Starred by 1 user

Owner:	reillyg@chromium.org
CC:	reillyg@chromium.org
Status:	Fixed *(Closed)*
Components:	IO>USB
Modified:	Sep 21, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	1
Type:	Bug-Security

reward-0
Security_Severity-Low
Security_Impact-Stable
Arch-x86_64
Deadline-Exceeded
allpublic
Via-Wizard-Security
CVE_description-submitted
Target-76
Target-77
Target-78
Target-79
Target-80
Target-81
M-83
Target-83
Release-0-M85
CVE-2020-6569

Issue 995732: Potential out of bounds write vulnerability in webusb (usb_device_handle_usbfs.cc) (Linux 32bit)

Reported by guaix...@gmail.com on Tue, Aug 20, 2019, 9:03 AM EDT

Code

UserAgent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/537.36

Steps to reproduce the problem:
(1). After open the usb device from webusb. Renderer can call the function "DeviceImpl::IsochronousTransferIn" from mojo. The param "std::vector<uint32_t>& packet_lengths" can be controlled
https://cs.chromium.org/chromium/src/services/device/usb/mojo/device_impl.cc?dr&g=0&l=353

(2). A compromised renderer may send the malicious packet_lengths to browswer process, out of bounds write will occur (Linux 32 bit).

I tried to send a large vector(size = 0x15555556), but it didn't pass to the browser process, I'm not sure if there are other ways. May be a potential risk.

What is the expected behavior?

What went wrong?
DETAILS:

on Linux, it will call the function "UsbDeviceHandleUsbfs::IsochronousTransferIn"
https://cs.chromium.org/chromium/src/services/device/usb/usb_device_handle_usbfs.cc?dr&g=0&l=767

This function contains the wrong logic:

```
void UsbDeviceHandleUsbfs::IsochronousTransferIn ==> void UsbDeviceHandleUsbfs::IsochronousTransferInternal
      ----------------------------------------
      DCHECK_GE(buffer->size(), total_length);
      std::unique_ptr<Transfer> transfer(new (packet_lengths.size())
                                             Transfer(buffer, std::move(callback))); <==(1) if packet_lengths.size() > 0x15555555, it will overflow.The transfer will be
small size.
      transfer->urb.type = USBDEVFS_URB_TYPE_ISO;
      transfer->urb.endpoint = endpoint_address;
      transfer->urb.buffer_length = total_length;

      for (size_t i = 0; i < packet_lengths.size(); ++i)
       transfer->urb.iso_frame_desc[i].length = packet_lengths[i];              <==(2) oob write ( packet_lengths[i], i => 0x15555555)

https://cs.chromium.org/chromium/src/services/device/usb/usb_device_handle_usbfs.cc?dr&g=0&l=398
void* UsbDeviceHandleUsbfs::Transfer::operator new(
   std::size_t size,
   size_t number_of_iso_packets) {
  void* p = ::operator new(size + sizeof(usbdevfs_iso_packet_desc) *                  <=======   sizeof(usbdevfs_iso_packet_desc) == 12
                           number_of_iso_packets);                          <=======   number_of_iso_packets == packet_lengths.size()
                                                                            <=======   if number_of_iso_packets > 0x15555555, it will overflow, p will be small buffer

  Transfer* transfer = static_cast<Transfer*>(p);
  transfer->urb.number_of_packets = number_of_iso_packets;
```

```
 return p;
}
```

Did this work before? N/A

Chrome version: 77.0.3835.0 (32 bit) Channel: stable
OS Version:
Flash Version:

Comment 1 by mbarb...@chromium.org on Wed, Aug 21, 2019, 1:50 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Owner: reillyg@chromium.org
Labels: Security_Severity-High Security_Impact-Stable OS-Android OS-Chrome OS-Fuchsia OS-Mac OS-Windows
Components: IO>USB

reillyg: Would you mind taking a look at this?

Comment 2 by reillyg@chromium.org on Wed, Aug 21, 2019, 2:22 PM EDT Project Member

My initial thought on this is that the overflow is unreachable because it would require sending a packet_lengths array substantially larger than the maximum Mojo message size. As the reporter found, they were unable to get the browser process to receive such a message.

To avoid this problem the computation of this object size should be done with the utilities from checked_math.h.

Comment 3 by sheriffbot@chromium.org on Thu, Aug 22, 2019, 9:21 AM EDT Project Member

Labels: M-76 Target-76

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 4 by sheriffbot@chromium.org on Thu, Aug 22, 2019, 10:02 AM EDT Project Member

Labels: -Pri-2 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 5 by guaix...@gmail.com on Sun, Aug 25, 2019, 9:30 PM EDT

reillyg: Thanks for telling me the reason for this case.

Comment 6 by sheriffbot@chromium.org on Thu, Sep 5, 2019, 9:00 AM EDT Project Member

reillyg: 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 7 by sheriffbot@chromium.org on Wed, Sep 11, 2019, 9:00 AM EDT Project Member

Labels: -M-76 M-77 Target-77

Comment 8 by sheriffbot@chromium.org on Thu, Sep 19, 2019, 9:00 AM EDT Project Member

reillyg: Uh oh! This issue still open and hasn't been updated in the last 28 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 9 by drubery@chromium.org on Thu, Oct 17, 2019, 3:36 PM EDT Project Member

Friendly security sheriff ping - reillyg@, is there any update on this? Is there another person we could assign this to?

Comment 10 by reillyg@chromium.org on Thu, Oct 17, 2019, 5:08 PM EDT Project Member

Sorry for not putting an update on this issue. I recommend reducing the severity of this issue because as mentioned in comment #2 I don't think this case is reachable given Mojo message size limits. I will follow up to make sure that we start checking for overflow just in case but it has been low priority.

Comment 11 by sheriffbot@chromium.org on Sat, Oct 19, 2019, 10:48 AM EDT Project Member

Labels: Deadline-Exceeded

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

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

Comment 12 by sheriffbot@chromium.org on Wed, Oct 23, 2019, 9:11 AM EDT Project Member

Labels: -M-77 Target-78 M-78

Comment 13 by jdeblasio@chromium.org on Thu, Oct 24, 2019, 12:26 PM EDT Project Member

Labels: -Security_Severity-High Security_Severity-Low

Bumping down the severity, since it's probably not reachable, but we'd still love to make progress on this.

Comment 14 by sheriffbot@chromium.org on Wed, Dec 11, 2019, 9:12 AM EST Project Member

Labels: -M-78 Target-79 M-79

Comment 15 by sheriffbot@chromium.org on Wed, Feb 5, 2020, 10:48 AM EST Project Member

Labels: -M-79 M-80 Target-80

Comment 16 by sheriffbot on Thu, Apr 9, 2020, 12:29 PM EDT Project Member

Labels: -M-80 Target-81 M-81

Comment 17 by sheriffbot on Wed, May 20, 2020, 1:30 PM EDT Project Member

Labels: -M-81 M-83 Target-83

Comment 18 by bugdroid on Fri, Jun 12, 2020, 1:43 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/60945402f3a3fc907f38fe0548c02d6055184723

commit 60945402f3a3fc907f38fe0548c02d6055184723
Author: Reilly Grant <reillyg@chromium.org>
Date: Fri Jun 12 17:41:53 2020

[usb] Use checked math to calculate Transfer size

The size of the UsbDeviceHandleUsbfs::Transfer object depends on the number of isochronous packets that are requested. This change protects against integer overflow since the number of packets is controlled by script. Since the number of packets is also limited by the maximum Mojo message size this can be a CHECK rather than having code to handle overflow.

~~Bug: 995732~~
Change-Id: Ie64be2d8fb8c8c1e49c7f676fb81446fce77d984
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2242751
Auto-Submit: Reilly Grant <reillyg@chromium.org>
Commit-Queue: Ovidio de Jesús Ruiz-Henríquez <odejesush@chromium.org>
Reviewed-by: Ovidio de Jesús Ruiz-Henríquez <odejesush@chromium.org>
Cr-Commit-Position: refs/heads/master@{#777868}

[modify] https://crrev.com/60945402f3a3fc907f38fe0548c02d6055184723/services/device/usb/usb_device_handle_usbfs.cc

Comment 19 by reillyg@chromium.org on Fri, Jun 12, 2020, 1:46 PM EDT Project Member

Status: Fixed (was: Assigned)

Comment 20 by sheriffbot on Fri, Jun 12, 2020, 3:06 PM EDT Project Member

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

Comment 21 by natashapabrai@google.com on Mon, Jun 15, 2020, 3:27 PM EDT Project Member

Labels: reward-topanel

Comment 22 by natashapabrai@google.com on Wed, Jun 24, 2020, 7:22 PM EDT Project Member

Labels: -reward-topanel reward-0

Unfortunately the Panel declined to award this report.

Comment 23 by adetaylor@google.com on Mon, Aug 24, 2020, 1:38 PM EDT Project Member

Labels: Release-0-M85

Comment 24 by adetaylor@google.com on Mon, Aug 24, 2020, 2:25 PM EDT Project Member

guaixiaomei@gmail.com: how would you like to be credited in the Chrome release notes? Thanks for the report!

Comment 25 by adetaylor@google.com on Mon, Aug 24, 2020, 3:29 PM EDT Project Member

Labels: CVE-2020-6569 CVE_description-missing

Comment 26 by sheriffbot on Fri, Sep 18, 2020, 3:03 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 27 by adetaylor@google.com on Mon, Sep 21, 2020, 3:05 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted