

mozilla-api-keyfile=/builds/mozilla-desktop-geoloc-api.key --with-google-location-service-api-keyfile=/builds/gls-gapi.data --with-googlesafebrowsing-api-keyfile=/builds/sb-gapi.data DUMP SYMS=/builds/worker/fetches/dump syms/dump syms PDBSTR=/builds/worker/fetches/pdbstr/pdbstr.exe WINCHECKSEC=/builds/worker/fetches/winchecksec/winchecksec MAKE=/usr/bin/make MAKENSISU=/builds/worker/fetches/nsis-3.01/makensis.exe UPX=/builds/worker/fetches/upx-3.95-win64/upx.exe --enable-crashreporter --withbranding=browser/branding/nightly Please let me know if there's anything else worth collecting form this system. ¡Gracias! Alex Jan-Ivar Bruaroey [:jib] (needinfo? me) Comment 1 • 3 years ago (In reply to alex\_mayorga from comment #0) Filing as there are more reports over at https://crash-stats.mozilla.org/signature/? product=Firefox&signature=rtc%3A%3AVideoBroadcaster%3A%3AOnFrame Some of those look like UAF, so I'm marking this a security bug. Assignee: nobody → dminor Group: media-core-security Priority: -- → P2 Regressed by: 464 Jan-Ivar Bruaroey [:jib] (needinfo? me) \_ Updated • 3 years ago No longer regressed by: 1622/490 Jan-Ivar Bruaroey [:jib] (needinfo? me) Comment 2 • 3 years ago Byron, is this an area you're comfortable in at all, or should we wait for Dan? Flags: needinfo?(docfaraday) \_ Comment 3 • 3 years ago • Edited I'll look into this today. Flags: neodinfo?(docfaraday) Daniel Veditz [:dveditz] Updated • 3 years ago Keywords: csectype-uaf, sec-high Dan Minor [:dminor] Assignee Comment 4 • 3 years ago \_ It looks like we're crashing here [1] with a bad sink. From a bit of testing, the only sink present is the VideoStreamEncoder created as part of the VideoSendStream here [2]. The VideoBroadcaster and VideoSendStream (and so the VideoStreamEncoder) are all managed by the VideoConduit, so my first guess is this is some sort of shutdown race when we're tearing down the VideoSendStream. [1] https://searchfox.org/mozilla [2] https://searchfox.org/mozillacentral/rev/9aa7bebfd169bc2ead00ef596498a406e56bbb85/media/webrtc/trunk/webrtc/video/video\_send\_stream.cc#567 Release mgmt bot [:suhaib / :marco/ :calixte] \_ Crash Signature: [@ rtc::VideoBroadcaster::OnFrame]  $\rightarrow$  [@ rtc::VideoBroadcaster::OnFrame] [@ 0x4] [@ rtc::VideoBroadcaster::OnFrame(webrtc::VideoFrame const&)] Dan Minor [:dminor] Assignee Offhand, it looks like the RemoveSink code path should be fine. In both VideoBroadcaster::RemoveSink and VideoBroadcaster::OnFrame, the code takes the sinks, and, wants, lock. The VideoStreamEncoder appears to block indefinitely in its Stop method before the destructor will run, I'm going to check WebrtcVideoConduit::AddOrUpdateSink next, it will dispatch to main thread if called off main thread, so perhaps we're hitting some sort of Crash Signature: [@ rtc::VideoBroadcaster::OnFrame] [@ 0x4] [@ rtc::VideoBroadcaster::OnFrame(webrtc::VideoFrame const&)] → [@ rtc::VideoBroadcaster::OnFrame] [@ 0x4] [@ rtc::VideoBroadcaster::OnFrame(webrtc::VideoFrame const&)] Dan Minor [:dminor] Assignee Comment 7 • 3 years ago

So I think what might be happening is that we get a call to AddOrUpdateSink off main thread, it get dispatched to main thread, but before it runs, we get a call to RemoveSink because the encoder is going way. RemoveSink succeeds, the encoder is freed, but then when AddOrUpdateSink runs, it

looks like we're adding a new sink with the now invalid encoder pointer.

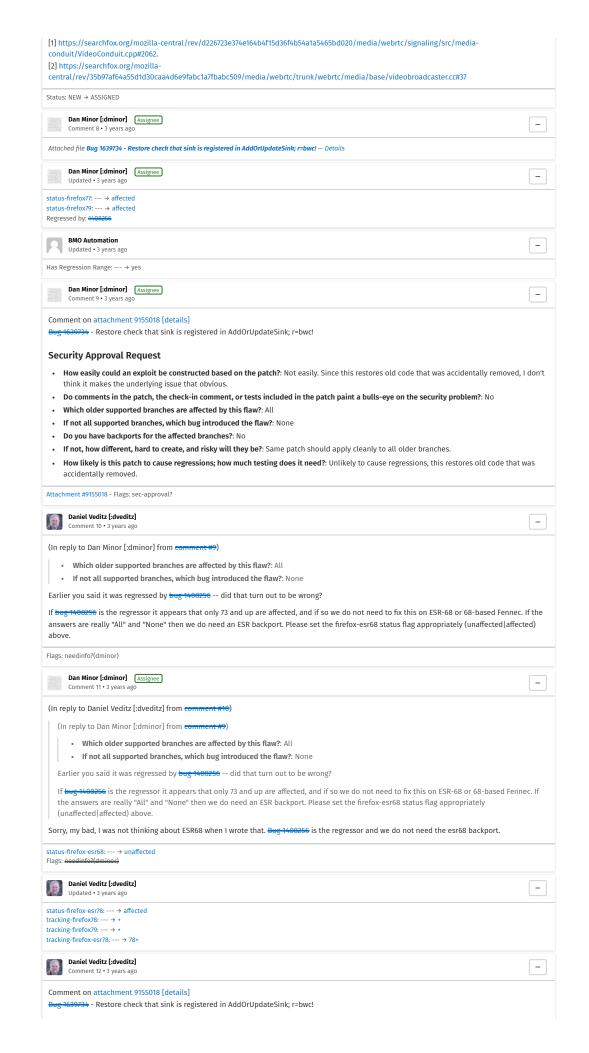
An earlier version of the code appears to guard against this by keeping a list of valid sinks and checking it in the dispatch code [1]. This was

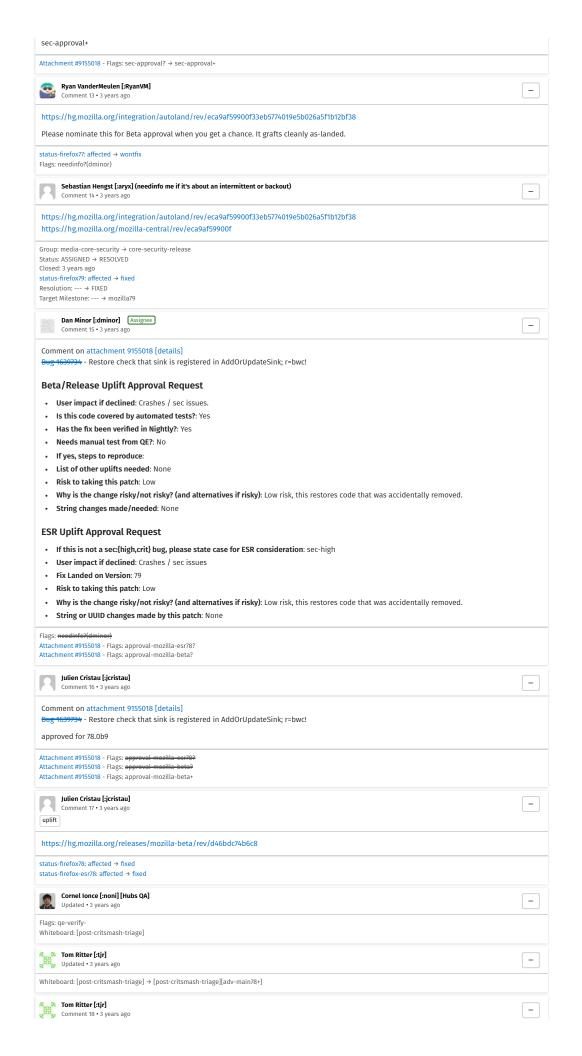
removed in Bug 1400256 which I reviewed: /. That change landed in Firefox 73 and I'm not seeing any hits on crash-stats over the past year for versions older than that, other than for ESR 60, which easily could be a separate problem.

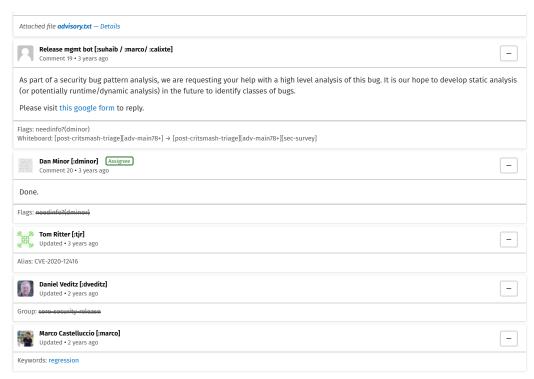
This dispatch to main is to avoid the threading assertion here [2], but VideoBroadcaster::RemoveSink takes a lock in that function anyway, so it's not

threads, but the webrtc.org code assumes creation and access will all occur on a separate "worker" thread.

clear that the assertion is really needed. We have a general problem in WebRTC where we create stuff on main and then need to access it from other







You need to  $\log$  in before you can comment on or make changes to this bug.

Top ↑