bugs chromium New issue Open issues Search chromium issue Sign in

☆ Starred by 4 users

Owner:	asully@chromium.org
CC:	adetaylor@chromium.org jsb...@chromium.org mek@chromium.org pwnall@chromium.org ayui@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>Storage>FileAPI
Modified:	Jul 29, 2022
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Android
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Severity-Medium
allpublic
Unreproducible
CVE_description-submitted
Target-97
M-97
FoundIn-96
Security_Impact-Extended
merge-merged-4664
Merge-Merged-96
merge-merged-4692
merge-merged-97
Release-0-M97
CVE-2022-0115

Issue 1268903: Security: Use of uninitialized on-stack pointer in storage::BlobBuilderFromStream

Reported by markbrand@google.com on Wed, Nov 10, 2021, 11:52 AM EST Project Member

Code

VULNERABILITY DETAILS
The interface blink.mojom.BlobRegistry has a method RegisterFromStream that takes a DataPipeConsumerHandle:

https://source.chromium.org/chromium/chromium/src/+/main:third_party/blink/public/mojom/blob/blob_registry.mojom;l=43

```
RegisterFromStream(string content_type, string content_disposition,
                   uint64 length_hint,
                   handle<data_pipe_consumer> data,
                   pending_associated_remote<ProgressClient>? progress_client)
    => (SerializedBlob? blob);
```

Mojo handles are interchangeable and opaque; if you have the wrong handle type then it may behave in an unexpected way when used. The "pipe consumer handle" is read from in the following code:

https://source.chromium.org/chromium/chromium/src/+/main:storage/browser/blob/blob_builder_from_stream.cc;l=90

```
void DataPipeReady(MojoResult result, const mojo::HandleSignalsState& state) {
  while (current_offset_ < max_bytes_to_read_) {
    const void* data;
    uint32_t size;
    result = pipe_->BeginReadData(&data, &size, MOJO_READ_DATA_FLAG_NONE);
    if (result == MOJO_RESULT_SHOULD_WAIT) {
      watcher_.ArmOrNotify();
      return;
    }

    if (result == MOJO_RESULT_FAILED_PRECONDITION) {
      // Pipe has closed, so we must be done.
      pipe_.reset();
      break;
    }

    DCHECK_EQ(MOJO_RESULT_OK, result);
    size = std::min<uint64_t>(size, max_bytes_to_read_ - current_offset_);
    if (!Populate(base::make_span(static_cast<const char*>(data), size),
                  current_offset_)) {
      InvokeDone(mojo::ScopedDataPipeConsumerHandle(), PassProgressClient(),
                 false, current_offset_);
      delete this;
      return;

    }
    if (progress_client_)
      progress_client_->OnProgress(size);
      progress_client_->OnProgress(size);
    current_offset_ += size;
    result = pipe_->EndReadData(size);
    DCHECK_EQ(MOJO_RESULT_OK, result);
  }

  // Either the pipe closed, or we filled the entire item.
  InvokeDone(std::move(pipe_), PassProgressClient(), true, current_offset_);
  delete this;
}
```

If pipe_ is any handle type other than DataPipeConsumer, the call to
BeginReadData will return MOJO_RESULT_INVALID_ARGUMENT, and will not set the
output arguments data and size, leaving them uninitialized. This case is not
handled, so we will continue and call Populate filling the blob data contents
with data read from the uninitialized pointer.

This is slightly unintuitive, since it seems like we should never reach
DataPipeReady if pipe_ is not the correct type.

However, DataPipeReady is called when the mojo::SimpleWatcher triggers:

https://source.chromium.org/chromium/chromium/src/+/main:storage/browser/blob/blob_builder_from_stream.cc;drc=a45490cf8d0fa082b53cfff5649e703075e2b933;l=71

```
watcher_.Watch(pipe_.get(), MOJO_HANDLE_SIGNAL_READABLE,
               MOJO_WATCH_CONDITION_SATISFIED,
               base::BindRepeating(&DataPipeConsumerHelper::DataPipeReady,
                                   base::Unretained(this)));
watcher_.ArmOrNotify();
```

There are multiple handle types that can trigger the event
MOJO_HANDLE_SIGNAL_READABLE, and MessagePipeDispatcher for example can be used
here; the attached PoC uses a MessagePipe handle to trigger this bug.

But; there's another detail in the use of mojo traps that's handled incorrectly
here; a trap is triggered under multiple different situations; not just when
the configured condition is met: see the definition of `MojoTrapEvent`:

https://source.chromium.org/chromium/chromium/src/+/main:mojo/public/c/system/trap.h;drc=65c9257f1777731d6d0669598f6fe6fe65fa61d3;l=26

```
// Structure passed to trap event handlers when invoked by a tripped trap.
struct MOJO_ALIGNAS(8) MojoTrapEvent {
  // The size of this structure, used for versioning.
  uint32_t struct_size;

  // May take on some combination of the following values:
  //
  //   |MOJO_TRAP_EVENT_FLAG_NONE|: No flags.
  //

  //   |MOJO_TRAP_EVENT_FLAG_WITHIN_API_CALL|: The trap was tripped within the
  //       extent of a user call to some Mojo API. This means that the event
  //       handler itself is re-entering user code. May happen, for example, if
  //       handler itself is re-entering user code. May happen, for example, if
  //       user code writes to an intra-process pipe and the receiving end trips
  //       a trap as a result. In that case the event handler executes within
  //       the extent of the |MojoWriteMessage()| call.
  MojoTrapEventFlags flags;

  // The context for the trigger which tripped the trap.
  MOJO_POINTER_FIELD(uintptr_t, trigger_context);

  // A result code indicating the cause of the event. May take on any of the
  // following values:
  //
  //   |MOJO_RESULT_OK|: The trigger's conditions were met.
  //   |MOJO_RESULT_FAILED_PRECONDITION|: The trigger's observed handle has
  //       changed state in such a way that the trigger's conditions can never
  //       be met again.
  //   |MOJO_RESULT_CANCELLED|: The trigger has been removed and will never
  //       cause another event to fire. This is always the last event fired by
  //       a trigger and it will fire when: the trigger is explicitly removed
  //       with |MojoRemoteTrigger()|, the trigger's owning trap handle is
  //       closed, or the handle observed by the trigger is closed.
  //
  //       Unlike the other result types above |MOJO_RESULT_CANCELLED| can
  //       fire even when the trap is disarmed.
  MojoResult result;

  // The last known signalling state of the trigger's observed handle at the
  // time the trap was tripped.
  struct MojoHandleSignalsState signals_state;
};
```

This means that if we request a trap on a condition that can never occur,
MojoArmTrap will immediately return MOJO_RESULT_FAILED_PRECONDITION, setting
result to MOJO_RESULT_FAILED_PRECONDITION instead of MOJO_RESULT_OK.

mojo::SimpleWatcher doesn't specially handle the latter case, and will forward
the result and signal state on to the calling code to process (which is
probably reasonable, and necessary, since the same thing would be expected if
we had a race here on handle closure). It's the responsibility of the user of
mojo::SimpleWatcher to check that the result callback parameter is
MOJO_RESULT_OK before assuming that the trap fired, and wasn't canceled due to
the state of the watched handle changing.

DataPipeConsumerHelper::DataPipeReady doesn't check the value of result or
state, so it misinterprets this error state as pipe_ instead being ready for
reading. This is a fairly common pattern,

Automatic stack variable initialization should prevent this issue from being
exploitable on all platforms where it is enabled; I think that is currently
official builds for all platforms other than Android.

This is probably exploitable on Android to leak arbitrary browser process
memory into a compromised renderer.

This bug is subject to a 90-day disclosure deadline. If a fix for this
issue is made available to users before the end of the 90-day deadline,
this bug report will become public 30 days after the fix was made
available. Otherwise, this bug report will become public at the deadline.
The scheduled deadline is 2022-02-08.

VERSION

Tested on Chromium 97.0.4677.0 (Developer Build) (64-bit)
Operating System: Crashes on Linux & Android

REPRODUCTION CASE

To reproduce you need a local build of chrome; run the attached script

```
$ python ./copy_mojo_js_bindings.py /path/to/chrome/.../out/Asan/gen
$ python -m http.server&
$ out/Asan/chrome --enable-blink-features=MojoJS --user-data-dir=/tmp/nonexist 'http://localhost:8000/index.html'
```

FOR CRASHES, PLEASE INCLUDE THE FOLLOWING ADDITIONAL INFORMATION

Type of crash: browser
Crash State: (Linux, so with uninitialized stack variables initialized to `\xaa`)

```
Received signal 11 <unknown> 458620021d6a8
#0 0x55e8b9475139 base::debug::CollectStackTrace()
#1 0x55e8b93dec43 base::debug::StackTrace::StackTrace()
#2 0x55e8b9474c11 base::debug::(anonymous namespace)::StackDumpSignalHandler()
   #0 0x7f154a6d98e0 in __funlockfile :?
   #1 0x7f154a6d98e0 in ?? ??:0
   #2 0x7f154994438f in ?? ./string/../sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S:387:0
#5 0x55e8bb62e7b4 storage::BlobBuilderFromStream::WritePipeToFutureDataHelper::Populate()
#6 0x55e8bb62e942 storage::(anonymous namespace)::DataPipeConsumerHelper::DataPipeReady()
#7 0x55e8b97ce5ae mojo::SimpleWatcher::OnHandleReady()
#8 0x55e8b97ce95a mojo::SimpleWatcher::Context::Notify()
#9 0x55e8b97cde10 mojo::SimpleWatcher::Context::CallNotify()
#10 0x55e8b6dc99db mojo::core::WatcherDispatcher::InvokeWatchCallback()
#11 0x55e8b6dc92a0 mojo::core::Watch::InvokeCallback()
#12 0x55e8b6dc5aff mojo::core::RequestContext::~RequestContext()
#13 0x55e8b6dbc6e6 mojo::core::NodeChannel::OnChannelMessage()
#14 0x55e8b6dad85f mojo::core::Channel::TryDispatchMessage()
#15 0x55e8b6dad553 mojo::core::Channel::OnReadComplete()
#16 0x55e8b6dce99d mojo::core::ChannelPosix::OnFileCanReadWithoutBlocking()
#17 0x55e8b94aaeb0 base::MessagePumpLibevent::OnLibeventNotification()
#18 0x55e8b95ee13d event_base_loop
#19 0x55e8b94ab211 base::MessagePumpLibevent::Run()
#20 0x55e8b9449562 base::sequence_manager::internal::ThreadControllerWithMessagePumpImpl::Run()
#21 0x55e8b9419c7d base::RunLoop::Run()
#22 0x55e8b945f998 base::Thread::Run()
#23 0x55e8b6fdde40 content::BrowserProcessIOThread::IOThreadRun()
#24 0x55e8b945fb22 base::Thread::ThreadMain()
#25 0x55e8b9485e9c base::(anonymous namespace)::ThreadFunc()
#26 0x7f154a6ceeae start_thread
#27 0x7f15498dfa5f clone

  r8: 00000c780042e003  r9: 0000000000000258 r10: 0000000000000baf r11: 000000000000031d
 r12: 0000000000000400 r13: aaaaaaaaaaaaaaaa r14: aaaaaaaaaaaaaaaa r15: 0000000000000000
  di: 00000c7801c50000  si: aaaaaaaaaaaaaaaa  bp: 00007f1542a57a90  bx: 00000c7801c11aa0
  di: 00000c7801c50000  si: aaaaaaaaaaaaaaaa  bp: 00007f1542a57a90  bx: 00000c7801c11aa0
  dx: 0000000000000400  ax: 00000c7801c50000  cx: 0000000000000400  sp: 00007f1542a57a58
  ip: 00007f154994438f efl: 0000000000010217 cgf: 002b000000000033 erf: 0000000000000000
 trp: 000000000000000d msk: 0000000000000000 cr2: 0000000000000000
[end of stack trace]
```

CREDIT INFORMATION
Reporter credit: Mark Brand of Google Project Zero

index.html
912 bytes View Download

copy_mojo_js_bindings.py
512 bytes View Download

Comment 1 by ClusterFuzz on Wed, Nov 10, 2021, 11:54 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5788860900245504.

Comment 2 by ClusterFuzz on Thu, Nov 11, 2021, 4:04 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=4896869387665408.

Comment 3 by ClusterFuzz on Thu, Nov 11, 2021, 4:46 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5666633814966272.

Comment 4 by ClusterFuzz on Thu, Nov 11, 2021, 5:11 AM EST Project Member

Detailed Report: https://clusterfuzz.com/testcase?key=5666633814966272

Fuzzer: None
Job Type: linux_asan_chrome_mp
Platform Id: linux

Crash Type: Null-dereference READ
Crash Address: 0x000000000380
Crash State:
  storage::BlobBuilderFromStream::WritePipeToFutureDataHelper::Populate
  storage::DataPipeConsumerHelper::DataPipeReady
  mojo::SimpleWatcher::OnHandleReady

Sanitizer: address (ASAN)

Crash Revision: https://clusterfuzz.com/revisions?job=linux_asan_chrome_mp&revision=940701

Reproducer Testcase: https://clusterfuzz.com/download?testcase_id=5666633814966272

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
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/5666633814966272 -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.

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

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

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

Comment 5 by ClusterFuzz on Thu, Nov 11, 2021, 8:43 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5742704530882560.

Comment 6 by ClusterFuzz on Thu, Nov 11, 2021, 9:32 AM EST Project Member

Labels: Unreproducible

ClusterFuzz testcase 5666633814966272 appears to be flaky, updating reproducibility label.

Comment 7 by ClusterFuzz on Thu, Nov 11, 2021, 11:44 AM EST Project Member

ClusterFuzz is analyzing your testcase. Developers can follow the progress at https://clusterfuzz.com/testcase?key=5702613091549184.

Comment 8 by tsepez@chromium.org on Thu, Nov 11, 2021, 5:34 PM EST Project Member

Owner: jianli@chromium.org
Components: Blink>Storage>FileAPI

Assigning per OWNERS, please re-assign as appropriate.

Comment 9 by sheriffbot on Fri, Nov 12, 2021, 2:28 PM EST Project Member

Status: Assigned (was: Unconfirmed)

Comment 10 by tsepez@chromium.org on Mon, Nov 15, 2021, 6:26 PM EST Project Member

Labels: Security_Severity-Medium

Comment 11 by sheriffbot on Tue, Nov 16, 2021, 1:19 PM EST Project Member

Labels: -Pri-3 Pri-1

Setting Pri-1 to match security severity Medium. 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 12 by sheriffbot on Wed, Nov 24, 2021, 12:21 PM EST Project Member

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

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

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

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

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

Comment 13 by sheriffbot on Mon, Dec 6, 2021, 11:10 AM EST Project Member

This issue hasn't been updated in the last 30 days - please update it or consider lowering its priority.

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

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

Comment 14 by adetaylor@google.com on Tue, Dec 7, 2021, 10:21 AM EST Project Member

Cc: pwnall@chromium.org mek@chromium.org

Comment 15 by adetaylor@google.com on Tue, Dec 7, 2021, 10:34 AM EST Project Member

Cc: adetaylor@chromium.org

Comment 16 by jianli@chromium.org on Tue, Dec 7, 2021, 12:12 PM EST Project Member

Owner: mek@chromium.org
Cc: -mek@chromium.org

Comment 17 by tsepez@chromium.org on Tue, Dec 7, 2021, 1:42 PM EST Project Member

Labels: FoundIn-96

I couldn't reproduce locally, likely due to configuration issues with my python, but blaming the code in question show little change since 2019, so setting extended stable.

Comment 18 by sheriffbot on Tue, Dec 7, 2021, 1:45 PM EST Project Member

Labels: Security_Impact-Extended

Comment 19 by pwnall@chromium.org on Tue, Dec 7, 2021, 3:02 PM EST Project Member

Owner: asully@chromium.org
Cc: mek@chromium.org ayui@chromium.org

asully@: Have you looked at this part of the code before?

cc-ing a bunch of Storage folks. mek@ is OOO until Jan 4 (see Monorail status message).

Comment 20 by sheriffbot on Wed, Dec 8, 2021, 12:52 PM EST Project Member

Labels: Target-97 M-97

Setting milestone and target because of medium severity.

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

Comment 21 by asully@chromium.org on Wed, Dec 8, 2021, 1:14 PM EST Project Member

I'm not familiar with this code and I also have not been able to repro this, but based on my understanding of the bug description the fix seems straightforward? (thank you markbrand@ for the detailed explanation!)

The first issue of `pipe_` being of the wrong type can be fixed by checking that result is MOJO_RESULT_INVALID_ARGUMENT after each BeginReadData call.

The second issue of `result` being ignored before being overwritten can be fixed by checking that MOJO_RESULT_OK before making any BeginReadData calls.

It looks like the [InvokeDone - delete - return] pattern is used on failure, so I'm assuming that's the behavior we want in both cases.

Anything I'm missing here? Should we bring in someone with more mojo expertise to confirm our thoughts here?

I noticed that `result` is also unused in OnPipeReady(), but it's unclear to me whether this is also problematic?
https://source.chromium.org/chromium/chromium/src/+/main:storage/browser/blob/blob_builder_from_stream.cc;l=27

```
  void DataPipeReady(MojoResult result, const mojo::HandleSignalsState& state) {
/* ====================== START NEW CODE ======================= */
/* ================== ADDRESSES SECOND ISSUE =================== */
    if (result != MOJO_RESULT_OK) {
      // We requested a trap on a condition that can never occur.
      // See ~~crbug.com/1268903~~.
      DCHECK(result == MOJO_RESULT_FAILED_PRECONDITION);
      InvokeDone(mojo::ScopedDataPipeConsumerHandle(), PassProgressClient(),
                 /*success=*/false, /*bytes_written=*/0);
      delete this;
      return;
    }
/* ======================= END NEW CODE ======================= */

    while (current_offset_ < max_bytes_to_read_) {
      const void* data;
      uint32_t size;
      result = pipe_->BeginReadData(&data, &size, MOJO_READ_DATA_FLAG_NONE);
/* ====================== START NEW CODE ======================= */
/* =================== ADDRESSES FIRST ISSUE =================== */

      if (result == MOJO_RESULT_INVALID_ARGUMENT) {
        // Mojo handles are interchangeable and opaque, so evidently `pipe_` is
        // not actually a ScopedDataPipeConsumerHandle
        // not actually a ScopedDataPipeConsumerHandle.
        InvokeDone(mojo::ScopedDataPipeConsumerHandle(), PassProgressClient(),
                   /*success=*/false, /*bytes_written=*/0);
        delete this;
        return;
      }
/* ======================== END NEW CODE ======================== */

      if (result == MOJO_RESULT_SHOULD_WAIT) {
        watcher_.ArmOrNotify();
        return;
      }

      if (result == MOJO_RESULT_FAILED_PRECONDITION) {
        // Pipe has closed, so we must be done.
        pipe_.reset();
        break;
      }
      DCHECK_EQ(MOJO_RESULT_OK, result);
      size = std::min<uint64_t>(size, max_bytes_to_read_ - current_offset_);
      if (!Populate(base::make_span(static_cast<const char*>(data), size),
                    current_offset_)) {
        InvokeDone(mojo::ScopedDataPipeConsumerHandle(), PassProgressClient(),
                   false, current_offset_);
        delete this;
        return;
      }
      if (progress_client_)
        progress_client_->OnProgress(size);
      current_offset_ += size;
      result = pipe_->EndReadData(size);
      DCHECK_EQ(MOJO_RESULT_OK, result);
    }

    // Either the pipe closed, or we filled the entire item.
    InvokeDone(std::move(pipe_), PassProgressClient(), true, current_offset_);
    delete this;
  }
```

Comment 22 by asully@chromium.org on Wed, Dec 8, 2021, 3:53 PM EST Project Member

Put up https://crrev.com/c/3324364 with these changes

Comment 23 by Git Watcher on Thu, Dec 9, 2021, 3:14 PM EST Project Member

The following revision refers to this bug:
 https://chromium.googlesource.com/chromium/src/+/07a2e4a07bb6e3f7435682fc924c843c6e89f494

commit 07a2e4a07bb6e3f7435682fc924c843c6e89f494
Author: Austin Sullivan <asully@chromium.org>
Date: Thu Dec 09 20:13:43 2021

Add checks in DataPipeReady

DataPipeReady is passed a MojoResult which it does not use. If `result`

DataPipeReady is passed a MojoResult which it does not use. If `result`
is MOJO_RESULT_FAILED_PRECONDITION, this will be improperly
interpreted as a closed pipe.

~~Bug: 1268903~~
Change-Id: I1cc7ca80686d761f6bd122f0cde2a70018c7bf50
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3324364
Reviewed-by: Victor Costan <pwnall@chromium.org>
Commit-Queue: Austin Sullivan <asully@chromium.org>
Cr-Commit-Position: refs/heads/main@{#950235}

[modify] https://crrev.com/07a2e4a07bb6e3f7435682fc924c843c6e89f494/storage/browser/blob/blob_builder_from_stream.cc

Comment 24 by asully@chromium.org on Thu, Dec 9, 2021, 3:45 PM EST Project Member

Status: Fixed (was: Assigned)

Marking fixed as per guidelines, since this is fixed on main (M98). Do we want to merge this back to M97?

Comment 25 by adetaylor@chromium.org on Thu, Dec 9, 2021, 4:04 PM EST Project Member

Labels: Merge-Request-97 Merge-Request-96

Adding merge requests to consider in a few days once this has had some bake time.

Comment 26 by markbrand@google.com on Fri, Dec 10, 2021, 9:55 AM EST Project Member

Sorry, didn't see this, I was off sick the last couple of days.

Fix looks good to me.

Comment 27 by sheriffbot on Fri, Dec 10, 2021, 1:41 PM EST Project Member

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

Comment 28 by sheriffbot on Fri, Dec 10, 2021, 3:17 PM EST Project Member

Labels: -Merge-Request-97 Hotlist-Merge-Review Merge-Review-97

Merge review required: M97 is already shipping to beta.

Please answer the following questions so that we can safely process your merge request:
1. Why does your merge fit within the merge criteria for these milestones?
- Chrome Browser: https://chromiumdash.appspot.com/branches
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. What changes specifically would you like to merge? Please link to Gerrit.
3. Have the changes been released and tested on canary?
4. Is this a new feature? If yes, is it behind a Finch flag and are experiments active in any release channels?
5. [Chrome OS only]: Was the change reviewed and approved by the Eng Prod Representative? https://goto.google.com/cros-engprodcomponents
6. If this merge addresses a major issue in the stable channel, does it require manual verification by the test team? If so, please describe required testing.

Please contact the milestone owner if you have questions.

Owners: benmason (Android), harrysouders (iOS), ceb (ChromeOS), pbommana (Desktop)

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

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

Comment 29 by sheriffbot on Fri, Dec 10, 2021, 3:17 PM EST Project Member

Labels: -Merge-Request-96 Merge-Review-96

Merge review required: M96 is already shipping to stable.

Please answer the following questions so that we can safely process your merge request:
1. Why does your merge fit within the merge criteria for these milestones?
- Chrome Browser: https://chromiumdash.appspot.com/branches
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. What changes specifically would you like to merge? Please link to Gerrit.
3. Have the changes been released and tested on canary?
4. Is this a new feature? If yes, is it behind a Finch flag and are experiments active in any release channels?
5. [Chrome OS only]: Was the change reviewed and approved by the Eng Prod Representative?
https://goto.google.com/cros-engprodcomponents
6. If this merge addresses a major issue in the stable channel, does it require manual verification by the test team? If so, please describe required testing.

Please contact the milestone owner if you have questions.
Owners: govind (Android), harrysouders (iOS), dgagnon (ChromeOS), srinivassista (Desktop)

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

Comment 30 by asully@chromium.org on Fri, Dec 10, 2021, 3:31 PM EST Project Member

1. Why does your merge fit within the merge criteria for these milestones?

Medium severity security issue

2. What changes specifically would you like to merge? Please link to Gerrit.

https://crrev.com/c/3324364

3. Have the changes been released and tested on canary?

Yes

4. Is this a new feature? If yes, is it behind a Finch flag and are experiments active in any release channels?

No

5. [Chrome OS only]: Was the change reviewed and approved by the Eng Prod Representative?
https://goto.google.com/cros-engprodcomponents

N/A

6. If this merge addresses a major issue in the stable channel, does it require manual verification by the test team? If so, please describe required testing.

No

Comment 31 by adetaylor@google.com on Mon, Dec 13, 2021, 12:46 PM EST Project Member

Labels: -Merge-Review-96 -Merge-Review-97 Merge-Approved-96 Merge-Approved-97

Approving merge to M96 (branch 4664) and M97 (branch 4692) assuming no problems have shown up in Canary.

Comment 32 by asully@chromium.org on Mon, Dec 13, 2021, 12:57 PM EST Project Member

Cc: jsb...@chromium.org

Created cherry-picks for M97 (https://crrev.com/c/3334500) and M96 (https://crrev.com/c/3335059)

+jsbell for bug visibility for OWNERs stamp (Victor is OOO)

Comment 33 by Git Watcher on Mon, Dec 13, 2021, 6:33 PM EST Project Member

Labels: -merge-approved-97 merge-merged-4692 merge-merged-97

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/5e03b6a7aaf1ab17745a2185b399da22c2426914

commit 5e03b6a7aaf1ab17745a2185b399da22c2426914
Author: Austin Sullivan <asully@chromium.org>
Date: Mon Dec 13 23:32:55 2021

M97: Add checks in DataPipeReady

DataPipeReady is passed a MojoResult which it does not use. If `result`
is MOJO_RESULT_FAILED_PRECONDITION, this will be improperly
interpreted as a closed pipe.

(cherry picked from commit 07a2e4a07bb6e3f7435682fc924c843c6e89f494)

~~Bug: 1268903~~
Change-Id: I1cc7ca80686d761f6bd122f0cde2a70018c7bf50
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3324364
Reviewed-by: Victor Costan <pwnall@chromium.org>
Commit-Queue: Austin Sullivan <asully@chromium.org>
Cr-Original-Commit-Position: refs/heads/main@{#950235}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3334500
Auto-Submit: Austin Sullivan <asully@chromium.org>
Reviewed-by: Joshua Bell <jsbell@chromium.org>
Commit-Queue: Joshua Bell <jsbell@chromium.org>
Cr-Commit-Position: refs/branch-heads/4692@{#953}
Cr-Branched-From: 038cd96142d384c0d2238973f1cb277725a62eba-refs/heads/main@{#938553}

[modify]
https://crrev.com/5e03b6a7aaf1ab17745a2185b399da22c2426914/storage/browser/blob/blob_builder_from_stream.cc

Comment 34 by Git Watcher on Tue, Dec 14, 2021, 12:04 PM EST Project Member

Labels: -merge-approved-96 merge-merged-4664 merge-merged-96

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/473c4c2cdc31b714bb56584a9f60c9e1b5bcd770

commit 473c4c2cdc31b714bb56584a9f60c9e1b5bcd770

Author: Austin Sullivan <asully@chromium.org>
Date: Tue Dec 14 17:03:55 2021

M96: Add checks in DataPipeReady

DataPipeReady is passed a MojoResult which it does not use. If `result`
is MOJO_RESULT_FAILED_PRECONDITION, this will be improperly
interpreted as a closed pipe.

(cherry picked from commit 07a2e4a07bb6e3f7435682fc924c843c6e89f494)

~~Bug: 1268903~~
Change-Id: I1cc7ca80686d761f6bd122f0cde2a70018c7bf50
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3324364
Reviewed-by: Victor Costan <pwnall@chromium.org>
Commit-Queue: Austin Sullivan <asully@chromium.org>
Cr-Original-Commit-Position: refs/heads/main@{#950235}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3335059
Auto-Submit: Austin Sullivan <asully@chromium.org>
Reviewed-by: Joshua Bell <jsbell@chromium.org>
Cr-Commit-Position: refs/branch-heads/4664@{#1304}
Cr-Branched-From: 24dc4ee75e01a29d390d43c9c264372a169273a7-refs/heads/main@{#929512}

[modify] https://crrev.com/473c4c2cdc31b714bb56584a9f60c9e1b5bcd770/storage/browser/blob/blob_builder_from_stream.cc

Comment 35 by amyressler@chromium.org on Tue, Jan 4, 2022, 12:01 PM EST Project Member

Labels: Release-0-M97

Comment 36 by amyressler@google.com on Tue, Jan 4, 2022, 1:35 PM EST Project Member

Labels: CVE-2022-0115 CVE_description-missing

Comment 37 by sheriffbot on Fri, Mar 18, 2022, 1:30 PM EDT Project Member

Labels: -Restrict-View-SecurityNotify allpublic

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

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

Comment 38 by amyressler@chromium.org on Fri, Jul 29, 2022, 5:36 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted