Starred by 4 users

Owner:	tbergquist@chromium.org
CC:	connily@chromium.org
Status:	Verified *(Closed)*
Components:	----
Modified:	May 14, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Windows, Chrome, Mac, Fuchsia, Lacros
Pri:	1
Type:	Bug-Security

Security_Impact-Stable
Security_Severity-High
reward-7500
ReleaseBlock-Stable
allpublic
reward-inprocess
CVE_description-submitted
M-90
Target-90
Merge-Rejected-88
LTS-Security-86
LTS-Security-NotApplicable-86
Release-0-M89
external_security_report
merge-merged-4389
merge-merged-89
CVE-2021-21161

Issue 1173702: Security: Heap buffer overflow in Tab Groups

Reported by chrom...@gmail.com on Tue, Feb 2, 2021, 5:15 PM EST

Code

Chrome Version: 90.0.4406.0 (Official Build) canary (x86_64) and stable
Operating System: MacOS

REPRODUCTION CASE

Similar to ~~issue 1163845~~

1. python -m SimpleHTTPServer
2. out/asan/chrome --user-data-dir=/tmp/xxxx "about:blank" "http://localhost:8000/poc.html"
3. Add http://localhost:8000/poc.html to a new group.
4. Click on the button then click on the button
5. Now hold the mouse over the grey point and keep dragging the tab from the right to the left

```
==1399==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6100002d0d28 at pc 0x00012e902fb9 bp 0x7fff53494d10 sp 0x7fff53494d08
READ of size 8 at 0x6100002d0d28 thread T0
    #0 0x12e902fb8 in TabStrip::SetSelection(ui::ListSelectionModel const&) view_model.h:81
    #1 0x12e865f4a in BrowserTabStripController::OnTabStripModelChanged(TabStripModel*, TabStripModelChange const&, TabStripSelectionChange const&)
browser_tab_strip_controller.cc:689
    #2 0x12dd72a0b in TabStripModel::SetSelection(ui::ListSelectionModel, TabStripModelObserver::ChangeReason, bool) tab_strip_model.cc:1884
    #3 0x12dd7aa92 in TabStripModel::SetSelectionFromModel(ui::ListSelectionModel) tab_strip_model.cc:915
    #4 0x12e8bcea5 in TabDragController::CompleteDrag() tab_drag_controller.cc:1794
    #5 0x12e8b0290 in TabDragController::EndDragImpl(TabDragController::EndDragType) tab_drag_controller.cc:1531
    #6 0x12e8a7147 in TabDragController::EndDrag(EndDragReason) tab_drag_controller.cc:646
    #7 0x12e909363 in TabStrip::EndDrag(EndDragReason) tab_strip.cc:477
    #8 0x12e8fbc5c in TabStrip::RemoveTabAt(content::WebContents*, int, bool) tab_strip.cc:1390
    #9 0x12e865d0d in BrowserTabStripController::OnTabStripModelChanged(TabStripModel*, TabStripModelChange const&, TabStripSelectionChange const&)
browser_tab_strip_controller.cc:647
    #10 0x12dd702bb in TabStripModel::SendDetachWebContentsNotifications(TabStripModel::DetachNotifications*) tab_strip_model.cc:526
    #11 0x12dd7703f in TabStripModel::InternalCloseTabs(base::span<content::WebContents* const, 18446744073709551615ul>, unsigned int) tab_strip_model.cc:1754
    #12 0x12dd77850 in TabStripModel::CloseWebContentsAt(int, unsigned int) tab_strip_model.cc:731
    #13 0x11c3189ae in content::WebContentsImpl::Close(content::RenderViewHost*) web_contents_impl.cc:6993
    #14 0x11be105c1 in content::RenderViewHostImpl::ClosePage() render_view_host_impl.cc:665
    #15 0x11bd5f07e in content::RenderFrameHostManager::BeforeUnloadCompleted(bool, base::TimeTicks const&) render_frame_host_manager.cc:341
    #16 0x11bd3e671 in base::internal::Invoker<base::internal::BindState<content::RenderFrameHostImpl::ProcessBeforeUnloadCompletedFromFrame(bool, bool,
content::RenderFrameHostImpl*, bool, base::TimeTicks const&, base::TimeTicks const&)::$_5, base::WeakPtr<content::RenderFrameHostImpl>, base::TimeTicks, bool,
bool>, void ()>::RunOnce(base::internal::BindStateBase*) render_frame_host_impl.cc:3557
    #17 0x11bca9b89 in content::RenderFrameHostImpl::ProcessBeforeUnloadCompletedFromFrame(bool, bool, content::RenderFrameHostImpl*, bool, base::TimeTicks
const&, base::TimeTicks const&) callback.h:101
    #18 0x11bcd2c65 in content::RenderFrameHostImpl::ProcessBeforeUnloadCompleted(bool, bool, base::TimeTicks const&, base::TimeTicks const&)
render_frame_host_impl.cc:3458
    #19 0x11bd57929 in base::internal::Invoker<base::internal::BindState<content::RenderFrameHostImpl::SendBeforeUnload(bool,
base::WeakPtr<content::RenderFrameHostImpl>)::$_23, base::WeakPtr<content::RenderFrameHostImpl> >, void (bool, base::TimeTicks,
base::TimeTicks)>::RunOnce(base::internal::BindStateBase*, bool, base::TimeTicks&&, base::TimeTicks&&) render_frame_host_impl.cc:9465
```

```
    #20 0x11929905f in blink::mojom::LocalFrame_BeforeUnload_ForwardToCallback::Accept(mojo::Message*) callback.h:101
    #21 0x1233247ee in mojo::InterfaceEndpointClient::HandleValidatedMessage(mojo::Message*) interface_endpoint_client.cc:549
    #22 0x12332d238 in mojo::MessageDispatcher::Accept(mojo::Message*) message_dispatcher.cc:41
    #23 0x1251aa16d in IPC::(anonymous namespace)::ChannelAssociatedGroupController::AcceptOnProxyThread(mojo::Message) ipc_mojo_bootstrap.cc:945
    #24 0x1251a313c in base::internal::Invoker<base::internal::BindState<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message),
scoped_refptr<IPC::(anonymous namespace)::ChannelAssociatedGroupController>, mojo::Message>, void ()>::RunOnce(base::internal::BindStateBase*) bind_internal.h:498
    #25 0x121b21b15 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) callback.h:101
    #26 0x121b5fe30 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
thread_controller_with_message_pump_impl.cc:351
    #27 0x121b5f5d7 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork() thread_controller_with_message_pump_impl.cc:264
    #28 0x121c4fc18 in ___ZN4base24MessagePumpCFRunLoopBase13RunWorkSourceEPv_block_invoke message_pump_mac.mm:358
    #29 0x121c3d1b9 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xb4221b9)
    #30 0x121c4e365 in base::MessagePumpCFRunLoopBase::RunWorkSource(void*) message_pump_mac.mm:334
    #31 0x7fffa8ac6e50 in __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__+0x10 (CoreFoundation:x86_64+0xa4e50)
    #32 0x7fffa8aa80cb in __CFRunLoopDoSources0+0x22b (CoreFoundation:x86_64+0x860cb)
    #33 0x7fffa8aa75b5 in __CFRunLoopRun+0x3a5 (CoreFoundation:x86_64+0x855b5)
    #34 0x7fffa8aa6fb3 in CFRunLoopRunSpecific+0x1a3 (CoreFoundation:x86_64+0x84fb3)
    #35 0x7fffa8005ebb in RunCurrentEventLoopInMode+0xef (HIToolbox:x86_64+0x30ebb)
    #36 0x7fffa8005bf8 in ReceiveNextEventCommon+0xb7 (HIToolbox:x86_64+0x30bf8)
    #37 0x7fffa8005b25 in _BlockUntilNextEventMatchingListInModeWithFilter+0x46 (HIToolbox:x86_64+0x30b25)
    #38 0x7fffa659aa03 in _DPSNextEvent+0x45f (AppKit:x86_64+0x46a03)
    #39 0x7fffa6d167ed in -[NSApplication(NSEvent) _nextEventMatchingEventMask:untilDate:inMode:dequeue:]+0xaeb (AppKit:x86_64+0x7c27ed)
    #40 0x122e4d842 in __71-[BrowserCrApplication nextEventMatchingMask:untilDate:inMode:dequeue:]_block_invoke chrome_browser_application_mac.mm:229
    #41 0x121c3d1b9 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xb4221b9)
    #42 0x122e4d3da in -[BrowserCrApplication nextEventMatchingMask:untilDate:inMode:dequeue:] chrome_browser_application_mac.mm:228
    #43 0x7fffa658f38a in -[NSApplication run]+0x39d (AppKit:x86_64+0x3b38a)
    #44 0x121c515da in base::MessagePumpNSApplication::DoRun(base::MessagePump::Delegate*) message_pump_mac.mm:691
    #45 0x121c4d1b8 in base::MessagePumpCFRunLoopBase::Run(base::MessagePump::Delegate*) message_pump_mac.mm:149
    #46 0x121b6181b in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
thread_controller_with_message_pump_impl.cc:460
    #47 0x121aa788b in base::RunLoop::Run() run_loop.cc:131
    #48 0x122173261 in ChromeBrowserMainParts::MainMessageLoopRun(int*) chrome_browser_main.cc:1736
    #49 0x11b04aef9 in content::BrowserMainLoop::RunMainMessageLoopParts() browser_main_loop.cc:970
    #50 0x11b0504c1 in content::BrowserMainRunnerImpl::Run() browser_main_runner_impl.cc:150
    #51 0x11b04276c in content::BrowserMain(content::MainFunctionParams const&) browser_main.cc:47
    #52 0x121884175 in content::ContentMainRunnerImpl::RunBrowser(content::MainFunctionParams&, bool) content_main_runner_impl.cc:555
    #53 0x1218834c3 in content::ContentMainRunnerImpl::Run(bool) content_main_runner_impl.cc:926
    #54 0x12188025b in content::RunContentProcess(content::ContentMainParams const&, content::ContentMainRunner*) content_main.cc:372
    #55 0x12188091c in content::ContentMain(content::ContentMainParams const&) content_main.cc:398
    #56 0x116820a35 in ChromeMain chrome_main.cc:141
    #57 0x10c765eff in main chrome_exe_main_mac.cc:114
    #58 0x7fffbe6ef234 in start+0x0 (libdyld.dylib:x86_64+0x5234)

0x6100002d0d28 is located 24 bytes to the left of 192-byte region [0x6100002d0d40,0x6100002d0e00)
allocated by thread T0 here:
    #0 0x10c949d30  (libclang_rt.asan_osx_dynamic.dylib:x86_64+0x45d30)
    #1 0x1219a5387 in operator new(unsigned long) new.cpp:67
    #2 0x12d84a6f6 in std::__1::vector<views::ViewModelBase::Entry, std::__1::allocator<views::ViewModelBase::Entry>
>::insert(std::__1::__wrap_iter<views::ViewModelBase::Entry const*>, views::ViewModelBase::Entry const&) __split_buffer:318
    #3 0x12d84b032 in views::ViewModelBase::AddUnsafe(views::View*, int) view_model.cc:74
    #4 0x12e8f7c82 in TabStrip::AddTabAt(int, TabRendererData, bool) tab_strip.cc:1271
    #5 0x12e860e4d in BrowserTabStripController::AddTab(content::WebContents*, int, bool) browser_tab_strip_controller.cc:775
    #6 0x12e865a40 in BrowserTabStripController::OnTabStripModelChanged(TabStripModel*, TabStripModelChange const&, TabStripSelectionChange const&)
browser_tab_strip_controller.cc:639
    #7 0x12dd6d30e in TabStripModel::InsertWebContentsAtImpl(int, std::__1::unique_ptr<content::WebContents, std::__1::default_delete<content::WebContents> >, int,
base::Optional<tab_groups::TabGroupId>) tab_strip_model.cc:1726
    #8 0x12dd6c6bb in TabStripModel::InsertWebContentsAt(int, std::__1::unique_ptr<content::WebContents, std::__1::default_delete<content::WebContents> >, int,
base::Optional<tab_groups::TabGroupId>) tab_strip_model.cc:379
    #9 0x12dbc17b9 in chrome::AddRestoredTab(Browser*, std::__1::vector<sessions::SerializedNavigationEntry, std::__1::allocator<sessions::SerializedNavigationEntry> >
const&, int, int, std::__1::basic_string<char, std::__1::char_traits<char>, std::__1::allocator<char> > const&, base::Optional<tab_groups::TabGroupId>, bool, bool,
base::TimeTicks, content::SessionStorageNamespace*, sessions::SerializedUserAgentOverride const&, bool) browser_tabrestore.cc:160
    #10 0x12321d0a5 in SessionRestoreImpl::RestoreTab(sessions::SessionTab const&, Browser*, std::__1::vector<SessionRestoreDelegate::RestoredTab,
std::__1::allocator<SessionRestoreDelegate::RestoredTab> >*, int, bool, base::TimeTicks) session_restore.cc:624
    #11 0x1232194a9 in SessionRestoreImpl::RestoreTabsToBrowser(sessions::SessionWindow const&, Browser*, int,
std::__1::vector<SessionRestoreDelegate::RestoredTab, std::__1::allocator<SessionRestoreDelegate::RestoredTab> >*) session_restore.cc:587
    #12 0x12321721e in SessionRestoreImpl::ProcessSessionWindows(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> > > >*, SessionID, std::__1::vector<SessionRestoreDelegate::RestoredTab,
std::__1::allocator<SessionRestoreDelegate::RestoredTab> >*) session_restore.cc:448
    #13 0x123216457 in SessionRestoreImpl::OnGotSession(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> > > >, SessionID) session_restore.cc:349
    #14 0x1232169b9 in void base::internal::FunctorTraits<void (SessionRestoreImpl::*)(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> > > >, SessionID), void>::Invoke<void (SessionRestoreImpl::*)
(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID),
base::WeakPtr<SessionRestoreImpl>, std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID>(void (SessionRestoreImpl::*)
(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID),
base::WeakPtr<SessionRestoreImpl>&&, std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >&&, SessionID&&) bind_internal.h:498
    #15 0x12322c87d in SessionService::OnGotSessionCommands(base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>, std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >) callback.h:101
    #16 0x123230dfe in void base::internal::FunctorTraits<void (SessionService::*)(base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow,
std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>, std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >), void>::Invoke<void (SessionService::*)(base::OnceCallback<void
(std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>,
std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >), base::WeakPtr<SessionService>,
base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>,
std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >>(void (SessionService::*)
(base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>,
```

```
std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >), base::WeakPtr<SessionService>&&,
base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionWindow, std::__1::default_delete<sessions::SessionWindow> > > >, SessionID)>&&,
std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >&&) bind_internal.h:498
    #17 0x11ce1d1dd in void base::internal::ReplyAdapter<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >, std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >(base::OnceCallback<void (std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >)>, std::__1::unique_ptr<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >, std::__1::default_delete<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > > >*) callback.h:101
    #18 0x11ce1d769 in base::internal::Invoker<base::internal::BindState<void (*)(base::OnceCallback<void
(std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >)>,
std::__1::unique_ptr<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >,
std::__1::default_delete<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > > >*), base::OnceCallback<void
(std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> >,
std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand, std::__1::default_delete<sessions::SessionCommand> > > >)>,
base::internal::OwnedWrapper<std::__1::unique_ptr<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > >, std::__1::default_delete<std::__1::vector<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> >, std::__1::allocator<std::__1::unique_ptr<sessions::SessionCommand,
std::__1::default_delete<sessions::SessionCommand> > > > >, void
()>::RunOnce(base::internal::BindStateBase*) bind_internal.h:393
    #19 0x121bb8316 in base::(anonymous namespace)::PostTaskAndReplyRelay::RunReply(base::(anonymous namespace)::PostTaskAndReplyRelay) callback.h:101
    #20 0x121bb8558 in base::internal::Invoker<base::internal::BindState<void (*)(base::(anonymous namespace)::PostTaskAndReplyRelay), base::(anonymous
namespace)::PostTaskAndReplyRelay>, void ()>::RunOnce(base::internal::BindStateBase*) bind_internal.h:393
    #21 0x121b21b15 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) callback.h:101
    #22 0x121b5fe30 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
thread_controller_with_message_pump_impl.cc:351
    #23 0x121b5f5d7 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork() thread_controller_with_message_pump_impl.cc:264
    #24 0x121c4fc18 in ___ZN4base24MessagePumpCFRunLoopBase13RunWorkSourceEPv_block_invoke message_pump_mac.mm:358
    #25 0x121c3d1b9 in base::mac::CallWithEHFrame(void () block_pointer)+0x9 (Chromium Framework:x86_64+0xb4221b9)
    #26 0x121c4e365 in base::MessagePumpCFRunLoopBase::RunWorkSource(void*) message_pump_mac.mm:334
    #27 0x7fffa8ac6e50 in __CFRUNLOOP_IS_CALLING_OUT_TO_A_SOURCE0_PERFORM_FUNCTION__+0x10 (CoreFoundation:x86_64+0xa4e50)
    #28 0x7fffa8aa80cb in __CFRunLoopDoSources0+0x22b (CoreFoundation:x86_64+0x860cb)
    #29 0x7fffa8aa75b5 in __CFRunLoopRun+0x3a5 (CoreFoundation:x86_64+0x855b5)

SUMMARY: AddressSanitizer: heap-buffer-overflow view_model.h:81 in TabStrip::SetSelection(ui::ListSelectionModel const&)
Shadow bytes around the buggy address:
  0x1c200005a150: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a160: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a170: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a180: fa fa fa fa fa fa fa fa fd fd fd fd fd fd fd fd
  0x1c200005a190: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fa
=>0x1c200005a1a0: fa fa fa fa fa[fa]fa fa 00 00 00 00 00 00 fc fc
  0x1c200005a1b0: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
  0x1c200005a1c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a1d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a1e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x1c200005a1f0: 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
```

poc.html
176 bytes View Download

screen.mov
2.6 MB View Download

Comment 1 by sheriffbot on Tue, Feb 2, 2021, 5:18 PM EST Project Member

Labels: external_security_report

Comment 2 by tsepez@chromium.org on Tue, Feb 2, 2021, 6:18 PM EST Project Member

Status: Assigned (was: Unconfirmed)
Owner: connily@chromium.org
Labels: Security_Impact-Head Security_Severity-High OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows OS-Lacros Pri-1
Components: UI>Browser>TabStrip

Perhaps similar to ~~https://crbug.com/1173260~~ but a different stack trace. Per the other bug, setting sev-high (not sev-critical) due to interaction required.

Comment 3 by sheriffbot on Wed, Feb 3, 2021, 12:54 PM EST Project Member

Labels: M-90 Target-90

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

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

Comment 4 by sheriffbot on Wed, Feb 3, 2021, 1:19 PM EST Project Member

Labels: ReleaseBlock-Stable

This is a serious security regression. If you are not able to fix this quickly, please revert the change that introduced it.

If this doesn't affect a release branch, or has not been properly classified for severity, please update the Security_Impact or Security_Severity labels, and remove the ReleaseBlock label. To disable this altogether, apply ReleaseBlock-NA.

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

Comment 5 by connily@chromium.org on Wed, Feb 3, 2021, 2:23 PM EST Project Member

Owner: tbergquist@chromium.org
Cc: connily@chromium.org

+Taylor has kindly agreed to help take a look here, as I'm not sure I can address these in a reasonable timeframe.

Taylor, these all look pretty similar to each other and to https://bugs.chromium.org/p/chromium/issues/detail?id=1151799, but with slightly different repros and stack traces. Please feel free to grab some time with me to go over them, or just chat asynchronously.

Thank you!!

Comment 6 by bugdroid on Wed, Feb 3, 2021, 8:40 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/cb3dde3e0e96f738f88a7a219c716b259baa83be

commit cb3dde3e0e96f738f88a7a219c716b259baa83be
Author: Taylor Bergquist <tbergquist@chromium.org>
Date: Thu Feb 04 01:36:46 2021

Fix crash trying to select an already closed tab on header drag completion.

~~Bug: 1173702~~
Change-Id: I9a7a668f89f16a89580fb7de61d19d0ae65ec9e0
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2673164
Reviewed-by: Connie Wan <connily@chromium.org>
Commit-Queue: Taylor Bergquist <tbergquist@chromium.org>
Cr-Commit-Position: refs/heads/master@{#850395}

[modify] https://crrev.com/cb3dde3e0e96f738f88a7a219c716b259baa83be/chrome/browser/ui/views/tabs/tab_drag_controller.cc

Comment 7 by chrom...@gmail.com on Thu, Feb 4, 2021, 12:01 AM EST

I verified the fix on Chromium 90.0.4408.0 refs/heads/master@{#850437} on MacOS.

Taylor, thanks for the quick fix!

screen.mov
2.7 MB View Download

Comment 8 by tbergquist@chromium.org on Thu, Feb 4, 2021, 7:45 PM EST Project Member

Status: Verified (was: Assigned)

Excellent! Glad it all worked out in the end

Comment 9 by sheriffbot on Fri, Feb 5, 2021, 12:43 PM EST Project Member

Labels: reward-topanel

Comment 10 by sheriffbot on Fri, Feb 5, 2021, 1:57 PM EST Project Member

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

Comment 11 by chrom...@gmail.com on Tue, Feb 16, 2021, 5:26 PM EST

Does this change need a merge to stable M88?

Comment 12 by tbergquist@google.com on Wed, Feb 17, 2021, 5:09 PM EST Project Member

Labels: Security_Impact-Stable

I believe it does affect stable, and 89. I think these are the labels to add to get security input on whether a merge is needed?

Comment 13 by chrom...@gmail.com on Wed, Feb 17, 2021, 5:41 PM EST

I think you should add Merge-Request-89 and Merge-Request-88 labels to get the security team's attention. (Please remove Security_Impact-Head label as this bug affects stable).

Comment 14 by adetaylor@google.com on Wed, Feb 17, 2021, 6:16 PM EST Project Member

Labels: -Security_Impact-Head Merge-Approved-89 Merge-Request-88

Sheriffbot would add the merge requests automatically now that #c12 has been done - thanks! However, to expedite matters, I'll just go ahead and approve merge to M89 - branch 4389.

Comment 15 by amyressler@google.com on Wed, Feb 17, 2021, 7:12 PM EST Project Member

Labels: -reward-topanel reward-unpaid reward-7500

*** 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 16 by amyressler@google.com on Wed, Feb 17, 2021, 7:25 PM EST Project Member

Hi, Khalil. The VRP Panel has decided to award you $7,500 for this report as the vulnerability does not give the attacker great control, but also the user interaction/gesture required is quite high. Thank you for your submission and nice work!

Comment 17 by chrom...@gmail.com on Wed, Feb 17, 2021, 7:29 PM EST

Great! It's a nice reward! - Thank you so much!

Comment 18 by bugdroid on Wed, Feb 17, 2021, 7:59 PM EST Project Member

Labels: -merge-approved-89 merge-merged-89 merge-merged-4389

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/6308a43c518dcb3c5eb1b0ff4c618c342a510b55

commit 6308a43c518dcb3c5eb1b0ff4c618c342a510b55
Author: Taylor Bergquist <tbergquist@chromium.org>
Date: Thu Feb 18 00:58:55 2021

Fix crash trying to select an already closed tab on header drag completion.

(cherry picked from commit cb3dde3e0e96f738f88a7a219c716b259baa83be)

~~Bug: 1173799~~
Change-Id: I9a7a668f89f16a89580fb7de61d19d0ae65ec9e0
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2673164
Reviewed-by: Connie Wan <connily@chromium.org>
Commit-Queue: Taylor Bergquist <tbergquist@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#850395}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2702758
Auto-Submit: Taylor Bergquist <tbergquist@chromium.org>
Commit-Queue: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Bot-Commit: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Cr-Commit-Position: refs/branch-heads/4389@{#1162}
Cr-Branched-From: 9251c5db2b6d5a59fe4eac7aafa5fed37c139bb7-refs/heads/master@{#843830}

[modify] https://crrev.com/6308a43c518dcb3c5eb1b0ff4c618c342a510b55/chrome/browser/ui/views/tabs/tab_drag_controller.cc

Comment 19 by awhalley@google.com on Fri, Feb 19, 2021, 5:34 PM EST Project Member

Labels: -reward-unpaid reward-inprocess

Comment 20 by adetaylor@google.com on Fri, Feb 26, 2021, 1:08 PM EST Project Member

Labels: Release-0-M89

Comment 21 by adetaylor@google.com on Fri, Feb 26, 2021, 4:44 PM EST Project Member

Labels: -Merge-Request-88 Merge-Rejected-88

Not merging to M88 - no further releases planned.

Comment 22 by asumaneev@google.com on Mon, Mar 1, 2021, 4:48 PM EST Project Member

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

Marking as not applicable for LTS since introducing code landed after M86 (same as ~~https://crbug.com/1163845~~).

Comment 23 by adetaylor@google.com on Mon, Mar 1, 2021, 7:26 PM EST Project Member

Labels: CVE-2021-21161 CVE_description-missing

Comment 24 by amyressler@google.com on Tue, Mar 9, 2021, 12:58 PM EST Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 25 by sheriffbot on Fri, May 14, 2021, 1:51 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