☆ Starred by 5 users

Owner:	sky@chromium.org
CC:	danakj@chromium.org adetaylor@chromium.org thomasanderson@chromium.org rsleevi@chromium.org gab@chromium.org
Status:	Fixed *(Closed)*
Components:	Internals>Aura Internals>Views>Desktop
Modified:	Aug 31, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Security_Severity-Medium
Hotlist-Merge-Approved
reward-7500
allpublic
reward-inprocess
Via-Wizard-Security
CVE_description-submitted
Target-90
M-91
Target-91
external_security_report
LTS-Security-90
LTS-Security-NotApplicable-90
Release-0-M92
CVE-2021-30579

Issue 1207277: Security: heap-use-after-free in BrowserView::ProcessFullscreen

Reported by merc....@gmail.com on Mon, May 10, 2021, 3:44 AM EDT

Code

UserAgent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36

Steps to reproduce the problem:
1. download asan-linux-release-880812.zip and poc.html, unzip chrome.
2. start a server at floder of poc.html : python -m SimpleHTTPServer 8605
3. ./asan-linux-release-880812/chrome http://127.0.0.1:8605/poc.html about:blank
4. click the button, then drag and drop the first tab repeatedly

What is the expected behavior?

What went wrong?
This is similar to ~~Issue 1197436~~, but with different UAF positions, so I submit a new Issue. It seems that 1197436 didn't fix this problem completely. In function `BrowserView::ProcessFullscreen` at line 3335, `frame_->SetFullscreen(fullscreen);` is called, it will finally call `Drag` and could delete tab, but there are no weak_ptr check for that, so UAF occurs at line 3350.

``` chrome/browser/ui/views/frame/browser_view.cc
```
3335   frame_->SetFullscreen(fullscreen); // call Drag, delete tab
3336
3337 #if !defined(OS_MAC)
3338   // On Mac, the pre-fullscreen bounds must be restored after an asynchronous
3339   // transition out of the fullscreen workspace; see http://crbug.com/1039874
3340   if (!fullscreen && restore_pre_fullscreen_bounds_callback_)
3341   std::move(restore_pre_fullscreen_bounds_callback_).Run();
3342 #endif // !OS_MAC
3343
3344   // Enable immersive before the browser refreshes its list of enabled commands.
3345   const bool should_stay_in_immersive =
3346   !fullscreen &&
3347   immersive_mode_controller_->ShouldStayImmersiveAfterExitingFullscreen();
3348   // Never use immersive in locked fullscreen as it allows the user to exit the
3349   // locked mode.
3350   if (platform_util::IsBrowserLockedFullscreen(browser_.get())) { // use freed memory without weak_ptr check.
```

```

```
=================================================================
==13980==ERROR: AddressSanitizer: heap-use-after-free on address 0x61c0002f7cf8 at pc 0x559d55781bf5 bp 0x7fff33dc5d30 sp 0x7fff33dc5d28
READ of size 8 at 0x61c0002f7cf8 thread T0 (chrome)
    #0 0x559d55781bf4 in get buildtools/third_party/libc++/trunk/include/memory:1569:19
    #1 0x559d55781bf4 in BrowserView::ProcessFullscreen(bool, GURL const&, ExclusiveAccessBubbleType, long) chrome/browser/ui/views/frame/browser_view.cc:3350:57
    #2 0x559d55781f5f in EnterFullscreen chrome/browser/ui/views/frame/browser_view.cc:1377:3
    #3 0x559d55781f5f in non-virtual thunk to BrowserView::EnterFullscreen(GURL const&, ExclusiveAccessBubbleType, long)
chrome/browser/ui/views/frame/browser_view.cc
    #4 0x559d550c542c in FullscreenController::EnterFullscreenModeInternal(FullscreenController::FullscreenInternalOption, content::RenderFrameHost*, long)
```

```
chrome/browser/ui/exclusive_access/fullscreen_controller.cc:407:42
    #5 0x559d550c49d1 in FullscreenController::EnterFullscreenModeForTab(content::RenderFrameHost*, long)
chrome/browser/ui/exclusive_access/fullscreen_controller.cc:164:5
    #6 0x559d42f7220e in content::WebContentsImpl::EnterFullscreenMode(content::RenderFrameHostImpl*, blink::mojom::FullscreenOptions const&)
content/browser/web_contents/web_contents_impl.cc:3185:16
    #7 0x559d42a7380b in content::RenderFrameHostImpl::EnterFullscreen(mojo::InlinedStructPtr<blink::mojom::FullscreenOptions>, base::OnceCallback<void (bool)>)
content/browser/renderer_host/render_frame_host_impl.cc:5154:14
    #8 0x559d3fadd55f in blink::mojom::LocalFrameHostStubDispatch::AcceptWithResponder(blink::mojom::LocalFrameHost*, mojo::Message*,
std::__1::unique_ptr<mojo::MessageReceiverWithStatus, std::__1::default_delete<mojo::MessageReceiverWithStatus> >)
gen/third_party/blink/public/mojom/frame/frame.mojom.cc:6625:13
    #9 0x559d4bce3b3f in mojo::InterfaceEndpointClient::HandleValidatedMessage(mojo::Message*) mojo/public/cpp/bindings/lib/interface_endpoint_client.cc:824:56
    #10 0x559d4bcf4dfa in mojo::MessageDispatcher::Accept(mojo::Message*) mojo/public/cpp/bindings/lib/message_dispatcher.cc:48:24
    #11 0x559d4d5d8759 in IPC::(anonymous namespace)::ChannelAssociatedGroupController::AcceptOnProxyThread(mojo::Message) ipc/ipc_mojo_bootstrap.cc:949:24
    #12 0x559d4d5d0f24 in Invoke<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), scoped_refptr<IPC::(anonymous
namespace)::ChannelAssociatedGroupController>, mojo::Message> base/bind_internal.h:509:12
    #13 0x559d4d5d0f24 in MakeItSo<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), scoped_refptr<IPC::(anonymous
namespace)::ChannelAssociatedGroupController>, mojo::Message> base/bind_internal.h:648:12
    #14 0x559d4d5d0f24 in RunImpl<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), std::tuple<scoped_refptr<IPC::(anonymous
namespace)::ChannelAssociatedGroupController>, mojo::Message>, 0, 1> base/bind_internal.h:721:12
    #15 0x559d4d5d0f24 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*)
base/bind_internal.h:690:12
    #16 0x559d4a364660 in Run base/callback.h:98:12
    #17 0x559d4a364660 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:178:33
    #18 0x559d4a39e4e6 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:357:25
    #19 0x559d4a39dcc4 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:270:36
    #20 0x559d4a260789 in HandleDispatch base/message_loop/message_pump_glib.cc:374:46
    #21 0x559d4a260789 in base::(anonymous namespace)::WorkSourceDispatch(_GSource*, int (*)(void*), void*) base/message_loop/message_pump_glib.cc:124:43
    #22 0x7ff32bde2fbc in g_main_context_dispatch (/lib/x86_64-linux-gnu/libglib-2.0.so.0+0x51fbc)

0x61c0002f7cf8 is located 1144 bytes inside of 1752-byte region [0x61c0002f7880,0x61c0002f7f58)
freed by thread T0 (chrome) here:
    #0 0x559d3ce66a2d in operator delete(void*) /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_new_delete.cpp:160:3
    #1 0x559d545351e9 in views::View::~View() ui/views/view.cc:240:9
    #2 0x559d560d9e45 in ~DesktopLinuxBrowserFrameView chrome/browser/ui/views/frame/desktop_linux_browser_frame_view.cc:24:61
    #3 0x559d560d9e45 in DesktopLinuxBrowserFrameView::~DesktopLinuxBrowserFrameView()
chrome/browser/ui/views/frame/desktop_linux_browser_frame_view.cc:24:61
    #4 0x559d546359d3 in operator() buildtools/third_party/libc++/trunk/include/memory:1335:5
    #5 0x559d546359d3 in reset buildtools/third_party/libc++/trunk/include/memory:1596:7
    #6 0x559d546359d3 in ~unique_ptr buildtools/third_party/libc++/trunk/include/memory:1550:19
    #7 0x559d546359d3 in ~NonClientView ui/views/window/non_client_view.cc:154:1
    #8 0x559d546359d3 in views::NonClientView::~NonClientView() ui/views/window/non_client_view.cc:150:33
    #9 0x559d54538761 in operator() buildtools/third_party/libc++/trunk/include/memory:1335:5
    #10 0x559d54538761 in reset buildtools/third_party/libc++/trunk/include/memory:1596:7
    #11 0x559d54538761 in ~unique_ptr buildtools/third_party/libc++/trunk/include/memory:1550:19
    #12 0x559d54538761 in views::View::DoRemoveChildView(views::View*, bool, bool, views::View*) ui/views/view.cc:2596:1
    #13 0x559d54538ad8 in views::View::RemoveAllChildViews(bool) ui/views/view.cc:309:5
    #14 0x559d545b285a in DestroyRootView ui/views/widget/widget.cc:1645:15
    #15 0x559d545b285a in views::Widget::~Widget() ui/views/widget/widget.cc:188:3
    #16 0x559d557a0c7d in BrowserFrame::~BrowserFrame() chrome/browser/ui/views/frame/browser_frame.cc:79:31
    #17 0x559d546997bf in views::DesktopNativeWidgetAura::~DesktopNativeWidgetAura() ui/views/widget/desktop_aura/desktop_native_widget_aura.cc
    #18 0x559d558ffab6 in ~DesktopBrowserFrameAuraLinux chrome/browser/ui/views/frame/desktop_browser_frame_aura_linux.cc:30:64
    #19 0x559d558ffab6 in DesktopBrowserFrameAuraLinux::~DesktopBrowserFrameAuraLinux()
chrome/browser/ui/views/frame/desktop_browser_frame_aura_linux.cc:30:63
    #20 0x559d5468904b in views::DesktopWindowTreeHostLinux::OnClosed() ui/views/widget/desktop_aura/desktop_window_tree_host_linux.cc:279:34
    #21 0x559d546d24b3 in views::DesktopWindowTreeHostPlatform::CloseNow() ui/views/widget/desktop_aura/desktop_window_tree_host_platform.cc:318:22
    #22 0x559d546dbf64 in Invoke<void (views::DesktopWindowTreeHostPlatform::*)(), base::WeakPtr<views::DesktopWindowTreeHostPlatform>>
base/bind_internal.h:509:12
    #23 0x559d546dbf64 in MakeItSo<void (views::DesktopWindowTreeHostPlatform::*)(), base::WeakPtr<views::DesktopWindowTreeHostPlatform>>
base/bind_internal.h:668:5
    #24 0x559d546dbf64 in RunImpl<void (views::DesktopWindowTreeHostPlatform::*)(), std::tuple<base::WeakPtr<views::DesktopWindowTreeHostPlatform> >, 0>
base/bind_internal.h:721:12
    #25 0x559d546dbf64 in base::internal::Invoker<base::internal::BindState<void (views::DesktopWindowTreeHostPlatform::*)(),
base::WeakPtr<views::DesktopWindowTreeHostPlatform> >, void ()>::RunOnce(base::internal::BindStateBase*) base/bind_internal.h:690:12
    #26 0x559d4a364660 in Run base/callback.h:98:12
    #27 0x559d4a364660 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:178:33
    #28 0x559d4a39e4e6 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:357:25
    #29 0x559d4a39dcc4 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:270:36
    #30 0x559d4a25fa10 in base::MessagePumpGlib::Run(base::MessagePump::Delegate*) base/message_loop/message_pump_glib.cc:404:48
    #31 0x559d4a39f797 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:463:12
    #32 0x559d4a2e2ff1 in base::RunLoop::Run(base::Location const&) base/run_loop.cc:134:14
    #33 0x559d4e7e6d04 in ui::X11WholeScreenMoveLoop::RunMoveLoop(bool, scoped_refptr<ui::X11Cursor>, scoped_refptr<ui::X11Cursor>)
ui/base/x/x11_whole_screen_move_loop.cc:196:12
    #34 0x559d546d8207 in views::DesktopWindowTreeHostPlatform::RunMoveLoop(gfx::Vector2d const&, views::Widget::MoveLoopSource,
views::Widget::MoveLoopEscapeBehavior) ui/views/widget/desktop_aura/desktop_window_tree_host_platform.cc:572:47
    #35 0x559d55e1ac83 in TabDragController::RunMoveLoop(gfx::Vector2d const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1487:61
    #36 0x559d55e1fa3f in TabDragController::DetachIntoNewBrowserAndRunMoveLoop(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1451:3
    #37 0x559d55e1d280 in TabDragController::DragBrowserToNewTabStrip(TabDragContext*, gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:889:5
    #38 0x559d55e1b5b5 in TabDragController::ContinueDragging(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:855:9
    #39 0x559d55e14935 in TabDragController::Drag(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:617:7
    #40 0x559d55e1c4a5 in TabDragController::OnWidgetBoundsChanged(views::Widget*, gfx::Rect const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:701:3
    #41 0x559d545c3656 in views::Widget::OnNativeWidgetSizeChanged(gfx::Size const&) ui/views/widget/widget.cc:1327:14
    #42 0x559d546a2ced in OnHostResized ui/views/widget/desktop_aura/desktop_native_widget_aura.cc:1292:28
    #43 0x559d546a2ced in non-virtual thunk to views::DesktopNativeWidgetAura::OnHostResized(aura::WindowTreeHost*)
ui/views/widget/desktop_aura/desktop_native_widget_aura.cc
    #44 0x559d4fcd5590 in aura::WindowTreeHost::OnHostResizedInPixels(gfx::Size const&) ui/aura/window_tree_host.cc:468:14

previously allocated by thread T0 (chrome) here:
    #0 0x559d3ce661cd in operator new(unsigned long) /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_new_delete.cpp:99:3
    #1 0x559d558f709e in BrowserWindow::CreateBrowserWindow(std::__1::unique_ptr<Browser, std::__1::default_delete<Browser> >, bool, bool)
chrome/browser/ui/views/frame/browser_window_factory.cc:41:23
    #2 0x559d5500b6b3 in CreateBrowserWindow chrome/browser/ui/browser.cc:309:10
    #3 0x559d5500b6b3 in Browser::Browser(Browser::CreateParams const&) chrome/browser/ui/browser.cc:518:29
    #4 0x559d5500a0e6 in Browser::Create(Browser::CreateParams const&) chrome/browser/ui/browser.cc:440:14
    #5 0x559d55e259ca in TabDragController::CreateBrowserForDrag(TabDragContext*, gfx::Point const&, gfx::Vector2d*, std::__1::vector<gfx::Rect,
std::__1::allocator<gfx::Rect> >*) chrome/browser/ui/views/tabs/tab_drag_controller.cc:2134:22
    #6 0x559d55e1f694 in TabDragController::DetachIntoNewBrowserAndRunMoveLoop(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1406:22
    #7 0x559d55e1d280 in TabDragController::DragBrowserToNewTabStrip(TabDragContext*, gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:889:5
    #8 0x559d55e1b5b5 in TabDragController::ContinueDragging(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:855:9
    #9 0x559d55e14935 in TabDragController::Drag(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:617:7
    #10 0x559d55e506d0 in TabStrip::TabDragContextImpl::ContinueDrag(views::View*, ui::LocatedEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:456:25
    #11 0x559d55e5d733 in TabStrip::OnMouseDragged(ui::MouseEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:3739:3
    #12 0x559d545501ab in views::View::ProcessMouseDragged(ui::MouseEvent*) ui/views/view.cc:3009:9
    #13 0x559d4d73cb17 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #14 0x559d4d73a429 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #15 0x559d4d73a429 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #16 0x559d4d739cf1 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #17 0x559d4d739cf1 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #18 0x559d545a43a0 in views::internal::RootView::OnMouseDragged(ui::MouseEvent const&) ui/views/widget/root_view.cc:457:9
    #19 0x559d545c4500 in views::Widget::OnMouseEvent(ui::MouseEvent*) ui/views/widget/widget.cc:1440:22
    #20 0x559d4d73cb17 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #21 0x559d4d73a429 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #22 0x559d4d73a429 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #23 0x559d4d739cf1 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #24 0x559d4d739cf1 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #25 0x559d4fcbbead in ui::EventProcessor::OnEventFromSource(ui::Event*) ui/events/event_processor.cc:49:17
    #26 0x559d4fcd9c9f in ui::EventSource::DeliverEventToSink(ui::Event*) ui/events/event_source.cc:113:16
    #27 0x559d4fcd9943 in ui::EventSource::SendEventToSinkFromRewriter(ui::Event const*, ui::EventRewriter const*) ui/events/event_source.cc:138:12
    #28 0x559d5468deb7 in aura::WindowTreeHostPlatform::DispatchEvent(ui::Event*) ui/aura/window_tree_host_platform.cc:247:38
    #29 0x559d54688c56 in views::DesktopWindowTreeHostLinux::DispatchEvent(ui::Event*) ui/views/widget/desktop_aura/desktop_window_tree_host_linux.cc:273:29
    #30 0x559d4e7d10d3 in ui::X11Window::DispatchUiEvent(ui::Event*, x11::Event const&) ui/platform_window/x11/x11_window.cc:1195:34
    #31 0x559d4e7d03df in ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc:1141:3
    #32 0x559d4e7d12ec in non-virtual thunk to ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc
    #33 0x559d4d39c5da in ui::PlatformEventSource::DispatchEvent(ui::Event*) ui/events/platform/platform_event_source.cc:100:29
    #34 0x559d4d893c84 in ui::X11EventSource::OnEvent(x11::Event const&) ui/events/platform/x11/x11_event_source.cc:299:5

SUMMARY: AddressSanitizer: heap-use-after-free buildtools/third_party/libc++/trunk/include/memory:1569:19 in get
Shadow bytes around the buggy address:
  0x0c3880056f40: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056f50: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056f60: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056f70: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056f80: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x0c3880056f90: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd[fd]
  0x0c3880056fa0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056fb0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056fc0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056fd0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880056fe0: fd fd fd fd fd fd fd fd fd fd fd 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
==13980==ABORTING
```

Did this work before? N/A

Chrome version: 90.0.4430.93 Channel: stable
OS Version:
Flash Version:

poc.html
270 bytes View Download

Comment 1 by sheriffbot on Mon, May 10, 2021, 3:46 AM EDT Project Member
Labels: external_security_report

Comment 2 by rsleevi@chromium.org on Mon, May 10, 2021, 5:29 PM EDT Project Member
Cc: sky@chromium.org
Labels: Security_Severity-High Security_Impact-Head Needs-Feedback
Components: Internals>Views>Desktop

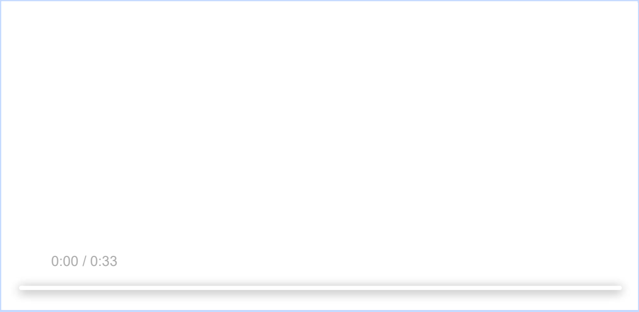
Could you possibly attach a screen recording? I'm having trouble reproducing this, both dragging the tab within window (e.g. between about blank) and as separate windows.

sky: I haven't confirmed yet, so not yet assigning to you, but CC'ing you because of the reported similarity to ~~Issue 1197436~~

Comment 3 by sky@chromium.org on Mon, May 10, 2021, 11:12 PM EDT Project Member
As the submitter mentions, this is another case that will need a weakref check. The core issues is the nested message loop that tab dragging spawns. While it may be possible to eliminate it on linux, it is not possible on windows (because win32 internals run the nested message loop, not chrome).

Comment 4 by merc....@gmail.com on Tue, May 11, 2021, 1:39 AM EDT
upload the screen recording:)

video.webm
7.1 MB View Download



Comment 5 by sheriffbot on Tue, May 11, 2021, 1:40 AM EDT Project Member

Cc: rsleevi@chromium.org
Labels: -Needs-Feedback

Thank you for providing more feedback. Adding the requester to the cc list.

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

Comment 6 by sheriffbot on Tue, May 11, 2021, 12:52 PM EDT Project Member

Labels: M-92 Target-92

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 7 by sheriffbot on Tue, May 11, 2021, 1:17 PM EDT 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 8 by sheriffbot on Tue, May 11, 2021, 1:27 PM EDT Project Member

Labels: -Pri-2 Pri-1

Setting Pri-1 to match security severity High. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

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

Comment 9 by sleevi@google.com on Tue, May 11, 2021, 2:01 PM EDT Project Member

Components: Internals>Aura

Comment 10 by rsleevi@chromium.org on Tue, May 11, 2021, 2:20 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Owner: sky@chromium.org
Cc: -sky@chromium.org

Thanks! It looks like the method I was using to test was a bit too CPU bound, which isn't surprising for timing issues, and was able to reproduce. Thanks also for the pointer to drag it outside the window bounds (e.g. vertically) rather than horizontally between tab order :)

Comment 11 by sky@chromium.org on Tue, May 11, 2021, 7:32 PM EDT Project Member

Please correct me if I'm wrong, but my understanding is this requires very finicky user interaction, and is very unlikely that a page could trigger this. If I have that right, is a P1 justified here? To be clear, I understand this should be fixed, but I'm trying to understand the urgency.

Comment 12 by rsleevi@chromium.org on Tue, May 11, 2021, 7:59 PM EDT Project Member

The priority is set based on the severity assessment. This would normally be Critical, due to where and how its crashing in the browser process, but this was downgraded to High because of the user interaction requirements.

Note that even if downgraded to Medium, we'd still target this as P-1 by sheriffbot.

Comment 13 by sky@chromium.org on Wed, May 12, 2021, 4:38 PM EDT Project Member

merc.ouc@gmail.com, what gn args do you compile with?

Comment 14 by sky@chromium.org on Wed, May 12, 2021, 4:40 PM EDT Project Member

Ryan, were you able to reproduce? If so, what gn args did you compile with?

Comment 15 by rsleevi@chromium.org on Wed, May 12, 2021, 4:58 PM EDT Project Member

sky: Yes. I just grabbed 880182 from our Linux ASAN Debug bot - https://commondatastorage.googleapis.com/chromium-browser-asan/index.html?prefix=linux-debug/asan-linux-debug-88 - and performed the steps in the video (Comment #4)

The builder is https://ci.chromium.org/p/chromium/builders/ci/Linux%20ASan%20LSan%20Builder and looking at the build, the GN args are

```
dcheck_always_on = true
is_asan = true
is_component_build = false
is_debug = false
is_lsan = true
symbol_level = 1
use_goma = true
```

Comment 16 by rsleevi@chromium.org on Wed, May 12, 2021, 5:09 PM EDT Project Member

Er, sorry, that may not be the exact builder (just realized the is_debug = false there). But yeah, I just grabbed our pre-existing builder.

Comment 17 by sky@chromium.org on Wed, May 12, 2021, 5:50 PM EDT Project Member

Thanks.

Comment 18 by sky@chromium.org on Wed, May 12, 2021, 6:47 PM EDT Project Member

What's the stack of the crash you're getting Ryan? I'm hitting a CHECK in tab_strip_model. When I make it so that doesn't hit, I can't seem to trigger this crash. I'm not confident the fix for the CHECK addresses the issue though. I would love deeper stacks than that pasted in the report.

Comment 19 by sky@chromium.org on Wed, May 12, 2021, 7:52 PM EDT Project Member

Status: Started (was: Assigned)

Comment 20 by merc....@gmail.com on Wed, May 12, 2021, 9:54 PM EDT

Hi sky, there is actually a CHECK fail in this poc, but it is a different crash to the UAF.(It seems that CHECK fail is not security bug). After click the button, you can drag the tab and don't release the mouse, when this tab become fullscreen, move the mouse to the origin position of the tab, CHECK failed occured, as shown in video. This UAF crash is triggered in the middle of fullscreen, the tab is closed by drag and drop(`Drag` function), which is called by the function `frame_->SetFullscreen(fullscreen);` , after that, the use of object bind to the tab will trigger UAF.
The drag and drop should occur at the same time as fullscreening, maybe you need to try many many times to repo it.

Comment 21 by sky@chromium.org on Fri, May 14, 2021, 10:37 AM EDT Project Member

I'm having a hard time reproducing. When fullscreen occurs, is the tab in a separate window, or in the original window?

merc.ous, if you compile with the flag enable_full_stack_frames_for_profiling=true can you still reproduce? Do you get longer stack frames?

Comment 22 by merc....@gmail.com on Sun, May 16, 2021, 9:50 PM EDT

Sorry for lately replying, I'll compile and try it today.

Comment 23 by merc....@gmail.com on Mon, May 17, 2021, 3:31 AM EDT

ASAN log with enable_full_stack_frames_for_profiling=true

```
=================================================================
==4842==ERROR: AddressSanitizer: heap-use-after-free on address 0x61c0002d2cf8 at pc 0x55791ed2d5b2 bp 0x7ffea87d4570 sp 0x7ffea87d4568
READ of size 8 at 0x61c0002d2cf8 thread T0 (chrome)
    #0 0x55791ed2d5b1 in get buildtools/third_party/libc++/trunk/include/memory:1569:19
    #1 0x55791ed2d5b1 in BrowserView::ProcessFullscreen(bool, GURL const&, ExclusiveAccessBubbleType, long)
chrome/browser/ui/views/frame/browser_view.cc:3350:57
    #2 0x55791ed2d91f in EnterFullscreen chrome/browser/ui/views/frame/browser_view.cc:1379:3
    #3 0x55791ed2d91f in non-virtual thunk to BrowserView::EnterFullscreen(GURL const&, ExclusiveAccessBubbleType, long)
chrome/browser/ui/views/frame/browser_view.cc
    #4 0x55791e67e4ac in FullscreenController::EnterFullscreenModeInternal(FullscreenController::FullscreenInternalOption, content::RenderFrameHost*, long)
chrome/browser/ui/exclusive_access/fullscreen_controller.cc:407:42
    #5 0x55791e67da51 in FullscreenController::EnterFullscreenModeForTab(content::RenderFrameHost*, long)
chrome/browser/ui/exclusive_access/fullscreen_controller.cc:164:5
    #6 0x55790c4c28ce in content::WebContentsImpl::EnterFullscreenMode(content::RenderFrameHostImpl*, blink::mojom::FullscreenOptions const&)
content/browser/web_contents/web_contents_impl.cc:3193:16
    #7 0x55790bfbe8ab in content::RenderFrameHostImpl::EnterFullscreen(mojo::InlinedStructPtr<blink::mojom::FullscreenOptions>, base::OnceCallback<void (bool)>)
content/browser/renderer_host/render_frame_host_impl.cc:5160:14
    #8 0x557909049222 in blink::mojom::LocalFrameHostStubDispatch::AcceptWithResponder(blink::mojom::LocalFrameHost*, mojo::Message*,
std::__1::unique_ptr<mojo::MessageReceiverWithStatus, std::__1::default_delete<mojo::MessageReceiverWithStatus> >)
gen/third_party/blink/public/mojom/frame/frame.mojom.cc:6735:13
    #9 0x55791522a22f in mojo::InterfaceEndpointClient::HandleValidatedMessage(mojo::Message*) mojo/public/cpp/bindings/lib/interface_endpoint_client.cc:829:56
    #10 0x55791523b7ba in mojo::MessageDispatcher::Accept(mojo::Message*) mojo/public/cpp/bindings/lib/message_dispatcher.cc:48:24
    #11 0x55791522e1b5 in mojo::InterfaceEndpointClient::HandleIncomingMessage(mojo::Message*) mojo/public/cpp/bindings/lib/interface_endpoint_client.cc:653:21
    #12 0x557916b21ad9 in IPC::(anonymous namespace)::ChannelAssociatedGroupController::AcceptOnProxyThread(mojo::Message) ipc/ipc_mojo_bootstrap.cc:949:24
    #13 0x557916b1a298 in Invoke<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), scoped_refptr<IPC::(anonymous
namespace)::ChannelAssociatedGroupController>, mojo::Message> base/bind_internal.h:509:12
    #14 0x557916b1a298 in MakeItSo<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), scoped_refptr<IPC::(anonymous
namespace)::ChannelAssociatedGroupController>, mojo::Message> base/bind_internal.h:648:12
    #15 0x557916b1a298 in RunImpl<void (IPC::(anonymous namespace)::ChannelAssociatedGroupController::*)(mojo::Message), std::tuple<scoped_refptr<IPC::
(anonymous namespace)::ChannelAssociatedGroupController>, mojo::Message>, 0, 1> base/bind_internal.h:721:12
    #16 0x557916b1a298 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*)
base/bind_internal.h:690:12
    #17 0x55791389c900 in Run base/callback.h:98:12
    #18 0x55791389c900 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:178:33
    #19 0x5579138d6c96 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:357:25
    #20 0x5579138d6474 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:270:36
    #21 0x5579137975c0 in base::MessagePumpGlib::Run(base::MessagePump::Delegate*) base/message_loop/message_pump_glib.cc:404:48
    #22 0x5579138d7dac in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:466:12
    #23 0x55791381afb1 in base::RunLoop::Run(base::Location const&) base/run_loop.cc:134:14
    #24 0x557917d37e34 in ui::X11WholeScreenMoveLoop::RunMoveLoop(bool, scoped_refptr<ui::X11Cursor>, scoped_refptr<ui::X11Cursor>)
ui/base/x/x11_whole_screen_move_loop.cc:196:12
    #25 0x55791dc82177 in views::DesktopWindowTreeHostPlatform::RunMoveLoop(gfx::Vector2d const&, views::Widget::MoveLoopSource,
views::Widget::MoveLoopEscapeBehavior) ui/views/widget/desktop_aura/desktop_window_tree_host_platform.cc:583:47
    #26 0x55791f3eb5f3 in TabDragController::RunMoveLoop(gfx::Vector2d const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1487:61
    #27 0x55791f3f033f in TabDragController::DetachIntoNewBrowserAndRunMoveLoop(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1451:3
    #28 0x55791f3edbf0 in TabDragController::DragBrowserToNewTabStrip(TabDragContext*, gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:889:5
    #29 0x55791f3ebf25 in TabDragController::ContinueDragging(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:855:9
    #30 0x55791f3e5361 in TabDragController::Drag(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:617:7
    #31 0x55791f420f50 in TabStrip::TabDragContextImpl::ContinueDrag(views::View*, ui::LocatedEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:456:25
    #32 0x55791f42d863 in TabStrip::OnMouseDragged(ui::MouseEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:3739:3
    #33 0x55791daf3abb in views::View::ProcessMouseDragged(ui::MouseEvent*) ui/views/view.cc:3014:9
    #34 0x557916c8b3e7 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #35 0x557916c88c39 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #36 0x557916c88c39 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #37 0x557916c88501 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #38 0x557916c88501 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #39 0x55791db49990 in views::internal::RootView::OnMouseDragged(ui::MouseEvent const&) ui/views/widget/root_view.cc:458:9
    #40 0x55791db6a070 in views::Widget::OnMouseEvent(ui::MouseEvent*) ui/views/widget/widget.cc:1445:22
    #41 0x557916c8b3e7 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #42 0x557916c88c39 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #43 0x557916c88c39 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #44 0x557916c88501 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #45 0x557916c88501 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #46 0x557919236efd in ui::EventProcessor::OnEventFromSource(ui::Event*) ui/events/event_processor.cc:49:17
    #47 0x5579192557ff in ui::EventSource::DeliverEventToSink(ui::Event*) ui/events/event_source.cc:113:16
    #48 0x5579192554a3 in ui::EventSource::SendEventToSinkFromRewriter(ui::Event const*, ui::EventRewriter const*) ui/events/event_source.cc:138:12
    #49 0x55791dc35f57 in aura::WindowTreeHostPlatform::DispatchEvent(ui::Event*) ui/aura/window_tree_host_platform.cc:247:38
    #50 0x55791dc308e6 in views::DesktopWindowTreeHostLinux::DispatchEvent(ui::Event*) ui/views/widget/desktop_aura/desktop_window_tree_host_linux.cc:273:29
    #51 0x557917d22473 in ui::X11Window::DispatchUiEvent(ui::Event*, x11::Event const&) ui/platform_window/x11/x11_window.cc:1195:34
    #52 0x557917d2177f in ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc:1141:3
    #53 0x557917d2268c in non-virtual thunk to ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc
    #54 0x5579168e6e04 in ui::PlatformEventSource::DispatchEvent(ui::Event*) ui/events/platform/platform_event_source.cc:100:29
    #55 0x557916de3284 in ui::X11EventSource::OnEvent(x11::Event const&) ui/events/platform/x11/x11_event_source.cc:299:5
    #56 0x5579078701ba in x11::Connection::DispatchEvent(x11::Event const&) ui/gfx/x/connection.cc:460:14
    #57 0x55790786f0d4 in ProcessNextEvent ui/gfx/x/connection.cc:511:3
    #58 0x55790786f0d4 in x11::Connection::Dispatch() ui/gfx/x/connection.cc:433:7
    #59 0x557916de1705 in ui::X11EventSource::DispatchXEvents() ui/events/platform/x11/x11_event_source.cc:156:25
    #60 0x557916df11cb in ui::(anonymous namespace)::XSourceDispatch(_GSource*, int (*)(void*), void*) ui/events/platform/x11/x11_event_watcher_glib.cc:42:15
    #61 0x7fcb2d71ae8d in g_main_context_dispatch (/lib/x86_64-linux-gnu/libglib-2.0.so.0+0x51e8d)

0x61c0002d2cf8 is located 1144 bytes inside of 1752-byte region [0x61c0002d2880,0x61c0002d2f58)
freed by thread T0 (chrome) here:
    #0 0x5579063d7fad in operator delete(void*) /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_new_delete.cpp:160:3
    #1 0x55791dad7f69 in views::View::~View() ui/views/view.cc:240:9
    #2 0x55791f6b2125 in ~DesktopLinuxBrowserFrameView chrome/browser/ui/views/frame/desktop_linux_browser_frame_view.cc:24:61
    #3 0x55791f6b2125 in DesktopLinuxBrowserFrameView::~DesktopLinuxBrowserFrameView()
chrome/browser/ui/views/frame/desktop_linux_browser_frame_view.cc:24:61
    #4 0x55791dbdce83 in operator() buildtools/third_party/libc++/trunk/include/memory:1335:5
    #5 0x55791dbdce83 in reset buildtools/third_party/libc++/trunk/include/memory:1596:7
    #6 0x55791dbdce83 in ~unique_ptr buildtools/third_party/libc++/trunk/include/memory:1550:19
    #7 0x55791dbdce83 in ~NonClientView ui/views/window/non_client_view.cc:154:1
    #8 0x55791dbdce83 in views::NonClientView::~NonClientView() ui/views/window/non_client_view.cc:150:33
    #9 0x55791dadb9d8 in operator() buildtools/third_party/libc++/trunk/include/memory:1335:5
    #10 0x55791dadb9d8 in reset buildtools/third_party/libc++/trunk/include/memory:1596:7
    #11 0x55791dadb9d8 in ~unique_ptr buildtools/third_party/libc++/trunk/include/memory:1550:19
    #12 0x55791dadb9d8 in views::View::DoRemoveChildView(views::View*, bool, bool, views::View*) ui/views/view.cc:2601:1
    #13 0x55791dadbdc8 in views::View::RemoveAllChildViews(bool) ui/views/view.cc:309:5
    #14 0x55791db5832a in DestroyRootView ui/views/widget/widget.cc:1650:15
    #15 0x55791db5832a in views::Widget::~Widget() ui/views/widget/widget.cc:189:3
    #16 0x55791ed4cbad in BrowserFrame::~BrowserFrame() chrome/browser/ui/views/frame/browser_frame.cc:83:31
    #17 0x55791dc41bff in views::DesktopNativeWidgetAura::~DesktopNativeWidgetAura() ui/views/widget/desktop_aura/desktop_native_widget_aura.cc
    #18 0x55791eeb39f6 in ~DesktopBrowserFrameAuraLinux chrome/browser/ui/views/frame/desktop_browser_frame_aura_linux.cc:30:64
    #19 0x55791eeb39f6 in DesktopBrowserFrameAuraLinux::~DesktopBrowserFrameAuraLinux()
chrome/browser/ui/views/frame/desktop_browser_frame_aura_linux.cc:30:63
    #20 0x55791dc30cdb in views::DesktopWindowTreeHostLinux::OnClosed() ui/views/widget/desktop_aura/desktop_window_tree_host_linux.cc:279:34
    #21 0x55791dc7c423 in views::DesktopWindowTreeHostPlatform::CloseNow() ui/views/widget/desktop_aura/desktop_window_tree_host_platform.cc:329:22
    #22 0x55791dc86a94 in Invoke<void (views::DesktopWindowTreeHostPlatform::*)(), base::WeakPtr<views::DesktopWindowTreeHostPlatform>>
base/bind_internal.h:509:12
    #23 0x55791dc86a94 in MakeItSo<void (views::DesktopWindowTreeHostPlatform::*)(), base::WeakPtr<views::DesktopWindowTreeHostPlatform>>
base/bind_internal.h:668:5
    #24 0x55791dc86a94 in RunImpl<void (views::DesktopWindowTreeHostPlatform::*)(), std::tuple<base::WeakPtr<views::DesktopWindowTreeHostPlatform> >, 0>
base/bind_internal.h:721:12
    #25 0x55791dc86a94 in base::internal::Invoker<base::internal::BindState<void (views::DesktopWindowTreeHostPlatform::*)(),
base::WeakPtr<views::DesktopWindowTreeHostPlatform> >, void ()>::RunOnce(base::internal::BindStateBase*) base/bind_internal.h:690:12
    #26 0x55791389c900 in Run base/callback.h:98:12
    #27 0x55791389c900 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) base/task/common/task_annotator.cc:178:33
    #28 0x5579138d6c96 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:357:25
    #29 0x5579138d6474 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWork()
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:270:36
    #30 0x5579137975c0 in base::MessagePumpGlib::Run(base::MessagePump::Delegate*) base/message_loop/message_pump_glib.cc:404:48
    #31 0x5579138d7f47 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:463:12
    #32 0x55791381afb1 in base::RunLoop::Run(base::Location const&) base/run_loop.cc:134:14
    #33 0x557917d37e34 in ui::X11WholeScreenMoveLoop::RunMoveLoop(bool, scoped_refptr<ui::X11Cursor>, scoped_refptr<ui::X11Cursor>)
ui/base/x/x11_whole_screen_move_loop.cc:196:12
    #34 0x55791dc82177 in views::DesktopWindowTreeHostPlatform::RunMoveLoop(gfx::Vector2d const&, views::Widget::MoveLoopSource,
views::Widget::MoveLoopEscapeBehavior) ui/views/widget/desktop_aura/desktop_window_tree_host_platform.cc:583:47
    #35 0x55791f3eb5f3 in TabDragController::RunMoveLoop(gfx::Vector2d const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1487:61
    #36 0x55791f3f033f in TabDragController::DetachIntoNewBrowserAndRunMoveLoop(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1451:3
    #37 0x55791f3edbf0 in TabDragController::DragBrowserToNewTabStrip(TabDragContext*, gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:889:5
    #38 0x55791f3ebf25 in TabDragController::ContinueDragging(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:855:9
    #39 0x55791f3e5361 in TabDragController::Drag(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:617:7
    #40 0x55791f3ece15 in TabDragController::OnWidgetBoundsChanged(views::Widget*, gfx::Rect const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:701:3
    #41 0x55791db691c6 in views::Widget::OnNativeWidgetSizeChanged(gfx::Size const&) ui/views/widget/widget.cc:1332:14
    #42 0x55791dc4b47d in OnHostResized ui/views/widget/desktop_aura/desktop_native_widget_aura.cc:1297:28
    #43 0x55791dc4b47d in non-virtual thunk to views::DesktopNativeWidgetAura::OnHostResized(aura::WindowTreeHost*)
ui/views/widget/desktop_aura/desktop_native_widget_aura.cc
    #44 0x55791925095a in aura::WindowTreeHost::OnHostResizedInPixels(gfx::Size const&) ui/aura/window_tree_host.cc:468:14

previously allocated by thread T0 (chrome) here:
    #0 0x5579063d774d in operator new(unsigned long) /b/s/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_new_delete.cpp:99:3
    #1 0x55791eeaaabe in BrowserWindow::CreateBrowserWindow(std::__1::unique_ptr<Browser, std::__1::default_delete<Browser> >, bool, bool)
chrome/browser/ui/views/frame/browser_window_factory.cc:41:23
    #2 0x55791e5c5c23 in CreateBrowserWindow chrome/browser/ui/browser.cc:310:10
    #3 0x55791e5c5c23 in Browser::Browser(Browser::CreateParams const&) chrome/browser/ui/browser.cc:519:29
    #4 0x55791e5c4656 in Browser::Create(Browser::CreateParams const&) chrome/browser/ui/browser.cc:441:14
    #5 0x55791f3f5fea in TabDragController::CreateBrowserForDrag(TabDragContext*, gfx::Point const&, gfx::Vector2d*, std::__1::vector<gfx::Rect,
std::__1::allocator<gfx::Rect> >*) chrome/browser/ui/views/tabs/tab_drag_controller.cc:2134:22
    #6 0x55791f3eff94 in TabDragController::DetachIntoNewBrowserAndRunMoveLoop(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:1406:22
    #7 0x55791f3edbf0 in TabDragController::DragBrowserToNewTabStrip(TabDragContext*, gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:889:5
    #8 0x55791f3ebf25 in TabDragController::ContinueDragging(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:855:9
    #9 0x55791f3e5361 in TabDragController::Drag(gfx::Point const&) chrome/browser/ui/views/tabs/tab_drag_controller.cc:617:7
    #10 0x55791f420f50 in TabStrip::TabDragContextImpl::ContinueDrag(views::View*, ui::LocatedEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:456:25
    #11 0x55791f42d863 in TabStrip::OnMouseDragged(ui::MouseEvent const&) chrome/browser/ui/views/tabs/tab_strip.cc:3739:3
    #12 0x55791daf3abb in views::View::ProcessMouseDragged(ui::MouseEvent*) ui/views/view.cc:3014:9
    #13 0x557916c8b3e7 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #14 0x557916c88c39 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #15 0x557916c88c39 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #16 0x557916c88501 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #17 0x557916c88501 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #18 0x55791db49990 in views::internal::RootView::OnMouseDragged(ui::MouseEvent const&) ui/views/widget/root_view.cc:458:9
    #19 0x55791db6a070 in views::Widget::OnMouseEvent(ui::MouseEvent*) ui/views/widget/widget.cc:1445:22
    #20 0x557916c8b3e7 in ui::EventHandler::OnEvent(ui::Event*) ui/events/event_handler.cc
    #21 0x557916c88c39 in DispatchEvent ui/events/event_dispatcher.cc:191:12
    #22 0x557916c88c39 in ui::EventDispatcher::ProcessEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:140:5
    #23 0x557916c88501 in DispatchEventToTarget ui/events/event_dispatcher.cc:84:14
    #24 0x557916c88501 in ui::EventDispatcherDelegate::DispatchEvent(ui::EventTarget*, ui::Event*) ui/events/event_dispatcher.cc:56:15
    #25 0x557919236efd in ui::EventProcessor::OnEventFromSource(ui::Event*) ui/events/event_processor.cc:49:17
    #26 0x5579192557ff in ui::EventSource::DeliverEventToSink(ui::Event*) ui/events/event_source.cc:113:16
    #27 0x5579192554a3 in ui::EventSource::SendEventToSinkFromRewriter(ui::Event const*, ui::EventRewriter const*) ui/events/event_source.cc:138:12
    #28 0x55791dc35f57 in aura::WindowTreeHostPlatform::DispatchEvent(ui::Event*) ui/aura/window_tree_host_platform.cc:247:38
    #29 0x55791dc308e6 in views::DesktopWindowTreeHostLinux::DispatchEvent(ui::Event*) ui/views/widget/desktop_aura/desktop_window_tree_host_linux.cc:273:29
    #30 0x557917d22473 in ui::X11Window::DispatchUiEvent(ui::Event*, x11::Event const&) ui/platform_window/x11/x11_window.cc:1195:34
    #31 0x557917d2177f in ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc:1141:3
    #32 0x557917d2268c in non-virtual thunk to ui::X11Window::DispatchEvent(ui::Event* const&) ui/platform_window/x11/x11_window.cc
    #33 0x5579168e6e04 in ui::PlatformEventSource::DispatchEvent(ui::Event*) ui/events/platform/platform_event_source.cc:100:29
    #34 0x557916de3284 in ui::X11EventSource::OnEvent(x11::Event const&) ui/events/platform/x11/x11_event_source.cc:299:5

SUMMARY: AddressSanitizer: heap-use-after-free buildtools/third_party/libc++/trunk/include/memory:1569:19 in get
Shadow bytes around the buggy address:
  0x0c3880052540: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880052550: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880052560: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880052570: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c3880052580: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
=>0x0c3880052590: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd[fd]
  0x0c38800525a0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c38800525b0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c38800525c0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c38800525d0: fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd fd
  0x0c38800525e0: fd fd fd fd fd fd fd fd fd fd fd 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
==4842==ABORTING
```

Comment 24 by sky@chromium.org on Tue, May 18, 2021, 11:53 AM EDT Project Member

Cc: thomasanderson@chromium.org gab@chromium.org

merc.ouc@gmail.com, thanks for the extra details.

+thomasanderson and +gab

AFAICT the only way this could happen is if we dispatch multiple mouse events at a time, without checking if we should quit the message loop between the mouse events. On Windows we dispatch a single mouse event, and then check if the loop should be quit. On Linux, we may dispatch multiple mouse events before checking the quit status:
https://source.chromium.org/chromium/chromium/src/+/main:ui/events/platform/x11/x11_event_watcher_glib.cc;drc=6531e02635a3f00784aacad5c641cd16532657e5;l=46 .

I could make TabDragController deal with this, or we could try to make linux behave like windows and dispatch a single mouse event before checking the exit status. Tom, is it possible to change the linux side? I favor making linux behave like windows as it is entirely possible this could bite other parts of the code base.

Comment 25 by gab@chromium.org on Tue, May 18, 2021, 12:04 PM EDT Project Member

I also prefer making Linux match Windows as honoring quits ASAP is key to the message pump contract.

Note: on Windows, it's *possible* to have multiple "sent messages" run before honoring a quit because "sent messages" are dispatched by the OS from within ::PeekMessage(), but that's exception. Input messages come in as "posted messages" and are dispatched one by one (Chrome's pump regains control between each one).
https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-peekmessagew

Comment 26 by thomasanderson@chromium.org on Tue, May 18, 2021, 2:04 PM EDT Project Member

Does ui::PlatformEventSource::StopCurrentEventStream() achieve what you're looking for? On X11, that will prevent further events from being dispatched (see "&& continue_stream_) [2]). The only issue is that StopCurrentEventStream() is not implemented for WaylandEventSource [3], so you'll have to implement it if you want this to work with wayland.

[1] https://source.chromium.org/chromium/chromium/src/+/main:ui/events/platform/platform_event_source.h;drc=a60b6a5add634cf45ee443f226a04e99b0f6af42;l=73
[2] https://source.chromium.org/chromium/chromium/src/+/main:ui/events/platform/x11/x11_event_source.cc;drc=6531e02635a3f00784aacad5c641cd16532657e5;l=156
[3]
https://source.chromium.org/chromium/chromium/src/+/main:ui/ozone/platform/wayland/host/wayland_event_source.h;drc=7dd58adeb690920a7474b4f7c5d71d23b15d4698;l=46

Comment 27 by sky@chromium.org on Wed, May 19, 2021, 4:49 PM EDT Project Member

Ryan, are we sure this is new? I think the right fix for this is going to be tricky, and would like to remove RBS.

Comment 28 by rsleevi@chromium.org on Wed, May 19, 2021, 10:49 PM EDT Project Member

Cc: adetaylor@chromium.org

RBS is set based on severity and impact, as covered in https://chromium.googlesource.com/chromium/src/+/refs/heads/main/docs/security/severity-guidelines.md

I'm OOO, but adetaylor@ can provide any answers timelines (Ade: See Comment #11 and Comment #27 in particular)

Comment 29 by adetaylor@chromium.org on Thu, May 20, 2021, 12:47 AM EDT Project Member

Labels: -Security_Impact-Head -ReleaseBlock-Stable -M-92 -Target-92 Security_Impact-Stable M-90 Target-90

As Ryan says, we don't ship security regressions of this severity, so that's why RBS is set. However, the user agent string in #c0 says the reporter originally reproduced this on M90 which means this is not a new regression. As such Security_Impact should be Stable, and this is not a regression, and I have cleared RBS.

As to severity, #c12 is absolutely right. If there were not the UI interaction, this would be Critical and we'd be wanting to make a release to fix this within 24-48 hours. The UI interaction makes this less urgent. But the UI interaction is not wildly far-fetched... It feels to me quite realistic that an attacker could distribute this exploit code to a large number of Chrome users (e.g. via a compromised ad network or watering-hole website) then just wait for some of them to drag some tabs. They'd get lucky with one of the users in the end.

As this is a browser process bug, the consequences are really bad. We don't have site isolation or sandboxes to save us. It's easy to imagine a variety of bad consequences: ransomware, bank account theft, impersonating enterprise users through cookie theft, etc. etc.

So, in my opinion this remains a valid High severity bug and we'll ship the fix in the next bi-weekly release as soon as it's ready. As (I believe) it's Linux only, I think we will likely have to wait for a beta cycle to get some real-user testing, before merging to stable.

Comment 30 by Git Watcher on Thu, May 20, 2021, 3:26 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/dcea5da2b3813203fc36c2c9c695de107c712cdd

commit dcea5da2b3813203fc36c2c9c695de107c712cdd
Author: Scott Violet <sky@chromium.org>
Date: Thu May 20 19:25:20 2021

x: make event dispatch dispatch a single event at a time

Without this multiple events may be dispatched during a single run
of the MessagePump. This means things like RunLoop::Quit() don't
work as expected, and can lead to unexpected state, resulting in
crashes.

On the Windows side, a single UI event is processed at a time. This
change in theory makes Linux/X behave the same as Windows.

~~BUG=1207277~~

Change-Id: I1294a04ee1539ab96059b26bbc13ba993c3dff9e
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2907732
Reviewed-by: Thomas Anderson <thomasanderson@chromium.org>
Commit-Queue: Scott Violet <sky@chromium.org>
Cr-Commit-Position: refs/heads/master@{#885156}

[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/platform_event_source.cc
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/platform_event_source.h
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/platform_event_source_unittest.cc
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/x11/x11_event_source.cc
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/x11/x11_event_source.h
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/x11/x11_event_watcher_fdwatch.cc
[modify] https://crrev.com/dcea5da2b3813203fc36c2c9c695de107c712cdd/ui/events/platform/x11/x11_event_watcher_glib.cc

Comment 31 by sky@chromium.org on Mon, May 24, 2021, 11:56 AM EDT Project Member

merc.ouc@gmail.com, any chance you could try to reproduce again on tip of tree?

Comment 32 by merc....@gmail.com on Mon, May 24, 2021, 9:41 PM EDT

Hi sky, I try to reproduce it with #885849 and it's no longer able to reproduce.

Comment 33 by sky@chromium.org on Tue, May 25, 2021, 11:28 AM EDT Project Member

Status: Fixed (was: Started)

Glad to hear it. Thanks Merc!

I'm going to move this to fixed. The fix changes how native events are processed, which is a risky change and not something I think should be merged. There is at least one regression: ~~issue 1211737~~.

Comment 34 by sheriffbot on Tue, May 25, 2021, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 35 by sheriffbot on Tue, May 25, 2021, 2:03 PM EDT Project Member

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

Comment 36 by sheriffbot on Tue, May 25, 2021, 2:24 PM EDT Project Member

Labels: Merge-Request-92 Merge-Request-90 Merge-Request-91

Requesting merge to stable M90 because latest trunk commit (885156) appears to be after stable branch point (857950).

Requesting merge to beta M91 because latest trunk commit (885156) appears to be after beta branch point (870763).

Requesting merge to future beta M92 because latest trunk commit (885156) appears to be after future beta branch point (884198).

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

Comment 37 by sheriffbot on Tue, May 25, 2021, 2:27 PM EDT Project Member

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

This bug requires manual review: Request affecting a post-stable build
Before a merge request will be considered, the following information is required to be added to this bug:

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

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

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

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

Comment 38 by sky@chromium.org on Tue, May 25, 2021, 3:47 PM EDT Project Member

See my earlier concerns about merging this. This is a risky change (with at least one regression). This needs to bake for a while before we consider any merges. IMO, we shouldn't attempt to merge it. This is also very tricky to reproduce.

Comment 39 by adetaylor@chromium.org on Tue, May 25, 2021, 4:17 PM EDT Project Member

OK. The problem is that adversaries monitor our git repositories and will have been working on weaponizing this since the first revision of your CL six days ago. Their glossy marketing brochures proudly advertise that they can weaponize our fixes in five business days, so that's about now :) That's why we need to push to merge high severity security fixes back.

For low-risk fixes we merge them back to the current stable branch after only 24-48 hours of Canary time. As this affects only Linux, though, I believe we don't have a Canary build. As you consider this high risk I'd like us to merge this to M92 only after it's been in Dev for a week, and then we can revisit whether we merge back to M91 at all (possibly not).

(I don't think this is a prime candidate for our n-day adversaries, but still, we definitely wouldn't want to leave it until M93)

Comment 40 by sky@chromium.org on Tue, May 25, 2021, 5:06 PM EDT Project Member

To trigger this crash requires a page to go fullscreen at just the right moment while the user is rapidly dragging the window between two windows. Please correct me if I'm wrong, but it seems like the chance of this happening is extremely low. Because of this, and because the fix is risky to merge, I don't think the merge is worth the risk.

Adrian, did you try to reproduce this? I wasn't able to do it, and I think it took Ryan quite a few tries.

Comment 41 by adetaylor@chromium.org on Tue, May 25, 2021, 5:52 PM EDT Project Member

I haven't tried to reproduce it. If you can persuade Ryan that this should be lower severity, that's OK with me, though as he says this has already been downgraded one notch due to the UI interaction.

FWIW I assume a skilled attacker will be able to detect when the tab is being dragged rapidly (whether or not we intentionally expose that) and will be able to go to full screen at just the right moment. So I don't see that as a huge impediment to a successful attack. On the other hand, the attacker has to get quite lucky for a user to drag the tab at all, _plus_ it's a visually-jarring thing for a page to go full screen and is unlikely to fly beneath the user's radar. So, I think realistically an attacker would have to attack at quite a lot of users to get lucky here. Attackers are in the fortunate position that they can repeat an attack many times on many users, but they'd probably choose a less intrusive bug.

But then again, that's why this is High not Critical... browser UaFs are really bad.

Sorry for banging on about severity. Our merge guidelines are closely tied to severity, and it's quite exceptional for us not to merge a High bug into beta. (It's fairly unusual for us not to merge such fixes to stable). For medium bugs, we're much more flexible.

Still, I don't want us to merge a risky fix even if we do decide this remains High. I think we'll know more in a few weeks after it's been through dev.

Comment 42 by sky@chromium.org on Wed, May 26, 2021, 12:06 PM EDT Project Member

Waiting for the fix to bake SGTM.

To trigger the use-after-free it's not enough to go fullscreen at the right time (which I think would be very tricky), the event queue has to have multiple pending move/drag events. This is why I think it would be quite hard to exploit this. I'm not saying impossible, rather extremely hard.

Comment 43 by sheriffbot on Wed, May 26, 2021, 12:21 PM EDT Project Member

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

Comment 44 by sheriffbot on Wed, May 26, 2021, 2:27 PM EDT Project Member

Labels: -Merge-Request-92 Hotlist-Merge-Approved Merge-Approved-92

Your change meets the bar and is auto-approved for M92. Please go ahead and merge the CL to branch 4515 (refs/branch-heads/4515) manually. Please contact milestone owner if you have questions.
Merge instructions: https://www.chromium.org/developers/how-tos/drover
Owners: govind@(Android), bindusuvarna@(iOS), dgagnon@(ChromeOS), srinivassista@(Desktop)

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

Comment 45 by danakj@chromium.org on Wed, May 26, 2021, 2:45 PM EDT Project Member

Cc: danakj@chromium.org

> To trigger the use-after-free it's not enough to go fullscreen at the right time (which I think would be very tricky), the event queue has to have multiple pending move/drag events.

Drive-by question. Do they _have_ to be move/drag events? If so, can a renderer cause those events through the window API?

Comment 46 by sky@chromium.org on Wed, May 26, 2021, 3:39 PM EDT Project Member

It's actually not enough to move a window. I think the sequence is something like:

1. in a window with multiple tabs start dragging one.
2. drag the tab out of the window so that another window is created.
3. as the newly created window is being dragged around, multiple events have to be queued by the OS (with the right coordaintes), the tab has to go fullscreen, and the mouse has to be in just the right position.

By the window api, do you mean window.moveto? I believe that does nothing for tabbed-browser windows:
https://chromium.googlesource.com/chromium/src/+/refs/heads/main/chrome/browser/ui/browser.cc#1697 .

Comment 47 by sky@chromium.org on Wed, May 26, 2021, 3:40 PM EDT Project Member

Labels: -Merge-Approved-92

This change actually landed in 92, so no need to merge to 92. I'm going to remove the merge to 92 labels because of this.

Comment 48 by danakj@chromium.org on Wed, May 26, 2021, 3:41 PM EDT Project Member

Oh, TIL about window.moveto's behaviour.

Thanks sky, I wondered if it seems reasonable that while _this path_ to the UAF is requires tricky user interaction, there may reasonably be other unknown paths to the same UAF. It doesn't sound likely though. IMO that's an argument for a lower severity.

Comment 49 by adetaylor@google.com on Wed, May 26, 2021, 6:25 PM EDT Project Member

Labels: -Security_Severity-High -Merge-Request-90 -Merge-Review-91 Security_Severity-Medium

> This change actually landed in 92, so no need to merge to 92. I'm going to remove the merge to 92 labels because of this.

Great!

Consensus seems to be that we can get away with lowering this to medium (especially with regards to #c48 concluding that other exploitation paths are unlikely), so I'll just do that, and in that case I'm completely happy removing the M91 merge request.

Comment 50 by sky@chromium.org on Wed, May 26, 2021, 6:37 PM EDT Project Member

Thanks Adrian.

Comment 51 by amyressler@google.com on Wed, Jun 16, 2021, 6:50 PM EDT 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 52 by amyressler@chromium.org on Wed, Jun 16, 2021, 7:08 PM EDT Project Member

Congratulations -- the VRP Panel has decided to award you $7,500 for this report. Nice work!

Comment 53 by amyressler@google.com on Fri, Jun 18, 2021, 4:49 PM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 54 by amyressler@chromium.org on Mon, Jul 19, 2021, 3:15 PM EDT Project Member

Labels: Release-0-M92

Comment 55 by amyressler@google.com on Mon, Jul 19, 2021, 7:17 PM EDT Project Member

Labels: CVE-2021-30579 CVE_description-missing

Comment 56 by rzanoni@google.com on Thu, Jul 29, 2021, 10:09 AM EDT Project Member

Labels: LTS-Security-90 LTS-Security-NotApplicable-90

Comment 57 by amyressler@google.com on Tue, Aug 3, 2021, 3:42 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 58 by sheriffbot on Tue, Aug 31, 2021, 1:33 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