Starred by 3 users

Owner:	hongchan@chromium.org **OOO (12.15-1.8)**
CC:	mlippautz@chromium.org gov...@chromium.org adetaylor@chromium.org rtoy@chromium.org jonor...@microsoft.com haraken@chromium.org josep...@microsoft.com nmehta@google.com
Status:	Verified *(Closed)*
Components:	Blink>WebAudio
Modified:	May 21, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Security_Severity-High
allpublic
CVE_description-submitted
Target-88
M-88
merge-merged-4240
merge-merged-86
LTR-Merged-86
LTS-Security-86
merge-merged-4324
merge-merged-88
Release-0-M89
merge-merged-4389
merge-merged-89
CVE-2021-21165

Issue 1174582: Security: ScriptProcessorNode allows write of Float32Array across threads

Reported by ahuff...@microsoft.com on Thu, Feb 4, 2021, 12:24 PM EST Project Member

Code

VULNERABILITY DETAILS

ScriptProcessorNode buffers are not modified on the main thread. And they are re-used on subsequent entries into the ScriptProcessorHandler::Process function. The buffer can be modified / used in the main thread and it will be reset back to the source data by an audio thread. This could be abused in a few ways to get code execution however the below is the simplest crash demonstrating the issue.

This issue appears to effect both Chrome and Safari, trunk of the WebKit project, and should also be reported to Apple. We could submit that ourselves through our own process, however, if you have your own steps for situations like this which would streamline the process I am happy to let y'all handle it.

Just an FYI, the poc is shockingly unreliable on WebKit.

VERSION
Chrome Version: 88.0.4324.146 + stable
Operating System: Ubuntu 20.04

REPRODUCTION CASE
1) Run the attached poc.html
2) Click the words click me.
3) Renderer should crash.

CREDIT INFORMATION
Reporter credit: Alison Huffman, Microsoft Browser Vulnerability Research

webkit.asan.log
16.1 KB View Download

asan.log
17.3 KB View Download

poc.html
995 bytes View Download

Comment 1 by ahuff...@microsoft.com on Thu, Feb 4, 2021, 12:51 PM EST Project Member
Cc: josep...@microsoft.com

Comment 2 by tsepez@chromium.org on Thu, Feb 4, 2021, 2:58 PM EST
Status: Assigned (was: Unconfirmed)
Owner: hongchan@chromium.org
Labels: Security_Severity-High Security_Impact-Stable Pri-1

hongchan - you've done some work with threads in the script processor, could you take a look or re-assign as appropriate?

Comment 3 by adetaylor@google.com on Thu, Feb 4, 2021, 3:13 PM EST

ahuffman@ thanks for the report. I think you should go ahead and report this to WebKit independently. We do have a process for reporting WebKit bugs, but I think it would just add complexity to have us in the communications loop.

Could you let us know when you've got some reference number/ticket ID from Apple about this bug? That'd be helpful. Thanks!

Comment 4 by hongchan@chromium.org on Thu, Feb 4, 2021, 3:14 PM EST

Re #c2: I'll take a look.

Comment 5 by hongchan@chromium.org on Thu, Feb 4, 2021, 3:49 PM EST

We already have |process_event_lock_| to avoid this race, but it seems not enough.

https://source.chromium.org/chromium/chromium/src/+/master:third_party/blink/renderer/modules/webaudio/script_processor_node.cc;l=242

Looking further...

Comment 6 by hongchan@chromium.org on Fri, Feb 5, 2021, 12:25 PM EST

Status: Started (was: Assigned)

Comment 7 by sheriffbot on Fri, Feb 5, 2021, 12:48 PM EST

Labels: Target-88 M-88

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

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

Comment 8 by hongchan@chromium.org on Fri, Feb 5, 2021, 2:59 PM EST

My guess is that AudioNode::DispatchEvent() will not fire the dispatched event immediately. In some cases, it will be queued in the event vector, and get fired later. In that case, the MutexLocker and its scope do not work anymore.

Comment 9 by ahuff...@microsoft.com on Sat, Feb 6, 2021, 2:13 AM EST Project Member

Correct me if I am wrong, but it appears simpler than that to me.

https://source.chromium.org/chromium/chromium/src/+/master:third_party/blink/renderer/modules/webaudio/script_processor_node.cc;drc=2ac64302ae161cd6b5e4b1254497bdf5fd6d3415;l=162

I believe the shared_input_buffer and shared_output_buffer have the same backing stores as the the input_buffers_ and output_buffers_.

https://source.chromium.org/chromium/chromium/src/+/master:third_party/blink/renderer/modules/webaudio/audio_buffer.cc;l=319;drc=2ac64302ae161cd6b5e4b1254497bdf5fd6d3415;bpv=1;bpt=1

So this audio thread is copying into the buffer that the main thread has access to.

Comment 10 by hongchan@chromium.org on Tue, Feb 9, 2021, 12:23 PM EST

Cc: rtoy@chromium.org
Labels: -OS-iOS

Comment 11 by bugdroid on Tue, Feb 9, 2021, 12:41 PM EST

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/60987aa224f369fc0ea38c56e498389440921356

commit 60987aa224f369fc0ea38c56e498389440921356
Author: Hongchan Choi <hongchan@chromium.org>
Date: Tue Feb 09 17:40:12 2021

Prevent accessing shared buffers from audio rendering thread

The shared buffer in ScriptProcessorNode can be accessed by the
audio rendering thread when it is held by the main thread.

The solution suggested here is simply to expand the scope of
the mutex to minimize the code change. This is a deprecated
feature in Web Audio, so making significant changes is not
sensible. By locking the entire scope of Process() call, this
area would be immune to the similar problems in the future.

~~Bug: 1174682~~
Test: The repro case doesn't crash on ASAN.
Change-Id: I2b292f94be65e6ec26c6eb0e0ed32b3fb2d88466
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2681193
Commit-Queue: Hongchan Choi <hongchan@chromium.org>
Reviewed-by: Raymond Toy <rtoy@chromium.org>
Cr-Commit-Position: refs/heads/master@{#852240}

[modify] https://crrev.com/60987aa224f369fc0ea38c56e498389440921356/third_party/blink/renderer/modules/webaudio/script_processor_node.cc

Comment 12 Deleted

Comment 13 by hongchan@chromium.org on Tue, Feb 9, 2021, 3:28 PM EST

To fix the POC #c12, we can pass a new buffer for every audio processing event. That will impact the performance of ScriptProcessorNode significantly.

ahuffman@ Do you have other thoughts/suggestion?

Comment 14 by ahuff...@microsoft.com on Tue, Feb 9, 2021, 4:00 PM EST Project Member

The way that I know to exploit this relies on the buffer being marked non-shared. I don't think the issue would exist if that weren't the case as APIs consuming the buffer would typically make a copy. Are the ramifications for just marking the inputBuffer as a shared buffer? Could be other subtle issues that I am not considering?

Looking at the standard it (https://www.w3.org/TR/webaudio/#AudioProcessingEvent-attributes) it says:

inputBuffer, of type AudioBuffer, readonly
An AudioBuffer containing the input audio data. It will have a number of channels equal to the numberOfInputChannels parameter of the createScriptProcessor() method. This AudioBuffer is only valid while in the scope of the onaudioprocess function. Its values will be meaningless outside of this scope.

Comment 15 by hongchan@chromium.org on Tue, Feb 9, 2021, 4:18 PM EST

> Are the ramifications for just marking the inputBuffer as a shared buffer?

I believe marking a buffer as shared or non-shared is only possible when the other thread is a proper Worker thread. (I am assuming you're referring to APIs like SharedArrayBuffer or ArrayBuffer). The Web Audio rendering thread is not exposed to the web platform by design, so the generic shared memory scheme in the web platform doesn't work for this case. It's unfortunate, but this is how ScriptProcessorNode works and that's why it is deprecated from the spec.

> This AudioBuffer is only valid while in the scope of the onaudioprocess function. Its values will be meaningless outside of this scope.

The buffer object is not "invalidated" outside of onaudioprocess function. It only says values in it are meaningless, but still accessible. So the spec text isn't wrong...

Comment 16 by hongchan@chromium.org on Wed, Feb 10, 2021, 3:02 PM EST

Cc: mlippautz@chromium.org haraken@chromium.org

mlippautz@ and haraken@:

https://source.chromium.org/chromium/chromium/src/+/master:third_party/blink/renderer/modules/webaudio/script_processor_node.h;l=102

SharedAudioBuffer was created to allow both threads (main, web audio render) to access the buffer contents. Based on the current design, I don't think it's possible to limit the access from the main thread while the render thread is touching it.

My conclusion so far is to create new buffers per onaudioprocess call, but this has a significant audio performance implication. (GC and malloc) Based on the metric, this change will impact some popular audio apps even though the feature is officially deprecated from the spec years ago.

Could you share your opinion on this issue?

Comment 17 by haraken@google.com on Thu, Feb 11, 2021, 12:05 AM EST

In general, shared memory programming is fragile. If you can create new buffers per onaudioprocess call and let the main thread and the render thread access different buffers, that's the best from the architecture perspective.

If the feature is deprecated, how much do we need to worry about the performance impact? We should probably encourage the website to stop using it...?

Comment 18 by mlippautz@chromium.org on Thu, Feb 11, 2021, 3:12 AM EST

I remember introducing the SharedAudioBuffer to clean up roots and avoid even broader concurrency access and roots that could lead to cycles.

+1 to haraken's comment if we can get away with performance. I have no other suggestion as I remember there was also an issue where the spec mandated using non-shared buffers for the actual data which should rule out any sane design around shared memory.

Comment 19 by hongchan@chromium.org on Thu, Feb 11, 2021, 11:11 AM EST

Thanks for your responses.

rtoy@

Per comments above, creating new buffers per every AudioProcessingEvent is the only sane way to resolve this. I still have to flesh out the details for corner cases, but this will be a breaking change for some apps.

Comment 20 by hongchan@chromium.org on Thu, Feb 11, 2021, 11:12 AM EST

Cc: adetaylor@chromium.org

adtaylor@

For easier tracking, I suggest we merge the change #c11 first, and then open a new issue to track the new change because it might take longer than change a few lines of code. What do you think?

Comment 21 by adetaylor@chromium.org on Thu, Feb 11, 2021, 3:33 PM EST

That sounds good. Please could you cc ahuffman@microsoft.com on the new issue and add the label reward_to-ahuffman_at_microsoft.com such that the reporter gets properly credited when the new issue is eventually fixed.

Comment 22 by hongchan@chromium.org on Thu, Feb 11, 2021, 3:45 PM EST

Now the remaining issue is tracked in issue 1177465.

Comment 23 by hongchan@chromium.org on Thu, Feb 11, 2021, 3:47 PM EST

Labels: Merge-Request-88

Comment 24 by hongchan@chromium.org on Thu, Feb 11, 2021, 3:47 PM EST

Status: Verified (was: Started)

Comment 25 by gov...@chromium.org on Thu, Feb 11, 2021, 8:38 PM EST

Labels: Merge-Request-89

This will also need a merge to M89.

Comment 26 by sheriffbot on Thu, Feb 11, 2021, 8:43 PM EST

Labels: -Merge-Request-89 Merge-Review-89 Hotlist-Merge-Review

This bug requires manual review: M89's targeted beta branch promotion date has already passed, so this requires manual review
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), geohsu@(ChromeOS), pbommana@(Desktop)

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

Comment 27 by hongchan@chromium.org on Fri, Feb 12, 2021, 11:38 AM EST

1. Yes.
2. https://chromium-review.googlesource.com/c/chromium/src/+/2681193
3. Yes.
4. Yes.
5. It's Security_Severity_High.
6. No.
7. N/A

Comment 28 by adetaylor@chromium.org on Fri, Feb 12, 2021, 11:52 AM EST

Labels: -Merge-Review-89 Merge-Approved-89

Approving merge to M89, branch 4389, but please could you wait till after the weekend such that we've got a few days of Canary coverage?

Comment 29 by hongchan@chromium.org on Fri, Feb 12, 2021, 11:55 AM EST

Yes. Will do.

Comment 30 by sheriffbot on Fri, Feb 12, 2021, 1:56 PM EST

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

Comment 31 by sheriffbot on Mon, Feb 15, 2021, 12:14 PM EST

Cc: gov...@chromium.org

This issue has been approved for a merge. Please merge the fix to any appropriate branches as soon as possible!

If all merges have been completed, please remove any remaining Merge-Approved labels from this issue.

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

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

Comment 32 by bugdroid on Tue, Feb 16, 2021, 1:42 PM EST

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

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/53eedb3282b894c16e5b8f5e54df3f294045b3ed

commit 53eedb3282b894c16e5b8f5e54df3f294045b3ed
Author: Hongchan Choi <hongchan@chromium.org>
Date: Tue Feb 16 18:42:36 2021

Prevent accessing shared buffers from audio rendering thread

The shared buffer in ScriptProcessorNode can be accessed by the
audio rendering thread when it is held by the main thread.

The solution suggested here is simply to expand the scope of
the mutex to minimize the code change. This is a deprecated
feature in Web Audio, so making significant changes is not
sensible. By locking the entire scope of Process() call, this
area would be immune to the similar problems in the future.

(cherry picked from commit 60987aa224f369fc0ea38c56e498389440921356)

~~Bug: 1174582~~
Test: The repro case doesn't crash on ASAN.
Change-Id: I2b292f94be65e6ec26c6eb0e0ed32b3fb2d88466
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2681193
Commit-Queue: Hongchan Choi <hongchan@chromium.org>
Reviewed-by: Raymond Toy <rtoy@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#852240}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2697471
Reviewed-by: Hongchan Choi <hongchan@chromium.org>
Cr-Commit-Position: refs/branch-heads/4389@{#1095}
Cr-Branched-From: 9251c5db2b6d5a59fe4eac7aafa5fed37c139bb7-refs/heads/master@{#843830}

[modify] https://crrev.com/53eedb3282b894c16e5b8f5e54df3f294045b3ed/third_party/blink/renderer/modules/webaudio/script_processor_node.cc

Comment 33 by adetaylor@google.com on Tue, Feb 23, 2021, 4:00 PM EST

Issue 1181341 has been merged into this issue.

Comment 34 by adetaylor@google.com on Tue, Feb 23, 2021, 4:02 PM EST

Labels: -Merge-Request-88 Merge-Approved-88

Approving merge to M88, branch 4324, due to information in issue 1181341.

Comment 35 by hongchan@chromium.org on Tue, Feb 23, 2021, 4:16 PM EST

ahuffman@ - I moved your comment above to issue 1177465.

Comment 36 by ClusterFuzz on Tue, Feb 23, 2021, 4:16 PM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5137335457742848.

Comment 37 by adetaylor@google.com on Tue, Feb 23, 2021, 5:29 PM EST

Cc: nmehta@google.com

Comment 38 by bugdroid on Tue, Feb 23, 2021, 6:28 PM EST

Labels: -merge-approved-88 merge-merged-4324 merge-merged-88

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/aeb6bc551b607e0c80c232ed4817c0ff5e9a7784

commit aeb6bc551b607e0c80c232ed4817c0ff5e9a7784
Author: Hongchan Choi <hongchan@chromium.org>
Date: Tue Feb 23 23:27:31 2021

Prevent accessing shared buffers from audio rendering thread

The shared buffer in ScriptProcessorNode can be accessed by the
audio rendering thread when it is held by the main thread.

The solution suggested here is simply to expand the scope of
the mutex to minimize the code change. This is a deprecated
feature in Web Audio, so making significant changes is not
sensible. By locking the entire scope of Process() call, this
area would be immune to the similar problems in the future.

(cherry picked from commit 60987aa224f369fc0ea38c56e498389440921356)

~~Bug: 1174582~~
Test: The repro case doesn't crash on ASAN.
Change-Id: I2b292f94be65e6ec26c6eb0e0ed32b3fb2d88466
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2681193
Commit-Queue: Hongchan Choi <hongchan@chromium.org>

Reviewed-by: Raymond Toy <rtoy@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#852240}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2715585
Commit-Queue: Krishna Govind <govind@chromium.org>
Reviewed-by: Srinivas Sista <srinivassista@chromium.org>
Cr-Commit-Position: refs/branch-heads/4324@{#2238}
Cr-Branched-From: c73b5a651d37a6c4d0b8e3262cc4015a5579c6c8-refs/heads/master@{#827102}

[modify] https://crrev.com/aeb6bc551b607e0c80c232ed4817c0ff5e9a7784/third_party/blink/renderer/modules/webaudio/script_processor_node.cc

Comment 39 by adetaylor@google.com on Fri, Feb 26, 2021, 12:50 PM EST

Labels: Release-0-M89

Comment 40 by asumaneev@google.com on Mon, Mar 1, 2021, 2:25 PM EST

Labels: LTS-Security-86 LTS-Merge-Request-86

Comment 41 by adetaylor@google.com on Mon, Mar 1, 2021, 7:27 PM EST

Labels: CVE-2021-21165 CVE_description-missing

Comment 42 by gianluca@google.com on Tue, Mar 2, 2021, 9:03 AM EST

Labels: LTS-Merge-Approved-86

Comment 43 by asumaneev@google.com on Tue, Mar 2, 2021, 9:06 AM EST

Labels: -LTS-Merge-Request-86

Comment 44 by bugdroid on Tue, Mar 2, 2021, 10:17 AM EST

Labels: merge-merged-4240 merge-merged-86

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/dea071d8b30fde63954eef55fd2c685ff3b3f083

commit dea071d8b30fde63954eef55fd2c685ff3b3f083
Author: Hongchan Choi <hongchan@chromium.org>
Date: Tue Mar 02 15:15:59 2021

Prevent accessing shared buffers from audio rendering thread

The shared buffer in ScriptProcessorNode can be accessed by the
audio rendering thread when it is held by the main thread.

The solution suggested here is simply to expand the scope of
the mutex to minimize the code change. This is a deprecated
feature in Web Audio, so making significant changes is not
sensible. By locking the entire scope of Process() call, this
area would be immune to the similar problems in the future.

(cherry picked from commit 60987aa224f369fc0ea38c56e498389440921356)

(cherry picked from commit aeb6bc551b607e0c80c232ed4817c0ff5e9a7784)

~~Bug: 1174582~~
Test: The repro case doesn't crash on ASAN.
Change-Id: I2b292f94be65e6ec26c6eb0e0ed32b3fb2d88466
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2681193
Commit-Queue: Hongchan Choi <hongchan@chromium.org>
Reviewed-by: Raymond Toy <rtoy@chromium.org>
Cr-Original-Original-Commit-Position: refs/heads/master@{#852240}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2715585
Commit-Queue: Krishna Govind <govind@chromium.org>
Reviewed-by: Srinivas Sista <srinivassista@chromium.org>
Cr-Original-Commit-Position: refs/branch-heads/4324@{#2238}
Cr-Original-Branched-From: c73b5a651d37a6c4d0b8e3262cc4015a5579c6c8-refs/heads/master@{#827102}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2726911
Reviewed-by: Victor-Gabriel Savu <vsavu@google.com>
Commit-Queue: Artem Sumaneev <asumaneev@google.com>
Cr-Commit-Position: refs/branch-heads/4240@{#1552}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/dea071d8b30fde63954eef55fd2c685ff3b3f083/third_party/blink/renderer/modules/webaudio/script_processor_node.cc

Comment 45 by asumaneev@google.com on Tue, Mar 2, 2021, 10:20 AM EST

Labels: -LTS-Merge-Approved-86 LTR-Merged-86

Comment 46 by amyressler@google.com on Tue, Mar 9, 2021, 12:58 PM EST

Labels: -CVE_description-missing CVE_description-submitted

Comment 47 by adetaylor@google.com on Mon, Mar 22, 2021, 4:23 PM EDT

ahuffman@ - the immediate fix for this caused some performance issues, which Hongchan is resolving in ~~issue 1187016~~ by subdividing the mutex. I wondered if you wanted to cast an adversarial eye over that fix to spot any security concerns. That'd be very much appreciated. Please do bear in mind that ~~issue 1187016~~ is public.

Comment 48 by sheriffbot on Fri, May 21, 2021, 1:50 PM EDT

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