













☆ Starred by 2 users

Owner:

 hongchan@chromium.org  
OOO (12.15-1.8)

CC:

 adetaylor@chromium.org  
 benmason@chromium.org  
 rtoy@chromium.org  
 pbomm...@chromium.org  
 scheib@chromium.org  
 haraken@chromium.org  
 achuith@chromium.org  
 mlippautz@chromium.org  
 vahl@chromium.org  
 hpayer@chromium.org  
 ecmziegler@google.com

Status:

Fixed (Closed)

Components:

Blink>JavaScript>GarbageCollection  
Blink>WebAudio

Modified:

Mar 20, 2020

Backlog-Rank:

----

Editors:

----

EstimatedDays:

----

NextAction:

2020-01-02

OS:

Linux, Android, Windows, Chrome, Fuchsia

Pri:

1

Type:

Bug-Security

Hotlist-Merge-Review  
Security\_Impact-Stable  
Arch-x86\_64  
Hotlist-Merge-Approved  
M-80  
Security\_Severity-High  
reward-7500  
allpublic  
reward-inprocess  
Via-Wizard-Security  
CVE\_description-submitted  
VulnerabilityAnalysis-Requested  
VulnerabilityAnalysis-Submitted  
merge-merged-3945  
Merge-Merged-79  
merge-merged-3987

Issue 1029462: use-after-free in AudioWorklet

Reported by cdsr...@gmail.com on Fri, Nov 29, 2019, 5:47 AM EST

 Code

UserAgent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36

Steps to reproduce the problem:  
Repro Steps:  
1 Build chromium with ASAN(Chromium 80.0.3979.0 ).  
2 Start webserver.  
python3.6m -m http.server 8605  
3 ./chrome --js-flags="--expose-gc" http://127.0.0.1:8605  
4 Click "play" button,and repro use-after-free.

What is the expected behavior?

What went wrong?  
The cause of this bug is that the memory pointed to by the AudioBus pointers stored in the local variables "input\_buses" and "output\_buses" in the AudioWorkletHandler :: Process function is incorrectly gc, causing UAF.  
The local variable,input\_buses and output\_buses are shown below.

void AudioWorkletHandler::Process(uint32\_t frames\_to\_process) {  
.....  
if (processor\_ && !processor\_>hasErrorOccured()) {  
Vector<AudioBus\*> input\_buses;  
Vector<AudioBus\*> output\_buses;  
for (unsigned i = 0; i < NumberOfInputs(); ++i) {  
// If the input is not connected, inform the processor of that  
// fact by setting the bus to null.  
AudioBus\* bus = Input(i).IsConnected() ? Input(i).Bus() : nullptr;  
input\_buses.push\_back(bus);  
}  
for (unsigned i = 0; i < NumberOfOutputs(); ++i)  
output\_buses.push\_back(Output(i).Bus());  
.....  
// Run the render code and check the state of processor. Finish the  
// processor if needed.  
if (!processor\_>Process(&input\_buses, &output\_buses, &param\_value\_map\_) ||  
processor\_>hasErrorOccured()) {  
FinishProcessorOnRenderThread();  
}  
}.....  
}  
}

During the execution of processor\_> Process, if the AudioWorkletNode object is released, it will cause the AudioWorkletHandler object to be released, thereby releasing the inputs\_ member and outputs\_ member in the AudioNode object, while the input\_buses and output\_buses in the above code snippet are still retained Pointer to AudioBus.

```
Vector<std::unique_ptr<AudioNodeInput>> inputs_;
Vector<std::unique_ptr<AudioNodeOutput>> outputs_;
```

PoC process:

1. Create an AudioContext object and AudioWorkletNode object, and set the numberOfInputs property of the AudioWorkletNode object to a larger value.
2. After adding AudioWorkletNode to DestinationNode, the above-mentioned AudioWorkletHandler :: Process is called.
3. Call location.reload () at the appropriate time to refresh the page, and then call gc () to force garbage collection.
4. During the garbage collection process, for the AudioContext object in the page, because the IsContextDestroyed () method returns true, the AudioWorkletNode object in the page cannot be tracked, so it will be GCed. In the end, the inputs\_ and outputs\_ members of the AudioNode will be Released, but local variables input\_buses and output\_buses in AudioWorkletHandler :: Process are still in use.

```
void ActiveScriptWrappableBase::TraceActiveScriptWrappables(
    v8::Isolate* isolate,
    Visitor* visitor) {
    .....
    .....
    for (const auto& active_wrappable : *active_script_wrappables) {
        HeapObjectHeader const* const header =
            active_wrappable->GetHeapObjectHeader();
        if ((header == BlinkGC::kNotFullyConstructedObject) ||
            header->IsInConstruction())
            continue;

        if (active_wrappable->IsContextDestroyed())
            continue;

        if (!active_wrappable->DispatchHasPendingActivity())
            continue;

        ScriptWrappable* script_wrappable = active_wrappable->ToScriptWrappable();
        visitor->Trace(script_wrappable);
    }
}
```

Did this work before? N/A

Chrome version: 80.0.3979.0 Channel: stable  
OS Version: 6.1 (Windows 7, Windows Server 2008 R2)  
Flash Version:

The cause of this bug is that the memory pointed to by the AudioBus pointers stored in the local variables "input\_buses" and "output\_buses" in the AudioWorkletHandler :: Process function is incorrectly gc, causing UAF.  
The local variable, input\_buses and output\_buses are shown below.

```
void AudioWorkletHandler::Process(uint32_t frames_to_process) {
    .....
    if (processor_ && !processor_->hasErrorOccured()) {
        Vector<AudioBus*> input_buses;
        Vector<AudioBus*> output_buses;
        for (unsigned i = 0; i < NumberOfInputs(); ++i) {
            // If the input is not connected, inform the processor of that
            // fact by setting the bus to null.
            AudioBus* bus = Input(i).IsConnected() ? Input(i).Bus() : nullptr;
            input_buses.push_back(bus);
        }
        for (unsigned i = 0; i < NumberOfOutputs(); ++i)
            output_buses.push_back(Output(i).Bus());
    }
    .....
    // Run the render code and check the state of processor. Finish the
    // processor if needed.
    if (!processor_->Process(&input_buses, &output_buses, &param_value_map_) ||
        processor_->hasErrorOccured()) {
        FinishProcessorOnRenderThread();
    }
    }.....
}
```

During the execution of processor\_ -> Process, if the AudioWorkletNode object is released, it will cause the AudioWorkletHandler object to be released, thereby releasing the inputs\_ member and outputs\_ member in the AudioNode object, while the input\_buses and output\_buses in the above code snippet are still retained Pointer to AudioBus.

```
Vector<std::unique_ptr<AudioNodeInput>> inputs_;
Vector<std::unique_ptr<AudioNodeOutput>> outputs_;
```

PoC process:

1. Create an AudioContext object and AudioWorkletNode object, and set the numberOfInputs property of the AudioWorkletNode object to a larger value.
2. After adding AudioWorkletNode to DestinationNode, the above-mentioned AudioWorkletHandler :: Process is called.
3. Call location.reload () at the appropriate time to refresh the page, and then call gc () to force garbage collection.
4. During the garbage collection process, for the AudioContext object in the page, because the IsContextDestroyed () method returns true, the AudioWorkletNode object in the page cannot be tracked, so it will be GCed. In the end, the inputs\_ and outputs\_ members of the AudioNode will be Released, but local variables input\_buses and output\_buses in AudioWorkletHandler :: Process are still in use.

```
void ActiveScriptWrappableBase::TraceActiveScriptWrappables(
    v8::Isolate* isolate,
    Visitor* visitor) {
    .....
    .....
    for (const auto& active_wrappable : *active_script_wrappables) {
        HeapObjectHeader const* const header =
            active_wrappable->GetHeapObjectHeader();
        if ((header == BlinkGC::kNotFullyConstructedObject) ||
            header->IsInConstruction())
            continue;

        if (active_wrappable->IsContextDestroyed())
            continue;

        if (!active_wrappable->DispatchHasPendingActivity())
            continue;

        ScriptWrappable* script_wrappable = active_wrappable->ToScriptWrappable();
```

```
visitor->Trace(script_wrappable);
}
}
```

[Deleted] poc.zip

[Deleted] symbolise.txt

Comment 1 by ClusterFuzz on Mon, Dec 2, 2019, 5:38 PM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at <https://clusterfuzz.com/testcase?key=6502121713041408>.

Comment 2 by palmer@chromium.org on Mon, Dec 2, 2019, 5:38 PM EST

Status: Assigned (was: Unconfirmed)

Owner: hongchan@chromium.org

Cc: rtoy@chromium.org

Labels: Security\_Severity-High M-80 OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac Pri-1

Comment 3 by rtoy@chromium.org on Mon, Dec 2, 2019, 5:46 PM EST

Components: Blink>WebAudio

Comment 4 by rtoy@chromium.org on Mon, Dec 2, 2019, 6:13 PM EST

The repro tests works very reliably for me on linux. Crashes in about a sec after pressing Play.

Comment 5 by ClusterFuzz on Mon, Dec 2, 2019, 6:29 PM EST

Labels: Security\_Impact-Stable

Testcase 6502121713041408 failed to reproduce the crash. Please inspect the program output at <https://clusterfuzz.com/testcase?key=6502121713041408>.

Comment 6 by hongchan@chromium.org on Tue, Dec 3, 2019, 7:06 PM EST

I can see how UAF may happen, but I failed to reproduce it on MacOS. (ASAN 80.0.3985.0)

cdsrc2016@ Is the original report from Windows?

Comment 7 by cdsrc...@gmail.com on Tue, Dec 3, 2019, 7:27 PM EST

I've only tested it on Linux. But windows should be the same.

Comment 8 by hongchan@chromium.org on Wed, Dec 4, 2019, 1:49 PM EST

Status: Started (was: Assigned)

Cc: haraken@chromium.org

Labels: -OS-Mac

Just Confirmed this is reproducible on Linux. But not on Mac OS.

Also some other findings:

1. Changed the 'numberOfInputs' in the repro case to 8, but didn't make any difference.
2. Placed a MutexTryLocker where AudioBuses are accessed in the AudioWorkletGlobalScope, but then UAF moved to a different line. (AudioNodeOutput)
3. Placed a MutexTryLocker within AudioWorkletHandler::Process() method, but still getting UAF.
4. Forced AudioWorkletNode::HasPendingActivity() returns always true, but also didn't make any difference.

It seems like once GS sweep start marching it, there no way to stop it. I wish there is a way for GC to wait while the audio rendering doing the for-loop inside of Process(). This is a common theme of UAFs in the WebAudio.

haraken@ Could you take a look?

The latest stack trace below:

```
==1==ERROR: AddressSanitizer: heap-use-after-free on address 0x603000f3cec4 at pc 0x5596b60371ca bp 0x7f3324fe1c50 sp 0x7f3324fe1c48
READ of size 4 at 0x603000f3cec4 thread T20 (AudioWorklet th)
#0 0x5596b60371c9 in size J.J.J./third_party/blink/renderer/platform/wtf/vector.h:1044:36
#1 0x5596b60371c9 in NumberOfChannels J.J.J./third_party/blink/renderer/platform/audio/audio_bus.h:81:56
#2 0x5596b60371c9 in blink::AudioWorkletGlobalScope::Process(blink::AudioWorkletProcessor*, WTF::Vector<blink::AudioBus*, 0u, WTF::PartitionAllocator>*, WTF::Vector<blink::AudioBus*, 0u, WTF::PartitionAllocator>*, WTF::HashMap<WTF::String, std::__1::unique_ptr<blink::AudioArray<float>, std::__1::default_delete<blink::AudioArray<float>>>, WTF::StringHash, WTF::HashTraits<WTF::String>, WTF::HashTraits<std::__1::unique_ptr<blink::AudioArray<float>, std::__1::default_delete<blink::AudioArray<float>>>>, WTF::PartitionAllocator>*) J.J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_global_scope.cc:249:21
#3 0x5596b604d3d6 in blink::AudioWorkletHandler::Process(unsigned int) J.J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc:118:22
#4 0x5596b5f817cf in blink::AudioHandler::ProcessIfNecessary(unsigned int) J.J.J./third_party/blink/renderer/modules/webaudio/audio_node.cc:363:7
#5 0x5596b5f75bdf in blink::AudioNodeOutput::Pull(blink::AudioBus*, unsigned int) J.J.J./third_party/blink/renderer/modules/webaudio/audio_node_output.cc:137:13
#6 0x5596b5f96aa6 in SumAllConnections J.J.J./third_party/blink/renderer/modules/webaudio/audio_node_input.cc:128:40
#7 0x5596b5f96aa6 in blink::AudioNodeInput::Pull(blink::AudioBus*, unsigned int) J.J.J./third_party/blink/renderer/modules/webaudio/audio_node_input.cc:158:3
#8 0x5596b60f0ecc in blink::RealtimeAudioDestinationHandler::Render(blink::AudioBus*, unsigned int, blink::AudioIOPosition const&, blink::AudioCallbackMetric const&) J.J.J./third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc:204:39
#9 0x5596b110483 in blink::AudioDestination::RequestRender(unsigned long, unsigned long, double, double, unsigned long) J.J.J./third_party/blink/renderer/platform/audio/audio_destination.cc:247:17
#10 0x5596a5dbbf92 in Run J.J.J./base/callback.h:98:12
#11 0x5596a5dbbf92 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) J.J.J./base/task/common/task_annotator.cc:142:33
#12 0x5596a5df46c8 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*, bool*) J.J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:365:23
#13 0x5596a5df4049 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoSomeWork() J.J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:219:7
#14 0x5596a5cfd160 in base::MessagePumpDefault::Run(base::MessagePump::Delegate*) J.J.J./base/message_loop/message_pump_default.cc:39:55
#15 0x5596a5df464e in Run J.J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:463:12
#16 0x5596a5df464e in non-virtual thunk to base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta) J.J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:0:0
#17 0x5596a5db681 in base::RunLoop::Run() J.J.J./base/run_loop.cc:156:14
#18 0x5596a40c4785 in blink::scheduler::WorkerThread::SimpleThreadImpl::Run() J.J.J./third_party/blink/renderer/platform/scheduler/worker/worker_thread.cc:169:14
#19 0x5596a5f1a581 in base::(anonymous namespace)::ThreadFunc(void*) J.J.J./base/threading/platform_thread_posix.cc:81:13
#20 0x7f33482cfc72 in start_thread ??:0
```

0x603000f3cec4 is located 20 bytes inside of 32-byte region [0x603000f3ceb0,0x603000f3ced0)

freed by thread T0 (chrome) here:

```
#0 0x55969bec320d in free /b/swarming/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_malloc_linux.cpp:123:3
#1 0x5596b5f8f915 in operator delete J.J.J./third_party/blink/renderer/platform/wtf/thread_safe_ref_counted.h:54:3
#2 0x5596b5f8f915 in DeleteInternal<blink::AudioBus> J.J.J./third_party/blink/renderer/platform/wtf/thread_safe_ref_counted.h:64:5
#3 0x5596b5f8f915 in Destruct J.J.J./third_party/blink/renderer/platform/wtf/thread_safe_ref_counted.h:44:5
#4 0x5596b5f8f915 in Release J.J.J./base/memory/ref_counted.h:407:7
#5 0x5596b5f8f915 in Release J.J.J./base/memory/scoped_refptr.h:322:8
#6 0x5596b5f8f915 in ~scoped_refptr J.J.J./base/memory/scoped_refptr.h:224:7
#7 0x5596b5f8f915 in ~AudioNodeOutput J.J.J./third_party/blink/renderer/modules/webaudio/audio_node_output.h:43:22
#8 0x5596b5f8f915 in std::__1::default_delete<blink::AudioNodeOutput>::operator()(blink::AudioNodeOutput*) const J.J.J./buildtools/third_party/libc++/trunk/include/memory:2378:5
```

#9 0x5596b5f7e497 in reset *J.J./buildtools/third\_party/libc++/trunk/include/memory*:2633:7  
#10 0x5596b5f7e497 in ~unique\_ptr *J.J./buildtools/third\_party/libc++/trunk/include/memory*:2587:19  
#11 0x5596b5f7e497 in Destruct *J.J./third\_party/blink/renderer/platform/wtf/vector.h*:91:13  
#12 0x5596b5f7e497 in Destruct *J.J./third\_party/blink/renderer/platform/wtf/vector.h*:337:5  
#13 0x5596b5f7e497 in Finalize *J.J./third\_party/blink/renderer/platform/wtf/vector.h*:1284:7  
#14 0x5596b5f7e497 in ~ConditionalDestructor *J.J./third\_party/blink/renderer/platform/wtf/conditional\_destructor.h*:24:59  
#15 0x5596b5f7e497 in blink::AudioHandler::~AudioHandler() *J.J./third\_party/blink/renderer/modules/webaudio/audio\_node.cc*:94:1  
#16 0x5596b604bcd in blink::AudioWorkletHandler::~AudioWorkletHandler() *J.J./third\_party/blink/renderer/modules/webaudio/audio\_worklet\_node.cc*:71:45  
#17 0x5596b5f82ec4 in DeleteInternal<blink::AudioHandler> *J.J./third\_party/blink/renderer/platform/wtf/thread\_safe\_ref\_counted.h*:64:5  
#18 0x5596b5f82ec4 in Destruct *J.J./third\_party/blink/renderer/platform/wtf/thread\_safe\_ref\_counted.h*:44:5  
#19 0x5596b5f82ec4 in Release *J.J./base/memory/ref\_counted.h*:407:7  
#20 0x5596b5f82ec4 in Release *J.J./base/memory/scoped\_refptr.h*:322:8  
#21 0x5596b5f82ec4 in ~scoped\_refptr *J.J./base/memory/scoped\_refptr.h*:224:7  
#22 0x5596b5f82ec4 in reset *J.J./base/memory/scoped\_refptr.h*:254:18  
#23 0x5596b5f82ec4 in operator= *J.J./base/memory/scoped\_refptr.h*:240:5  
#24 0x5596b5f82ec4 in blink::AudioNode::~AudioNode() *J.J./third\_party/blink/renderer/modules/webaudio/audio\_node.cc*:598:14  
#25 0x5596a3e88b83 in Finalize *J.J./third\_party/blink/renderer/platform/heap/heap\_page.cc*:94:5  
#26 0x5596a3e88b83 in blink::NormalPage::ToBeFinalizedObject::Finalize() *J.J./third\_party/blink/renderer/platform/heap/heap\_page.cc*:1373:11  
#27 0x5596a3e89ae3 in blink::NormalPage::Sweep(blink::FinalizeType) *J.J./third\_party/blink/renderer/platform/heap/heap\_page.cc*:1464:16  
#28 0x5596a3e7fa6 in blink::BaseArena::SweepUnswepPage(blink::BasePage\*) *J.J./third\_party/blink/renderer/platform/heap/heap\_page.cc*:290:31  
#29 0x5596a3e8196b in blink::BaseArena::CompleteSweep() *J.J./third\_party/blink/renderer/platform/heap/heap\_page.cc*:389:5  
#30 0x5596a3e6ede4 in blink::ThreadHeap::CompleteSweep() *J.J./third\_party/blink/renderer/platform/heap/heap.cc*:486:17  
#31 0x5596a3e9ec89 in blink::ThreadState::CompleteSweep() *J.J./third\_party/blink/renderer/platform/heap/thread\_state.cc*:79:12  
#32 0x5596a3e0f2e in blink::ThreadState::ScheduleForcedGCForTesting() *J.J./third\_party/blink/renderer/platform/heap/thread\_state.cc*:441:3  
#33 0x5596b1713cfc in blink::V8GCController::GcEpilogue(v8::Isolate\*, v8::GCType, v8::GCCallbackFlags)  
*J.J./third\_party/blink/renderer/bindings/core/v8/v8\_gc\_controller.cc*:166:29  
#34 0x5596a21a8334 in CallGCEpilogueCallbacks *J.J./v8/src/heap/heap.cc*:2182:7  
#35 0x5596a21a8334 in v8::internal::Heap::PerformGarbageCollection(v8::internal::GarbageCollector, v8::GCCallbackFlags) *J.J./v8/src/heap/heap.cc*:2073:7  
#36 0x5596a21a051f in v8::internal::Heap::CollectGarbage(v8::internal::AllocationSpace, v8::internal::GarbageCollectionReason, v8::GCCallbackFlags)  
*J.J./v8/src/heap/heap.cc*:1571:11  
#37 0x5596a21a418e in CollectAllGarbage *J.J./v8/src/heap/heap.cc*:1293:3  
#38 0x5596a21a418e in v8::internal::Heap::PreciseCollectAllGarbage(int, v8::internal::GarbageCollectionReason, v8::GCCallbackFlags) *J.J./v8/src/heap/heap.cc*:1429:3  
#39 0x5596a20f66e0 in InvokeGC *J.J./v8/src/extensions/gc-extension.cc*:86:13  
#40 0x5596a20f66e0 in v8::internal::GCExtension::GC(v8::FunctionCallbackInfo<v8::Value> const&) *J.J./v8/src/extensions/gc-extension.cc*:135:5  
#41 0x5596a1df32d9 in v8::internal::FunctionCallbackArguments::Call(v8::internal::CallHandlerInfo) *J.J./v8/src/api/api-arguments-inl.h*:158:3  
#42 0x5596a1df0e4b in v8::internal::MaybeHandle<v8::internal::Object> v8::internal::(anonymous namespace)::HandleApiCallHelper<false>(v8::internal::Isolate\*, v8::internal::Handle<v8::internal::HeapObject>, v8::internal::Handle<v8::internal::HeapObject>, v8::internal::Handle<v8::internal::FunctionTemplateInfo>, v8::internal::Handle<v8::internal::Object>, v8::internal::BuiltinArguments) *J.J./v8/src/builtins/builtins-api.cc*:111:36  
#43 0x5596a1deec9d in v8::internal::Builtin\_Impl\_HandleApiCall(v8::internal::BuiltinArguments, v8::internal::Isolate\*) *J.J./v8/src/builtins/builtins-api.cc*:141:5  
#44 0x5596a3d942d7 in Builtins\_CEntry\_Return1\_DontSaveFPRegs\_ArgvOnStack\_BuiltinExit ???:0  
#45 0x5596a3d1e1aa in Builtins\_InterpreterEntryTrampoline ???:0  
#46 0x5596a3d1b979 in Builtins\_JSEntryTrampoline ???:0  
#47 0x5596a3d1b757 in Builtins\_JSEntry ???:0  
#48 0x5596a2078342 in Call *J.J./v8/src/execution/simulator.h*:142:12  
#49 0x5596a2078342 in v8::internal::(anonymous namespace)::Invoke(v8::internal::Isolate\*, v8::internal::(anonymous namespace)::InvokeParams const&)  
*J.J./v8/src/execution/execution.cc*:266:33  
#50 0x5596a207755e in v8::internal::Execution::Call(v8::internal::Isolate\*, v8::internal::Handle<v8::internal::Object>, v8::internal::Handle<v8::internal::Object>, int, v8::internal::Handle<v8::internal::Object>\*) *J.J./v8/src/execution/execution.cc*:360:10  
#51 0x5596a1c99f19 in v8::Script::Run(v8::Local<v8::Context>) *J.J./v8/src/api/api.cc*:2155:7  
#52 0x5596b00c4b57 in blink::V8ScriptRunner::RunCompiledScript(v8::Isolate\*, v8::Local<v8::Script>, blink::ExecutionContext\*)  
*J.J./third\_party/blink/renderer/bindings/core/v8/v8\_script\_runner.cc*:341:22  
#53 0x5596b16a73ae in blink::ScriptController::ExecuteScriptAndReturnValue(v8::Local<v8::Context>, blink::ScriptSourceCode const&, blink::KURL const&, blink::SanitizeScriptErrors, blink::ScriptFetchOptions const&) *J.J./third\_party/blink/renderer/bindings/core/v8/script\_controller.cc*:133:20  
#54 0x5596b16a9f1e in blink::ScriptController::EvaluateScriptInMainWorld(blink::ScriptSourceCode const&, blink::KURL const&, blink::SanitizeScriptErrors, blink::ScriptFetchOptions const&, blink::ScriptController::ExecuteScriptPolicy) *J.J./third\_party/blink/renderer/bindings/core/v8/script\_controller.cc*:360:33

previously allocated by thread T0 (chrome) here:

#0 0x55969bec348d in malloc */b/swarming/w/ir/cache/builder/src/third\_party/lvm/compiler-rt/lib/asan/asan\_malloc\_linux.cpp*:145:3  
#1 0x5596ab321b2d in PartitionAllocGenericFlags *J.J./base/allocator/partition\_allocator/partition\_alloc.h*:402:48  
#2 0x5596ab321b2d in Alloc *J.J./base/allocator/partition\_allocator/partition\_alloc.h*:437:10  
#3 0x5596ab321b2d in WTF::Partitions::FastMalloc(unsigned long, char const\*) *J.J./third\_party/blink/renderer/platform/wtf/allocator/partitions.cc*:232:33  
#4 0x5596af7da2c7 in operator new *J.J./third\_party/blink/renderer/platform/wtf/thread\_safe\_ref\_counted.h*:54:3  
#5 0x5596af7da2c7 in blink::AudioBus::Create(unsigned int, unsigned int, bool) *J.J./third\_party/blink/renderer/platform/audio/audio\_bus.cc*:60:25  
#6 0x5596b5f74fa8 in blink::AudioNodeOutput::AudioNodeOutput(blink::AudioHandler\*, unsigned int)  
*J.J./third\_party/blink/renderer/modules/webaudio/audio\_node\_output.cc*:50:19  
#7 0x5596b5f7ed82 in make\_unique<blink::AudioNodeOutput, blink::AudioHandler\*, unsigned int&> *J.J./buildtools/third\_party/libc++/trunk/include/memory*:3043:32  
#8 0x5596b5f7ed82 in blink::AudioHandler::AddOutput(unsigned int) *J.J./third\_party/blink/renderer/modules/webaudio/audio\_node.cc*:202:7  
#9 0x5596b604b3a5 in blink::AudioWorkletHandler::AudioWorkletHandler(blink::AudioNode&, float, WTF::String, WTF::HashMap<WTF::String, scoped\_refptr<blink::AudioParamHandler>, WTF::PartitionAllocator>, blink::AudioWorkletNodeOptions const\*) *J.J./third\_party/blink/renderer/modules/webaudio/audio\_worklet\_node.cc*:56:5  
#10 0x5596b604be99 in blink::AudioWorkletHandler::Create(blink::AudioNode&, float, WTF::String, WTF::HashMap<WTF::String, scoped\_refptr<blink::AudioParamHandler>, WTF::PartitionAllocator>, WTF::StringHash, WTF::HashTraits<WTF::String>, WTF::HashTraits<scoped\_refptr<blink::AudioParamHandler>>, WTF::PartitionAllocator>, blink::AudioWorkletNodeOptions const\*) *J.J./third\_party/blink/renderer/modules/webaudio/audio\_worklet\_node.cc*:81:29  
#11 0x5596b6050308 in blink::AudioWorkletNode::AudioWorkletNode(blink::BaseAudioContext&, WTF::String const&, blink::AudioWorkletNodeOptions const\*, WTF::Vector<blink::CrossThreadAudioParamInfo, 0u, WTF::PartitionAllocator>, blink::MessagePort\*)  
*J.J./third\_party/blink/renderer/modules/webaudio/audio\_worklet\_node.cc*:246:14  
#12 0x5596b605345f in blink::AudioWorkletNode\* blink::MakeGarbageCollected<blink::AudioWorkletNode, blink::BaseAudioContext&, WTF::String const&, blink::AudioWorkletNodeOptions const&, WTF::Vector<blink::CrossThreadAudioParamInfo, 0u, WTF::PartitionAllocator> const, blink::MessagePort> (blink::BaseAudioContext&, WTF::String const&, blink::AudioWorkletNodeOptions const&, WTF::Vector<blink::CrossThreadAudioParamInfo, 0u, WTF::PartitionAllocator> const&&, blink::MessagePort&&) *J.J./third\_party/blink/renderer/platform/heap/heap.h*:535:30  
#13 0x5596b60511cc in blink::AudioWorkletNode::Create(blink::ScriptState\*, blink::BaseAudioContext\*, WTF::String const&, blink::AudioWorkletNodeOptions const\*, blink::ExceptionState&) *J.J./third\_party/blink/renderer/modules/webaudio/audio\_worklet\_node.cc*:317:28  
#14 0x5596b60661ca in Constructor */gen/third\_party/blink/renderer/bindings/modules/v8/v8\_audio\_worklet\_node.cc*:174:28  
#15 0x5596b60661ca in blink::audio\_worklet\_node\_v8\_internal::ConstructorCallback(v8::FunctionCallbackInfo<v8::Value> const&)  
*/gen/third\_party/blink/renderer/bindings/modules/v8/v8\_audio\_worklet\_node.cc*:200:3  
#16 0x5596a1df32d9 in v8::internal::FunctionCallbackArguments::Call(v8::internal::CallHandlerInfo) *J.J./v8/src/api/api-arguments-inl.h*:158:3  
#17 0x5596a1df01f2 in v8::internal::MaybeHandle<v8::internal::Object> v8::internal::(anonymous namespace)::HandleApiCallHelper<true>(v8::internal::Isolate\*, v8::internal::Handle<v8::internal::HeapObject>, v8::internal::Handle<v8::internal::HeapObject>, v8::internal::Handle<v8::internal::FunctionTemplateInfo>, v8::internal::Handle<v8::internal::Object>, v8::internal::BuiltinArguments) *J.J./v8/src/builtins/builtins-api.cc*:111:36  
#18 0x5596a1deec4d in v8::internal::Builtin\_Impl\_HandleApiCall(v8::internal::BuiltinArguments, v8::internal::Isolate\*) *J.J./v8/src/builtins/builtins-api.cc*:137:5  
#19 0x5596a3d942d7 in Builtins\_CEntry\_Return1\_DontSaveFPRegs\_ArgvOnStack\_BuiltinExit ???:0  
#20 0x5596a3d19ce4 in Builtins\_JSBuiltinsConstructStub ???:0  
#21 0x5596a3e10f46 in Builtins\_ConstructHandler ???:0  
#22 0x5596a3d1e1aa in Builtins\_InterpreterEntryTrampoline ???:0  
#23 0x5596a3d177be in Builtins\_ArgumentsAdaptorTrampoline ???:0  
#24 0x5596a3d722f3 in Builtins\_PromiseFulfillReactionJob ???:0  
#25 0x5596a3d3fc4a in Builtins\_RunMicrotasks ???:0  
#26 0x5596a3d1b8d7 in Builtins\_JSRunMicrotasksEntry ???:0  
#27 0x5596a2077f82 in Call *J.J./v8/src/execution/simulator.h*:142:12  
#28 0x5596a2077f82 in v8::internal::(anonymous namespace)::Invoke(v8::internal::Isolate\*, v8::internal::(anonymous namespace)::InvokeParams const&)  
*J.J./v8/src/execution/execution.cc*:281:33  
#29 0x5596a207a0b7 in v8::internal::(anonymous namespace)::InvokeWithTryCatch(v8::internal::Isolate\*, v8::internal::(anonymous namespace)::InvokeParams const&)  
*J.J./v8/src/execution/execution.cc*:326:20

```

#30 0x5596a207a4a9 in v8::internal::Execution::TryRunMicrotasks(v8::internal::Isolate*, v8::internal::MicrotaskQueue*, v8::internal::MaybeHandle<v8::internal::Object*>)
J.J./v8/src/execution/execution.cc:405:10
#31 0x5596a20e88b2 in v8::internal::MicrotaskQueue::RunMicrotasks(v8::internal::Isolate*) J.J./v8/src/execution/microtask-queue.cc:164:22
#32 0x5596a409bc03 in blink::scheduler::MainThreadSchedulerImpl::OnTaskCompleted(base::WeakPtr<blink::scheduler::MainThreadTaskQueue>,
base::sequence_manager::Task const&, base::sequence_manager::TaskQueue::TaskTiming*, base::sequence_manager::LazyNow*)
J.J./third_party/blink/renderer/platform/scheduler/main_thread/main_thread_scheduler_impl.cc:2446:3
#33 0x5596a40a871c in blink::scheduler::MainThreadTaskQueue::OnTaskCompleted(base::sequence_manager::Task const&,
base::sequence_manager::TaskQueue::TaskTiming*, base::sequence_manager::LazyNow*)
J.J./third_party/blink/renderer/platform/scheduler/main_thread/main_thread_task_queue.cc:194:29
#34 0x5596a5dcacf3b in
base::sequence_manager::internal::SequenceManagerImpl::NotifyDidProcessTask(base::sequence_manager::internal::SequenceManagerImpl::ExecutingTask*,
base::sequence_manager::LazyNow*) J.J./base/task/sequence_manager/sequence_manager_impl.cc:838:35
#35 0x5596a5dcad89 in base::sequence_manager::internal::SequenceManagerImpl::DidRunTask()
J.J./base/task/sequence_manager/sequence_manager_impl.cc:678:3

Thread T20 (AudioWorklet th) created by T0 (chrome) here:
#0 0x55969beadd1a in pthread_create(b/swarming/wlr/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_interceptors.cpp:214:3
#1 0x5596a5f197ce in base::(anonymous namespace)::CreateThread(unsigned long, bool, base::PlatformThread::Delegate*, base::PlatformThreadHandle*,
base::ThreadPriority) J.J./base/threading/platform_thread_posix.cc:120:13
#2 0x5596a5e3dca3 in base::SimpleThread::StartAsync() J.J./base/threading/simple_thread.cc:51:13
#3 0x5596a40c2d65 in blink::scheduler::WorkerThread::Init() J.J./third_party/blink/renderer/platform/scheduler/worker/worker_thread.cc:61:12
#4 0x5596a402b84a in blink::Thread::CreateThread(blink::ThreadCreationParams const&) J.J./third_party/blink/renderer/platform/scheduler/common/thread.cc:82:11
#5 0x5596b3c64086 in blink::WorkerBackingThread::WorkerBackingThread(blink::ThreadCreationParams const&)
J.J./third_party/blink/renderer/core/workers/worker_backing_thread.cc:59:23
#6 0x5596b602f129 in make_unique<blink::WorkerBackingThread, const blink::ThreadCreationParams &>
J.J./buildtools/third_party/libc++/trunk/include/memory:3043:32
#7 0x5596b602f129 in blink::WorkletThreadHolder<blink::AudioWorkletThread>::EnsureInstance(blink::ThreadCreationParams const&)
J.J./third_party/blink/renderer/core/workers/worklet_thread_holder.h:34:9
#8 0x5596b6030106 in EnsureSharedBackingThread J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_thread.cc:67:3
#9 0x5596b6030106 in AudioWorkletThread J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_thread.cc:40:5
#10 0x5596b6030106 in blink::AudioWorkletThread::Create(blink::WorkerReportingProxy&)
J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_thread.cc:32:31
#11 0x5596b602b572 in blink::AudioWorkletMessagingProxy::CreateWorkerThread()
J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_messaging_proxy.cc:95:10
#12 0x5596b3c5dc8b in blink::ThreadedMessagingProxyBase::InitializeWorkerThread(std::____1:unique_ptr<blink::GlobalScopeCreationParams,
std::____1::default_delete<blink::GlobalScopeCreationParams> >, base::Optional<blink::WorkerBackingThreadStartupData> const&)
J.J./third_party/blink/renderer/core/workers/threaded_messaging_proxy_base.cc:71:20
#13 0x5596b60721b7 in blink::ThreadedWorkletMessagingProxy::Initialize(blink::WorkerClients*, blink::WorkletModuleResponsesMap*,
base::Optional<blink::WorkerBackingThreadStartupData> const&) J.J./third_party/blink/renderer/core/workers/threaded_worklet_messaging_proxy.cc:79:3
#14 0x5596b602925b in blink::AudioWorklet::CreateGlobalScope() J.J./third_party/blink/renderer/modules/webaudio/audio_worklet.cc:81:10
#15 0x5596b3c92f1b in blink::Worklet::FetchAndInvokeScript(blink::KURL const&, WTF::String const&, blink::WorkletPendingTasks*)
J.J./third_party/blink/renderer/core/workers/worklet.cc:164:24
#16 0x5596a5dbbf92 in Run J.J./base/callback.h:98:12
#17 0x5596a5dbbf92 in base::TaskAnnotator::RunTask(char const*, base::PendingTask*) J.J./base/task/common/task_annotator.cc:142:33
#18 0x5596a5d46c68 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoWorkImpl(base::sequence_manager::LazyNow*, bool*)
J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:365:23
#19 0x5596a5d4049 in base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::DoSomeWork()
J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:219:7
#20 0x5596a5cdf160 in base::MessagePumpDefault::Run(base::MessagePump::Delegate*) J.J./base/message_loop/message_pump_default.cc:39:55
#21 0x5596a5df646e in Run J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:463:12
#22 0x5596a5df646e in non-virtual thunk to base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run(bool, base::TimeDelta)
J.J./base/task/sequence_manager/thread_controller_with_message_pump_impl.cc:0:0
#23 0x5596a5d6b681 in base::RunLoop::Run() J.J./base/run_loop.cc:156:14
#24 0x5596b6f9b4cf in content::RendererMain(content::MainFunctionParams const&) J.J./content/renderer/renderer_main.cc:213:16
#25 0x5596a4d9d94f in content::RunZygote(content::ContentMainDelegate*) J.J./content/app/content_main_runner_impl.cc:492:14
#26 0x5596a4da0b0a in content::ContentMainRunnerImpl::Run(bool) J.J./content/app/content_main_runner_impl.cc:871:10
#27 0x5596a4f4392a in service_manager::Main(service_manager::MainParams const&) J.J./services/service_manager/embedder/main.cc:423:29
#28 0x5596a4d9be7f in content::ContentMain(content::ContentMainParams const&) J.J./content/app/content_main.cc:19:10
#29 0x55969bee7f73 in ChromeMain J.J./chrome/app/chrome_main.cc:110:12
#30 0x7f334b8852a in __libc_start_main ??:0:0

```

SUMMARY: AddressSanitizer: heap-use-after-free (/usr/local/google/home/hongchan/chromium/src/out/ASAN/chrome+0x22fd91c9)

Shadow bytes around the buggy address:

```

0x0c06801df980: fa fa fd fd fd fd fa fa fd fd fd fa fa fd fd
0x0c06801df990: fd fd fa fa fd fd fd fa fa fd fd fd fa fa fa
0x0c06801df9a0: fd fd fd fd fa fa fd fd fd fa fa fd fd fd fa
0x0c06801df9b0: fa fa fd fd fd fa fa fd fd fd fa fa fd fd
0x0c06801df9c0: fd fa fa fa fd fd fd fa fa fd fd fd fa fa
=>0x0c06801df9d0: fd fd fd fa fa fa fd [fd]fd fa fa fd fd fd fd
0x0c06801df9e0: fa fa fd fd fd fa fa fa fd fd fd fa fa 00 00
0x0c06801df9f0: 07 fa fa fa fd fd fa fa fa fd fd fd fa fa
0x0c06801dfa00: fd fd fd fd fa fa fd fd fd fa fa fd fd fd fd
0x0c06801dfa10: fa fa fd fd fd fa fa fd fd fd fa fa fd fd
0x0c06801dfa20: fd fd fa fa fd fd fd fa fa fd fd fd fd fa fa

```

Shadow byte legend (one shadow byte represents 8 application bytes):

```

Addressable: 00
Partially addressable: 01 02 03 04 05 06 07
Heap left redzone: fa
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack after return: f5
Stack use after scope: f8
Global redzone: f9
Global init order: f6
Poisoned by user: f7
Container overflow: fc
Array cookie: ac
Intra object redzone: bb
ASan internal: fe
Left alloca redzone: ca
Right alloca redzone: cb
Shadow gap: cc
==1==ABORTING

```

Comment 9 by hongchan@chromium.org on Wed, Dec 4, 2019, 1:51 PM EST

Components: Blink>JavaScript>GC

Comment 10 by haraken@google.com on Thu, Dec 5, 2019, 10:37 AM EST

> It seems like once GS sweep start marching it, there no way to stop it. I wish there is a way for GC to wait while the audio rendering doing the for-loop inside of Process(). This is a common theme of UAFs in the WebAudio.

Are you saying that you want to have a way to stop a GC of the main thread while the rendering thread (which is not attached to GC) is doing some operation? It is... a strange situation :)

It looks like that the crash trace is saying that the AudioBus object (on the main thread) is gone while the rendering thread is processing. Are we probably missing CrossThreadPersistent pointers to keep it alive?

Comment 11 by hongchan@chromium.org on Thu, Dec 5, 2019, 5:46 PM EST

Thanks for the advice! That was really helpful.

> Are you saying that you want to have a way to stop a GC of the main thread while the rendering thread (which is not attached to GC) is doing some operation? It is... a strange situation :)

Yeap. I know it is an absurd idea, but the unique multi-threading situation of WebAudio makes it easy to cause UAF. The root cause of all this is that we don't have a clear signal from the rendering thread when it is stopped. Ideally, we can hold everything with hasPendingActivity() until the audio thread is completely halted.

Also AudioBus is not a GC-ed object, so I tried `scoped_refptr<>` instead. It seems like working, but now I get a different crash. The trace as below:

```
==1==ERROR: AddressSanitizer: heap-use-after-free on address 0x611000111e18 at pc 0x557bcf11bbcc bp 0x7f1583fe6e10 sp 0x7f1583fe6e08
READ of size 8 at 0x611000111e18 thread T20 (AudioWorklet th)
#0 0x557bcf11bbcb in Get ../../third_party/blink/renderer/platform/heap/persistent.h:74:12
#1 0x557bcf11bbcb in operator-> ../../third_party/blink/renderer/platform/heap/persistent.h:84:34
#2 0x557bcf11bbcb in blink::AudioWorkletHandler::Process(unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc:119:9
#3 0x557bcf0490f in blink::AudioHandler::ProcessIfNecessary(unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node.cc:363:7
#4 0x557bcf043d3d in blink::AudioNodeOutput::Pull(blink::AudioBus*, unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node_output.cc:137:13
#5 0x557bcf064be6 in SumAllConnections ../../third_party/blink/renderer/modules/webaudio/audio_node_input.cc:128:40
#6 0x557bcf064be6 in blink::AudioNodeInput::Pull(blink::AudioBus*, unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node_input.cc:158:3
#7 0x557bcf1bf68c in blink::RealtimeAudioDestinationHandler::Render(blink::AudioBus*, unsigned int, blink::AudioOPosition const&, blink::AudioCallbackMetric const&)
../../third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc:204:39
#8 0x557bcf1dec43 in blink::AudioDestination::RequestRender(unsigned long, unsigned long, double, double, unsigned long)
../../third_party/blink/renderer/platform/audio/audio_destination.cc:247:17
...
```

So using `scoped_refptr<>` on AudioBus was effective, but now I can see is `CrossThreadPersistent<AudioWorkletProcess>` is gone:

[https://cs.chromium.org/chromium/src/third\\_party/blink/renderer/modules/webaudio/audio\\_worklet\\_node.cc?l=119](https://cs.chromium.org/chromium/src/third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc?l=119)

I thought holding an object with `CrossThreadPersistent<>` is good enough. Perhaps that's not the case here?

Comment 12 by haraken@google.com on Fri, Dec 6, 2019, 5:00 AM EST

> So using `scoped_refptr<>` on AudioBus was effective, but now I can see is `CrossThreadPersistent<AudioWorkletProcess>` is gone:

> [https://cs.chromium.org/chromium/src/third\\_party/blink/renderer/modules/webaudio/audio\\_worklet\\_node.cc?l=119](https://cs.chromium.org/chromium/src/third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc?l=119)

>

> I thought holding an object with `CrossThreadPersistent<>` is good enough. Perhaps that's not the case here?

It should be enough.

I'm confused...

- AudioWorkletHandler::Process is called on the rendering thread, which is not attached to Oilpan.
- AudioWorkletHandler::Process calls AudioWorkletProcessor::Process, which touches AudioWorkletProcessor::global\_scope\_ (Member).
- Why can the rendering thread (not attached to Oilpan) touch Member?

Comment 13 by hongchan@chromium.org on Fri, Dec 6, 2019, 2:12 PM EST

> AudioWorkletHandler::Process is called on the rendering thread, which is not attached to Oilpan.

In this case, the rendering thread is AudioWorkletThread, which is backed by WorkerBackingThread. All the worklets uses the same infrastructure.

> AudioWorkletHandler::Process calls AudioWorkletProcessor::Process, which touches AudioWorkletProcessor::global\_scope\_ (Member).

- AudioWorkletHandler is a non-Oilpan object, created by the main thread
- AudioWorkletProcessor is an Oilpan object, created by the Audio Worklet Thread.
- AudioWorkletHandler::Process() invoked by AudioWorkletThread for the graph rendering.

IIUC, there's no illegal access in this relationship. Please correct me if I am wrong.

> Why can the rendering thread (not attached to Oilpan) touch Member?

For the Audio Worklet operation, the rendering thread is AudioWorkletThread (WorkerBackingThread). So it is attached to Oilpan.

So I tried to avoid GC by returning always true from AudioWorkletNode::HasPendingActivity(). It didn't make any difference. AudioWorkletHandler is still getting destroyed while AudioWorkletNode is GCed.

```
==1==ERROR: AddressSanitizer: heap-use-after-free on address 0x6110000e3b58 at pc 0x563bf2461bcc bp 0x7f85ca07de10 sp 0x7f85ca07de08
READ of size 8 at 0x6110000e3b58 thread T20 (AudioWorklet th)
#0 0x563bf2461bcb in Get ../../third_party/blink/renderer/platform/heap/persistent.h:74:12
#1 0x563bf2461bcb in operator-> ../../third_party/blink/renderer/platform/heap/persistent.h:84:34
#2 0x563bf2461bcb in blink::AudioWorkletHandler::Process(unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc:119:9
#3 0x563bf239590f in blink::AudioHandler::ProcessIfNecessary(unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node.cc:363:7
#4 0x563bf2389d3d in blink::AudioNodeOutput::Pull(blink::AudioBus*, unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node_output.cc:137:13
#5 0x563bf23aabe6 in SumAllConnections ../../third_party/blink/renderer/modules/webaudio/audio_node_input.cc:128:40
#6 0x563bf23aabe6 in blink::AudioNodeInput::Pull(blink::AudioBus*, unsigned int) ../../third_party/blink/renderer/modules/webaudio/audio_node_input.cc:158:3
#7 0x563bf250568c in blink::RealtimeAudioDestinationHandler::Render(blink::AudioBus*, unsigned int, blink::AudioOPosition const&, blink::AudioCallbackMetric const&)
../../third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc:204:39
#8 0x563bf2524c43 in blink::AudioDestination::RequestRender(unsigned long, unsigned long, double, double, unsigned long)
../../third_party/blink/renderer/platform/audio/audio_destination.cc:247:17
...
```

freed by thread T0 (chrome) here:

```
#0 0x563bd82d734d in free /b/swarming/w/ir/cache/builder/src/third_party/lvm/compiler-rt/lib/asan/asan_malloc_linux.cpp:123:3
#1 0x563bf2397004 in DeleteInternal<blink::AudioHandler> ../../third_party/blink/renderer/platform/wtf/thread_safe_ref_counted.h:64:5
#2 0x563bf2397004 in Destruct ../../third_party/blink/renderer/platform/wtf/thread_safe_ref_counted.h:44:5
#3 0x563bf2397004 in Release ../../base/memory/ref_counted.h:407:7
#4 0x563bf2397004 in Release ../../base/memory/scoped_refptr.h:322:8
#5 0x563bf2397004 in ~scoped_refptr ../../base/memory/scoped_refptr.h:224:7
#6 0x563bf2397004 in reset ../../base/memory/scoped_refptr.h:254:18
#7 0x563bf2397004 in operator= ../../base/memory/scoped_refptr.h:240:5
#8 0x563bf2397004 in blink::AudioNode::~AudioNode() ../../third_party/blink/renderer/modules/webaudio/audio_node.cc:598:14
#9 0x563be029ccc3 in Finalize ../../third_party/blink/renderer/platform/heap/heap_page.cc:94:5
```

Perhaps there's some sorts of timeout for HasPendingActivity()?

Comment 14 by haraken@google.com on Sun, Dec 8, 2019, 8:07 PM EST

Would you confirm HasPendingActivity() is called before the AudioNode gets destructed? HasPendingActivity() is called only when the AudioNode has a wrapper.

If it's called and the AudioNode gets destructed, it should be a bug of the GC infrastructure...

[Comment 15](#) by [hongchan@chromium.org](#) on Mon, Dec 9, 2019, 1:02 PM EST

I see. I put some printf() and confirmed HasPendingActivity() is not getting called at all. I can see AudioWorkletNode destructor gets called. Not sure how to reason about this anymore.

[Comment 16](#) by [rtoy@chromium.org](#) on Mon, Dec 9, 2019, 6:41 PM EST

I added a bunch of prints (using DEBUG\_AUDIONODE\_REFERENCES 999 and others) and I see the following behavior:

```
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: AudioHandler::AudioHandler [ 1]
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: AudioHandler::AudioHandler() 0 [1] tot
al: 1
[ 0x7ebaa4cf1f58]: 0x7ebaa4cf2110: 1: AudioNode::AudioNode 0x6110000b2240
[ 0x7ebaa4cf1f58]: AudioContext::AudioContext(): 0 #1
[ 0x7ebaa4cf1f58]: AudioContext::HasPendingActivity 1 1
[ 0x7ebaa4cf1f58]: AudioContext::HasPendingActivity 1 1
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: AudioHandler::AudioHandler [ 1]
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: AudioHandler::AudioHandler() 0 [1] total: 2
[ 0x7ebaa4cf1f58]: 0x7ebaa4cf2da8: 21: AudioNode::AudioNode 0x6110000e6a40
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: AudioHandler::MakeConnection 1 [ 1] @0.4373333333333333
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: EnableOutputsIfNecessary: is_disabled 0 count 1 output size 1
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: AudioHandler::MakeConnection 1 [ 1] @0.4373333333333333
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: EnableOutputsIfNecessary: is_disabled 0 count 1 output size 0
[ 0x7ebaa4cf1f58]: AudioContext::Uninitialize
[ 0x7ebaa4cf1f58]: BaseAudioContext::Uninitialize dest init = 1
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: AudioHandler::BreakConnectionWithLock 0 [ 1] @0.4373333333333333
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: DisableOutputsIfNecessary is_disabled 0 count 0 tail 1
<another context is created>
[ 0x7ebaa4cf1f58]: 0x7ebaa4cf2da8: 21: AudioNode::dispose 0x6110000e6a40 @0.437333
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: AudioHandler::BreakConnectionWithLock 0 [ 2] @0.4373333333333333
[ 0x7ebaa4cf1f58]: 0x6110000b2240: 1: DisableOutputsIfNecessary is_disabled 0 count 0 tail 0
[ 0x7ebaa4cf1f58]: 0x6110000e6a40: 21: AudioHandler::~AudioHandler() 0 [0] remaining: 2
```

Ignoring the addresses, this is saying that the AudioContext is created, an AudioWorklet is created (the 21:) and is connected. Outputs are enabled.

Then at some point the AudioContext is being Uninitialize'd so we're destroying the context. Outputs are disabled and we're disposing of the AudioWorkletNode (21: AudioNode::dispose).

The AudioHandler for the node is destructed.

At this point UAF happens.

My guess is that the audio device is still rendering audio (because it hasn't shutdown yet) and pulls on the AudioWorkletGlobalScope, but things have already been destroyed.

[Comment 17](#) by [haraken@chromium.org](#) on Mon, Dec 9, 2019, 7:53 PM EST

Thanks for digging!

AudioContext should not get uninitialized while the rendering thread is running. If that's happening, it seems to be the cause of the UAF.

[Comment 18](#) by [rtoy@chromium.org](#) on Tue, Dec 10, 2019, 11:51 AM EST

How would the system know the rendering thread is running? We shut down the rendering thread when AudioContext::Uninitialize is called. My understanding is that even though we call Stop, the audio device can continue to call back for some time afterwards.

But there seem to be other odd things going on that I don't yet understand. More prints needed....

[Comment 19](#) by [hongchan@chromium.org](#) on Tue, Dec 10, 2019, 12:34 PM EST

So I made AudioContext::HasPendingActivity() return true. Now this happens:

```
ERROR: AddressSanitizer: heap-use-after-free on address 0x6040001a7a18 at pc 0x559f435433fd bp 0x7ffd1d9a48f0 sp 0x7ffd1d9a48e8
READ of size 4 at 0x6040001a7a18 thread T0 (chrome)
#0 0x559f435433fc in IsStatic J.J.J./third_party/blink/renderer/platform/wtf/text/string_impl.h:236:34
#1 0x559f435433fc in Release J.J.J./third_party/blink/renderer/platform/wtf/text/string_impl.h:292:10
#2 0x559f435433fc in Release J.J.J./base/memory/scoped_refptr.h:322:8
#3 0x559f435433fc in ~scoped_refptr J.J.J./base/memory/scoped_refptr.h:224:7
#4 0x559f435433fc in ~String J.J.J./third_party/blink/renderer/platform/wtf/text/wtf_string.h:61:18
#5 0x559f435433fc in ~InspectorHelperMixin J.J.J./third_party/blink/renderer/modules/webaudio/inspector_helper_mixin.h:22:35
#6 0x559f435433fc in blink::AudioNode::~AudioNode() J.J.J./third_party/blink/renderer/modules/webaudio/audio_node.cc:600:1
#7 0x559f3142aa23 in Finalize J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:94:5
#8 0x559f3142aa23 in blink::NormalPage::ToBeFinalizedObject::Finalize() J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:1373:11
#9 0x559f3142ac27 in blink::NormalPage::FinalizeSweep(blink::SweepResult) J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:1381:12
#10 0x559f31422f2b in blink::BaseArena::InvokeFinalizersOnSweptPages() J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:358:11
#11 0x559f31410cb4 in blink::ThreadHeap::InvokeFinalizersOnSweptPages() J.J.J./third_party/blink/renderer/platform/heap/heap.cc:492:17
#12 0x559f31440ae9 in SynchronizeAndFinishConcurrentSweeping J.J.J./third_party/blink/renderer/platform/heap/thread_state.cc:816:10
#13 0x559f31440ae9 in blink::ThreadState::CompleteSweep() J.J.J./third_party/blink/renderer/platform/heap/thread_state.cc:798:5
...(omitted)
```

0x6040001a7a18 is located 8 bytes inside of 48-byte region [0x6040001a7a10,0x6040001a7a40) freed by thread T0 (chrome) here:

```
#0 0x559f2948c18d in free /b/swarming/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/asan_malloc_linux.cpp:123:3
#1 0x559f43543339 in Release J.J.J./third_party/blink/renderer/platform/wtf/text/string_impl.h:304:7
#2 0x559f43543339 in Release J.J.J./base/memory/scoped_refptr.h:322:8
#3 0x559f43543339 in ~scoped_refptr J.J.J./base/memory/scoped_refptr.h:224:7
#4 0x559f43543339 in ~String J.J.J./third_party/blink/renderer/platform/wtf/text/wtf_string.h:61:18
#5 0x559f43543339 in ~InspectorHelperMixin J.J.J./third_party/blink/renderer/modules/webaudio/inspector_helper_mixin.h:22:35
#6 0x559f43543339 in blink::AudioNode::~AudioNode() J.J.J./third_party/blink/renderer/modules/webaudio/audio_node.cc:600:1
#7 0x559f436139dc in blink::AudioWorkletNode::~AudioWorkletNode() J.J.J./third_party/blink/renderer/modules/webaudio/audio_worklet_node.cc:369:14
#8 0x559f3142aa23 in Finalize J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:94:5
#9 0x559f3142aa23 in blink::NormalPage::ToBeFinalizedObject::Finalize() J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:1373:11
#10 0x559f3142ac27 in blink::NormalPage::FinalizeSweep(blink::SweepResult) J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:1381:12
#11 0x559f31422f2b in blink::BaseArena::InvokeFinalizersOnSweptPages() J.J.J./third_party/blink/renderer/platform/heap/heap_page.cc:358:11
#12 0x559f31410cb4 in blink::ThreadHeap::InvokeFinalizersOnSweptPages() J.J.J./third_party/blink/renderer/platform/heap/heap.cc:492:17
#13 0x559f31440ae9 in SynchronizeAndFinishConcurrentSweeping J.J.J./third_party/blink/renderer/platform/heap/thread_state.cc:816:10
#14 0x559f31440ae9 in blink::ThreadState::CompleteSweep() J.J.J./third_party/blink/renderer/platform/heap/thread_state.cc:798:5
...(omitted)
```

At least now I don't see AudioContext::Uninitialize(). Also the thread that's causing UAF is not Audio Worklet Thread anymore. This means that even if we fix the context not to call Uninitialize(), the UAF above might happen.

Comment 20 by [hongchan@chromium.org](mailto:hongchan@chromium.org) on Tue, Dec 10, 2019, 1:25 PM EST  
Cc: [scheib@chromium.org](mailto:scheib@chromium.org)

Comment 21 by [rtoy@chromium.org](mailto:rtoy@chromium.org) on Tue, Dec 10, 2019, 1:28 PM EST

Does this include your `scoped_refptr` changes? I'm using ToT and the backtrace I see is different. `AudioWorkletGlobalScope::Process` is running, called from the realtime destination.

Comment 22 by [hongchan@chromium.org](mailto:hongchan@chromium.org) on Tue, Dec 10, 2019, 2:09 PM EST

No. The trace above is from ToT, but I added `AudioWorkletNode()` destructor so I can print something.

Comment 23 by [hongchan@chromium.org](mailto:hongchan@chromium.org) on Tue, Dec 10, 2019, 2:13 PM EST

Edit: The trace above is from ToT with 2 changes: 1) `AudioContext::HasPendingActivity()` => true, 2) `~AudioWorkletNode()` to print something there.

Comment 24 by [rtoy@chromium.org](mailto:rtoy@chromium.org) on Tue, Dec 10, 2019, 3:56 PM EST

We've taken a closer look at what might be happening. Our current theory is that destination `Render()` is called to render the graph on the audio thread. At the same time, the context is uninitializing, and destroying objects that are being used.

We'll need to do a bit of debugging/printing to figure out how to handle this scenario.

Comment 25 by [rtoy@chromium.org](mailto:rtoy@chromium.org) on Tue, Dec 10, 2019, 4:55 PM EST

We've decided that we can probably add a render lock to `Uninitialize()`. The `Render()` method can use a `TryLock` so as not to block the audio thread and not to pull on the graph is `Uninitialize()` has the lock and is actively destroying things.

[hongchan@](mailto:hongchan@chromium.org) says this is working.

Comment 26 by [hongchan@chromium.org](mailto:hongchan@chromium.org) on Tue, Dec 10, 2019, 6:18 PM EST

Yeap. The fix seems to be effective. I let it run over 20 minutes without any crash. OTOH, the repro on ToT crashed almost instantly.

POC CL: <https://chromium-review.googlesource.com/c/chromium/src/+1960083>

Comment 27 by [bugdroid](mailto:bugdroid) on Thu, Dec 12, 2019, 12:52 PM EST

The following revision refers to this bug:  
<https://chromium.googlesource.com/chromium/src.git/+417a58a838349c46dfce49bba04a9e956142975c>

commit 417a58a838349c46dfce49bba04a9e956142975c  
Author: Hongchan Choi <[hongchan@chromium.org](mailto:hongchan@chromium.org)>  
Date: Thu Dec 12 17:50:59 2019

Introduce a Mutex for the rendering loop in `BaseAudioContext`

The render loop in Web Audio API is performed by the rendering thread, and this thread can access the data storage that is allocated by the main thread (Olpán). This relationship is prone to cause use-after-free error especially when the main thread objects gets collected.

This newly introduced mutex will be able to lock up the data storage when it is accessed by the render loop, so it can be protected even when the GC attempts to collect the object.

We believe the performance implication from the mutex would be negligible because it is locked only when `Uninitialize()` function gets called. The lock within the render loop uses `TryLock`, so it does not block the rendering thread.

Why introduces a new lock instead of using the existing graph lock?:

The graph lock is quite popular in various places in WebAudio, thus it is supposed to be very contentious. Each conflict will result in "silence" in the audio stream and we need to minimize such instance. This new lock is solely dedicated to the tear-down process, so we can guarantee that it will be locked from the main thread only once. Therefore, there is no risk causing redundant silence unless an `AudioContext` is getting collected.

the web tests without any problem.

~~Bug-1020462~~

Test: Locally confirmed that it does not repro anymore, and also passed  
Change-Id: I1fb7c302ff21c3d3ac763088a9d0bb4e8584b03f  
Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1960083>  
Reviewed-by: Raymond Toy <[rtoy@chromium.org](mailto:rtoy@chromium.org)>  
Reviewed-by: Kentaro Hara <[haraken@chromium.org](mailto:haraken@chromium.org)>  
Commit-Queue: Hongchan Choi <[hongchan@chromium.org](mailto:hongchan@chromium.org)>  
Cr-Commit-Position: refs/heads/master@{#724249}

[modify] [https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.cc](https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third_party/blink/renderer/modules/webaudio/base_audio_context.cc)  
[modify] [https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.h](https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third_party/blink/renderer/modules/webaudio/base_audio_context.h)  
[modify] [https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third\\_party/blink/renderer/modules/webaudio/offline\\_audio\\_destination\\_node.cc](https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third_party/blink/renderer/modules/webaudio/offline_audio_destination_node.cc)  
[modify] [https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third\\_party/blink/renderer/modules/webaudio/realtime\\_audio\\_destination\\_node.cc](https://crrev.com/417a58a838349c46dfce49bba04a9e956142975c/third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc)

Comment 28 by [rtoy@chromium.org](mailto:rtoy@chromium.org) on Fri, Dec 13, 2019, 10:39 AM EST

Labels: Merge-Request-80

Comment 29 by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Fri, Dec 13, 2019, 11:19 AM EST

Status: Fixed (was: Started)

Please mark security bugs as fixed as soon as the fix lands, and before requesting merges. This update is based on the merge-labels applied to this issue. Please reopen if this update was incorrect.

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

Comment 30 by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Sat, Dec 14, 2019, 10:32 AM EST

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

Comment 31 by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Sat, Dec 14, 2019, 10:41 AM EST

Labels: -Merge-Request-80 Merge-Approved-80 Hotlist-Merge-Approved



Your change meets the bar and is auto-approved for M80. Please go ahead and merge the CL to branch 3987 (refs/branch-heads/3987) manually. Please contact milestone owner if you have questions.

Merge instructions: <https://www.chromium.org/developers/how-tos/drover>

Owners: govind@ (Android), Kariahda@ (iOS), dgagnon@ (ChromeOS), srinivassista@ (Desktop)

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

**Comment 32** by gov...@chromium.org on Sun, Dec 15, 2019, 3:53 AM EST

Requesting to merge to M80 branch 3987 ASAP. Please use branch CQ for merge. Thank you.

**Comment 33** by bugdroid on Mon, Dec 16, 2019, 12:33 PM EST

**Labels:** -merge-approved-80 merge-merged-3987 merge-merged-80

The following revision refers to this bug:

<https://chromium.googlesource.com/chromium/src.git/+e63812084cb27d418749535749234979e1d8eb4f>

commit e63812084cb27d418749535749234979e1d8eb4f

Author: Hongchan Choi <hongchan@chromium.org>

Date: Mon Dec 16 17:32:24 2019

Introduce a Mutex for the rendering loop in BaseAudioContext

The render loop in Web Audio API is performed by the rendering thread, and this thread can access the data storage that is allocated by the main thread (Oilpan). This relationship is prone to cause use-after-free error especially when the main thread objects gets collected.

This newly introduced mutex will be able to lock up the data storage when it is accessed by the render loop, so it can be protected even when the GC attempts to collect the object.

We believe the performance implication from the mutex would be negligible because it is locked only when Uninitialize() function gets called. The lock within the render loop uses TryLock, so it does not block the rendering thread.

Why introduces a new lock instead of using the existing graph lock?:

The graph lock is quite popular in various places in WebAudio, thus it is supposed to be very contentious. Each conflict will result in "silence" in the audio stream and we need to minimize such instance. This new lock is solely dedicated to the tear-down process, so we can guarantee that it will be locked from the main thread only once. Therefore, there is no risk causing redundant silence unless an AudioContext is getting collected.

the web tests without any problem.

TBR=hongchan@chromium.org

(cherry picked from commit 417a58a838349c46dfce49bba04a9e956142975c)

~~Bug=4929462~~

Test: Locally confirmed that it does not repro anymore, and also passed

Change-Id: I1fb7c302ff21c3d3ac763088a9d0bb4e8584b03f

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1960083>

Reviewed-by: Raymond Toy <rtoy@chromium.org>

Reviewed-by: Kentaro Hara <haraken@chromium.org>

Commit-Queue: Hongchan Choi <hongchan@chromium.org>

Cr-Original-Commit-Position: refs/heads/master@(#724249)

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1969047>

Commit-Queue: Raymond Toy <rtoy@chromium.org>

Cr-Commit-Position: refs/branch-heads/3987@(#160)

Cr-Branched-From: c4e8da9871cc266be74481e212f3a5252972509d-refs/heads/master@(#722274)

[modify] [https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.cc](https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third_party/blink/renderer/modules/webaudio/base_audio_context.cc)

[modify] [https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.h](https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third_party/blink/renderer/modules/webaudio/base_audio_context.h)

[modify] [https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third\\_party/blink/renderer/modules/webaudio/offline\\_audio\\_destination\\_node.cc](https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third_party/blink/renderer/modules/webaudio/offline_audio_destination_node.cc)

[modify] [https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third\\_party/blink/renderer/modules/webaudio/realtime\\_audio\\_destination\\_node.cc](https://crrev.com/e63812084cb27d418749535749234979e1d8eb4f/third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc)

**Comment 34** by natashapabrai@google.com on Mon, Dec 16, 2019, 3:08 PM EST

**Labels:** reward-topanel

**Comment 35** by adetaylor@google.com on Wed, Dec 18, 2019, 3:54 PM EST

Sheriffbot didn't request merge to 79 because humans requested merge to just 80. Speaking as a human (mostly) I am going to request a merge to 79. I see no reason why we wouldn't want to ship this in a M79 stable refresh.

**Comment 36** by adetaylor@google.com on Wed, Dec 18, 2019, 3:54 PM EST

**Labels:** Merge-Request-79

**Comment 37** by sheriffbot@chromium.org on Wed, Dec 18, 2019, 3:59 PM EST

**Labels:** -Merge-Request-79 Hotlist-Merge-Review Merge-Review-79

This bug requires manual review: Request affecting a post-stable build

Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?  
- Chrome: <https://goto.google.com/chrome-release-branch-merge-guidelines>  
- 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 master/ToT?
4. Why are these changes required in this milestone after branch?
5. Is this a new feature?
6. If it is a new feature, is it behind a flag using finch?

Please contact the milestone owner if you have questions.

Owners: benmason@ (Android), kariahda@ (iOS), cindyb@ (ChromeOS), govind@ (Desktop)

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

**Comment 38** by gov...@chromium.org on Wed, Dec 18, 2019, 5:08 PM EST

**Cc:** [adetaylor@chromium.org](mailto:adetaylor@chromium.org) [pbomm...@chromium.org](mailto:pbomm...@chromium.org) [benmason@chromium.org](mailto:benmason@chromium.org)  
+[adetaylor@chromium.org](mailto:adetaylor@chromium.org) for M79 Merge review as part of security respin after holidays.

[Comment 39](#) by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Wed, Dec 18, 2019, 5:14 PM EST  
Yes, per [#c35](#) we should ship this.

[Comment 40](#) by [gov...@chromium.org](mailto:gov...@chromium.org) on Wed, Dec 18, 2019, 5:18 PM EST  
**NextAction:** 2020-01-02

Yes, saw [#35](#) :)

Requesting merge review after holidays.

[Comment 41](#) by [natashapabrai@google.com](mailto:natashapabrai@google.com) on Thu, Dec 19, 2019, 12:34 PM EST  
**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](mailto:security-vrp@chromium.org) with any questions.  
\*\*\*\*\*

[Comment 42](#) by [natashapabrai@google.com](mailto:natashapabrai@google.com) on Thu, Dec 19, 2019, 12:36 PM EST  
Congrats! The Panel decided to reward \$7,500 for this report!

[Comment 43](#) by [natashapabrai@google.com](mailto:natashapabrai@google.com) on Thu, Dec 19, 2019, 12:45 PM EST  
**Labels:** -reward-unpaid reward-inprocess

[Comment 44](#) Deleted

[Comment 45](#) by [cdsrc...@gmail.com](mailto:cdsrc...@gmail.com) on Thu, Dec 19, 2019, 9:56 PM EST

Thank u for the reward:).  
Can you change my credit name?  
New reporter credit name:  
Zhe Jin from cdsrc of Qihoo 360

Thanks~~

[Comment 46](#) by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Fri, Dec 20, 2019, 4:13 PM EST  
Hi, OK, I'll credit you like that for this bug and future bugs. Thanks for the report :)

[Comment 47](#) by [pbomm...@chromium.org](mailto:pbomm...@chromium.org) on Fri, Jan 3, 2020, 1:50 PM EST  
**Labels:** -Merge-Review-79 Merge-Approved-79

Checked the stability to make sure we don't have any stability spikes due to the on Beta, hence approving the CL to get merged to M79 Stable.

Please go ahead and merge the CL to M79 branch i.e., 3945 ASAP.

[Comment 48](#) by [bugdroid](mailto:bugdroid) on Mon, Jan 6, 2020, 11:07 AM EST  
**Labels:** -merge-approved-79 merge-merged-79 merge-merged-3945

The following revision refers to this bug:  
<https://chromium.googlesource.com/chromium/src.git/+d3fc0ed4452cff4ed813524f6b465b46ad4ec41d>

commit [d3fc0ed4452cff4ed813524f6b465b46ad4ec41d](https://chromium.googlesource.com/chromium/src.git/+d3fc0ed4452cff4ed813524f6b465b46ad4ec41d)  
Author: Hongchan Choi <[hongchan@chromium.org](mailto:hongchan@chromium.org)>  
Date: Mon Jan 06 16:06:27 2020

Introduce a Mutex for the rendering loop in BaseAudioContext

The render loop in Web Audio API is performed by the rendering thread, and this thread can access the data storage that is allocated by the main thread (Olpan). This relationship is prone to cause use-after-free error especially when the main thread objects gets collected.

This newly introduced mutex will be able to lock up the data storage when it is accessed by the render loop, so it can be protected even when the GC attempts to collect the object.

We believe the performance implication from the mutex would be negligible because it is locked only when Uninitialize() function gets called. The lock within the render loop uses TryLock, so it does not block the rendering thread.

Why introduces a new lock instead of using the existing graph lock?:

The graph lock is quite popular in various places in WebAudio, thus it is supposed to be very contentious. Each conflict will result in "silence" in the audio stream and we need to minimize such instance. This new lock is solely dedicated to the tear-down process, so we can guarantee that it will be locked from the main thread only once. Therefore, there is no risk causing redundant silence unless an AudioContext is getting collected.

the web tests without any problem.

(cherry picked from commit [417a58a838349c46dfce49bba04a9e956142975c](https://chromium.googlesource.com/chromium/src.git/+d3fc0ed4452cff4ed813524f6b465b46ad4ec41d))

[Bug-4020469](#)

Test: Locally confirmed that it does not repro anymore, and also passed  
Change-Id: [I1fb7c302ff21c3d3ac763088a9d0bb4e8584b03f](https://chromium.googlesource.com/chromium/src.git/+d3fc0ed4452cff4ed813524f6b465b46ad4ec41d)  
Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1960083>  
Reviewed-by: Raymond Toy <[rtoy@chromium.org](mailto:rtoy@chromium.org)>  
Reviewed-by: Kentaro Hara <[haraken@chromium.org](mailto:haraken@chromium.org)>  
Commit-Queue: Hongchan Choi <[hongchan@chromium.org](mailto:hongchan@chromium.org)>  
Cr-Original-Commit-Position: refs/heads/master@{#724249}

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1988157>  
Reviewed-by: Ben Mason <[benmason@chromium.org](mailto:benmason@chromium.org)>  
Cr-Commit-Position: refs/branch-heads/3945@(#1013)  
Cr-Branched-From: [e4635fff7defbae0f9c29e798349f6fc0cce4b1b](https://chromium-review.googlesource.com/c/chromium/src/+1988157)-refs/heads/master@(#706915)

[modify] [https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.cc](https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third_party/blink/renderer/modules/webaudio/base_audio_context.cc)  
[modify] [https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third\\_party/blink/renderer/modules/webaudio/base\\_audio\\_context.h](https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third_party/blink/renderer/modules/webaudio/base_audio_context.h)  
[modify] [https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third\\_party/blink/renderer/modules/webaudio/offline\\_audio\\_destination\\_node.cc](https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third_party/blink/renderer/modules/webaudio/offline_audio_destination_node.cc)  
[modify] [https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third\\_party/blink/renderer/modules/webaudio/realtime\\_audio\\_destination\\_node.cc](https://crrev.com/d3fc0ed4452cff4ed813524f6b465b46ad4ec41d/third_party/blink/renderer/modules/webaudio/realtime_audio_destination_node.cc)

[Comment 49](#) by [adetaylor@google.com](mailto:adetaylor@google.com) on Tue, Jan 7, 2020, 3:08 PM EST  
**Labels:** Release-2-M79

[Comment 50](#) by [mmoroz@chromium.org](mailto:mmoroz@chromium.org) on Tue, Jan 7, 2020, 5:06 PM EST  
**Labels:** VulnerabilityAnalysis-Requested

[hongchan@](#), thank you for fixing this issue. Chrome Security team needs your knowledge to prevent that whole class of bugs from happening elsewhere. We would greatly appreciate if you could tell us more about the issue by filling out the following form: <https://forms.gle/VWKDUv9a8GXCCRWm7>

[Comment 51](#) by [hongchan@chromium.org](#) on Wed, Jan 8, 2020, 12:29 PM EST  
Re #50:  
Done!

[Comment 52](#) by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Wed, Jan 8, 2020, 7:48 PM EST  
**Labels:** CVE-2020-6377 CVE\_description-missing

[Comment 53](#) by [mmoroz@chromium.org](mailto:mmoroz@chromium.org) on Thu, Jan 9, 2020, 2:51 PM EST  
**Labels:** VulnerabilityAnalysis-Submitted

[Comment 54](#) by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Fri, Jan 10, 2020, 4:09 PM EST  
**Labels:** -CVE\_description-missing CVE\_description-submitted

[Comment 55](#) by [adetaylor@google.com](mailto:adetaylor@google.com) on Wed, Mar 4, 2020, 1:44 PM EST  
**Cc:** [achuith@chromium.org](mailto:achuith@chromium.org)

[Comment 56](#) by [sheriffbot](#) on Fri, Mar 20, 2020, 1:54 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