Message const&, content::RenderFrameHost*)
chrome_extension_web_contents_observer.cc:94
    #18 0x118a49f72 in content::WebContentsImpl::OnMessageReceived(content::RenderFrameHostImpl*, IPC::Message const&) web_contents_impl.cc:1152
    #19 0x11854e376 in content::RenderFrameHostImpl::OnMessageReceived(IPC::Message const&) render_frame_host_impl.cc:1940
    #20 0x1222b6f7d in IPC::ChannelProxy::Context::OnDispatchMessage(IPC::Message const&) ipc_channel_proxy.cc:325
    #21 0x11ecda14f in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) callback.h:101
    #22 0x11ed1581a in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
thread_controller_with_message_pump_impl.cc:351
    #23 0x11ed15037 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork() thread_controller_with_message_pump_impl.cc:264
    #24 0x11ee02e03 in base::MessagePumpCFRunLoopBase::RunWork() message_pump_mac.mm:358
    #25 0x11edf0299 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xbae8299)
    #26 0x11ee017d5 in base::MessagePumpCFRunLoopBase::RunWorkSource(void*) message_pump_mac.mm:334
    #27 0x7fff81901e50 in __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__+0x10 (CoreFoundation:x86_64+0xa4e50)
    #28 0x7fff818e30cb in __CFRunLoopDoSources0+0x22b (CoreFoundation:x86_64+0x860cb)
    #29 0x7fff818e25b5 in __CFRunLoopRun+0x3a5 (CoreFoundation:x86_64+0x855b5)

previously allocated by thread T0 here:
    #0 0x10f3b8e80  (libclang_rt.asan_osx_dynamic.dylib:x86_64+0x44e80)
    #1 0x11eb54067 in operator new(unsigned long) new.cpp:67
    #2 0x12b8949cc in TabGroupViews::TabGroupViews(TabStrip*, tab_groups::TabGroupId const&) memory:2006
    #3 0x12b8bb2f2 in TabStrip::OnGroupCreated(tab_groups::TabGroupId const&) tab_strip.cc:1473
    #4 0x12b820a15 in BrowserTabStripController::OnTabGroupChanged(TabGroupChange const&) browser_tab_strip_controller.cc:689
    #5 0x12ad3cad0 in TabStripModel::CreateTabGroup(tab_groups::TabGroupId const&) tab_strip_model.cc:1168
    #6 0x12ad1a69a in TabGroup::AddTab() tab_group.cc:65
    #7 0x12ad3b110 in TabStripModel::GroupTab(int, tab_groups::TabGroupId const&) tab_strip_model.cc:2212
    #8 0x12ad47df3 in TabStripModel::MoveAndSetGroup(int, int, base::Optional<tab_groups::TabGroupId>) tab_strip_model.cc:2153
    #9 0x12ad38a44 in TabStripModel::MoveTabsAndSetGroupImpl(std::__1::vector<int, std::__1::allocator<int> > const&, int, base::Optional<tab_groups::TabGroupId>)
tab_strip_model.cc:2122
    #10 0x12ad37a39 in TabStripModel::AddToNewGroupImpl(std::__1::vector<int, std::__1::allocator<int> > const&, tab_groups::TabGroupId const&) tab_strip_model.cc:2069
    #11 0x12ad37465 in TabStripModel::AddToNewGroup(std::__1::vector<int, std::__1::allocator<int> > const&) tab_strip_model.cc:1057
    #12 0x1297febc7 in extensions::TabsGroupFunction::Run() tabs_api.cc:1844
    #13 0x119c485b7 in ExtensionFunction::RunWithValidation() extension_function.cc:466
    #14 0x119c507aa in extensions::ExtensionFunctionDispatcher::DispatchWithCallbackInternal(ExtensionHostMsg_Request_Params const&, content::RenderFrameHost*,
int, base::RepeatingCallback<void (ExtensionFunction::ResponseType, base::ListValue const&, std::__1::basic_string<char, std::__1::char_traits<char>,
std::__1::allocator<char> > const&)> const&) extension_function_dispatcher.cc:383
    #15 0x119c4f91a in extensions::ExtensionFunctionDispatcher::Dispatch(ExtensionHostMsg_Request_Params const&, content::RenderFrameHost*, int)
extension_function_dispatcher.cc:253
    #16 0x119cc4dd6 in extensions::ExtensionWebContentsObserver::OnMessageReceived(IPC::Message const&, content::RenderFrameHost*)
extension_web_contents_observer.cc:327
    #17 0x129889e9f in extensions::ChromeExtensionWebContentsObserver::OnMessageReceived(IPC::Message const&, content::RenderFrameHost*)
chrome_extension_web_contents_observer.cc:94
    #18 0x118a49f72 in content::WebContentsImpl::OnMessageReceived(content::RenderFrameHostImpl*, IPC::Message const&) web_contents_impl.cc:1152
    #19 0x11854e376 in content::RenderFrameHostImpl::OnMessageReceived(IPC::Message const&) render_frame_host_impl.cc:1940
    #20 0x1222b6f7d in IPC::ChannelProxy::Context::OnDispatchMessage(IPC::Message const&) ipc_channel_proxy.cc:325
    #21 0x11ecda14f in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) callback.h:101
    #22 0x11ed1581a in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
thread_controller_with_message_pump_impl.cc:351
    #23 0x11ed15037 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork() thread_controller_with_message_pump_impl.cc:264
    #24 0x11ee02e03 in base::MessagePumpCFRunLoopBase::RunWork() message_pump_mac.mm:358
    #25 0x11edf0299 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xbae8299)
    #26 0x11ee017d5 in base::MessagePumpCFRunLoopBase::RunWorkSource(void*) message_pump_mac.mm:334
    #27 0x7fff81901e50 in __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__+0x10 (CoreFoundation:x86_64+0xa4e50)
    #28 0x7fff818e30cb in __CFRunLoopDoSources0+0x22b (CoreFoundation:x86_64+0x860cb)
    #29 0x7fff818e25b5 in __CFRunLoopRun+0x3a5 (CoreFoundation:x86_64+0x855b5)

SUMMARY: AddressSanitizer: heap-use-after-free size.h:49 in TabStrip::GetSizeNeededForViews(std::__1::vector<TabSlotView*, std::__1::allocator<TabSlotView*> > const&)
Shadow bytes around the buggy address:
  0x1c3000063d60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x1c3000063d70: fd fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c3000063d80: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c3000063d90: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x1c3000063da0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x1c3000063db0: fd fd fd fd fd fd fd fd fd fd fd[fd]fd fd fd fd
  0x1c3000063dc0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
```

```
 0x1c3000063dd0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
 0x1c3000063de0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
 0x1c3000063df0: fd fd fd fd fd fd fd fd fd fa fa fa fa fa fa fa
 0x1c3000063e00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
 Addressable:           00
 Partially addressable: 01 02 03 04 05 06 07
 Heap left redzone:       fa
 Freed heap region:       fd
 Stack left redzone:      f1
 Stack mid redzone:       f2
 Stack right redzone:     f3
 Stack after return:      f5
 Stack use after scope:   f8
 Global redzone:          f9
 Global init order:       f6
 Poisoned by user:        f7
 Container overflow:      fc
 Array cookie:            ac
 Intra object redzone:    bb
 ASan internal:           fe
 Left alloca redzone:     ca
 Right alloca redzone:    cb
 Shadow gap:              cc
```

manifest.json
389 bytes View Download

background.js
251 bytes View Download

screen.mov
7.3 MB View Download



Comment 1 by sheriffbot on Thu, Mar 4, 2021, 7:29 PM EST Project Member

Labels: external_security_report

Comment 2 by dominickn@chromium.org on Thu, Mar 4, 2021, 7:47 PM EST Project Member

Status: Assigned (was: Unconfirmed)
Owner: tbergquist@chromium.org
Cc: dfried@chromium.org karandeepb@chromium.org
Labels: Security_Impact-Stable Security_Severity-Medium OS-Chrome OS-Linux OS-Mac OS-Windows OS-Lacros Pri-1
Components: UI>Browser>TabStrip Platform>Extensions>API

Looks like this is down to calling the chrome.tabs.group() method[1] (new in Chrome 88 with the tab groups feature) in the extension.

+folks from ~~Issue 1171049~~ and extensions folks. Assigning Medium severity due to needing to install an extension.

1. https://developer.chrome.com/docs/extensions/reference/tabs/#method-group

Comment 3 by chrom...@gmail.com on Thu, Mar 4, 2021, 7:50 PM EST

Thanks for the update!

Shouldn't be higher as in ~~issue 1151700~~?

Comment 4 by dominickn@chromium.org on Fri, Mar 5, 2021, 1:16 AM EST Project Member

Per [1], memory corruption that needs an extension to be installed is typically marked as Medium severity.

1. https://chromium.googlesource.com/chromium/src/+/master/docs/security/severity-guidelines.md

Comment 5 by sheriffbot on Fri, Mar 5, 2021, 1:01 PM EST Project Member

Labels: Target-89 M-89

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

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

Comment 6 by sheriffbot on Wed, Mar 10, 2021, 8:03 PM EST Project Member

Labels: reward-potential

Comment 7 by chrom...@gmail.com on Tue, Mar 16, 2021, 12:29 PM EDT

Friendly ping for tbergquist :)

Comment 8 by zhangtiff@google.com on Wed, Mar 17, 2021, 7:14 PM EDT Project Member

Labels: -reward-potential external_security_bug

Comment 9 by sheriffbot on Fri, Mar 19, 2021, 12:21 PM EDT Project Member

tbergquist: 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 10 by sheriffbot on Sat, Apr 3, 2021, 12:21 PM EDT Project Member

tbergquist: Uh oh! This issue still open and hasn't been updated in the last 29 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 11 by chrom...@gmail.com on Wed, Apr 14, 2021, 4:49 PM EDT

Any update on this bug? Thanks!

Comment 12 by sheriffbot on Thu, Apr 15, 2021, 12:21 PM EDT Project Member

Labels: -M-89 M-90 Target-90

Comment 13 by tbergquist@chromium.org on Mon, Apr 19, 2021, 8:19 PM EDT Project Member

Status: Started (was: Assigned)

I took a stab at this today!

First approach: End the drag if the dragged header is closed.
Works if the header is closed only after the drag has properly started, but if it's before drag start (i.e. we haven't moved the mouse far enough yet), nothing is being dragged so we still get a zombie drag session.

Second approach: End the drag if any header is closed.
Work better, but seems to run into issues when reverting header drags after we've detached into another window. The process of reverting the drag closes the (copy of the) group header, which triggers ending the drag within the process of ending the drag, and that unsurprisingly tends to go sideways.

Key difference here seems to be that while tabs have a canonical identity that's guaranteed to be unique (the WebContents managed by unique_ptr), group headers have no such guarantee. When dragging a group header into another tabstrip, we don't *move* the group so much as create a new one that's identical and ditch the old one.

For this general category of fix to work, we would need to be able to differentiate between a group closing and a group 'closing' as part of moving it around. That might be spaghetti-able

https://chromium-review.googlesource.com/c/chromium/src/+/2837489 << CL with the second approach (which doesn't work, as mentioned) and the logging that got me this far.

An alternate approach would be to make a drag session resilient to the dragged group disappearing at any time without notice. But it'd be much nicer if we could just make the giving notice part work.

Comment 14 by tbergquist@chromium.org on Wed, May 12, 2021, 2:48 PM EDT Project Member

Cc: solomonkinard@chromium.org

Comment 15 by tbergquist@chromium.org on Wed, May 12, 2021, 2:51 PM EDT Project Member

Cc: pbos@chromium.org

Comment 16 by tbergquist@chromium.org on Thu, May 13, 2021, 3:00 PM EDT Project Member

Status: Assigned (was: Started)
Owner: solomonkinard@chromium.org
Cc: corising@chromium.org tbergquist@chromium.org

I tried a slightly more targeted combination of the previous two approaches here: https://chromium-review.googlesource.com/c/chromium/src/+/2893209 Ran into the same core issues as in the second approach, where reverting a drag closes a group header, triggering a nested end-the-drag-session process. My analysis in #13 still stands.

The security issues will be fixed in https://chromium-review.googlesource.com/c/chromium/src/+/2891080, so at this point this is a code health / brittleness issue. Our team has some broader fixes to this class of problem in mind for the near-to-medium future, so I don't think it makes sense to try to whack this particular mole right now.

Solomon, I'll just hand this to you to track with all the other extensions-related stuff.

Comment 17 by chrom...@gmail.com on Mon, May 17, 2021, 5:36 PM EDT

Double-check, fixed on Chromium 92.0.4511.0 (Developer Build) (x86_64) refs/heads/master@{#883573}.

Comment 18 by chrom...@gmail.com on Thu, May 20, 2021, 2:23 PM EDT

is this issue should be marked as fixed?

Comment 19 by solomonkinard@chromium.org on Thu, May 20, 2021, 2:48 PM EDT Project Member

Status: Fixed (was: Assigned)

Comment 20 by sheriffbot on Fri, May 21, 2021, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 21 by sheriffbot on Fri, May 21, 2021, 2:01 PM EDT Project Member

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

Comment 22 by sheriffbot on Fri, May 21, 2021, 2:27 PM EDT Project Member

Labels: Merge-Request-91

This is sufficiently serious that it should be merged to beta. But I can't see a Chromium repo commit here, so you will need to investigate what - if anything - needs to be merged to M91. Is there a fix in some other repo which should be merged? Or, perhaps this ticket is a duplicate of some other ticket which has the real fix: please track that down and ensure it is merged appropriately.

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

Comment 23 by sheriffbot on Fri, May 21, 2021, 2:31 PM EDT Project Member

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

This bug requires manual review: We are only 3 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 24 by solomonkinard@chromium.org on Fri, May 21, 2021, 6:52 PM EDT Project Member

crrev.com/c/2904568 is merged into M91.

Comment 25 by adetaylor@google.com on Thu, Jun 3, 2021, 2:24 PM EDT Project Member

Labels: -Merge-Review-91 Release-0-M91 relnotes_update_needed Merge-NA

Comment 26 by asumaneev@google.com on Fri, Jun 4, 2021, 2:02 PM EDT Project Member

Labels: LTS-Security-86 LTS-Security-NotApplicable-86

Marking as not applicable for LTS-86. There already was an attempt to merge the fix:
https://chromium-review.googlesource.com/c/chromium/src/+/2919770

Comment 27 by amyressler@google.com on Mon, Jun 7, 2021, 10:45 AM EDT Project Member

Labels: CVE-2021-30542 CVE_description-missing

Comment 28 by amyressler@chromium.org on Mon, Jun 7, 2021, 11:00 AM EDT Project Member

Labels: -relnotes_update_needed

rel notes updated

Comment 29 by amyressler@google.com on Mon, Jun 7, 2021, 3:27 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 30 by amyressler@google.com on Thu, Jun 10, 2021, 12:32 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-10000

*** 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 31 by amyressler@chromium.org on Thu, Jun 10, 2021, 12:59 PM EDT Project Member

Congratulations, Khalil! The VRP Panel has decided to award you $10,000 for this report. Excellent work!

Comment 32 by amyressler@google.com on Mon, Jun 14, 2021, 11:32 AM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 33 by vsavu@google.com on Mon, Jun 14, 2021, 12:32 PM EDT Project Member

Labels: LTS-Merged-90 LTS-Security-90

Comment 34 by sheriffbot on Fri, Aug 27, 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