Security: ConvertToJavaBitmap heap-buffer-overflow.

Fixed Vulnerability P1 + Add Hotlist Security_Impact-Stable CVE_description-submitted

STATUS UPDATE No update yet.

DESCRIPTION ma...@google.com created issue #1 Oct 31, 2020 12:41AM

NOTE: We have evidence that the following bug is being used in the wild. Therefore, this bug is subject to a 7 day disclosure deadline.

VULNERABILITY DETAILS
This affects Chrome on Android only; it is a bug in Chrome code for the Android platform.

This is reachable in the browser process from a compromised renderer; eg. in WebContentsViewAndroid::StartDragging
https://source.chromium.org/chromium/chromium/src/+/master:content/browser/web_contents/web_contents_view_android.cc;l=336

Which is reachable from RenderWidgetHostImpl::StartDragging
https://source.chromium.org/chromium/chromium/src/+/master:content/browser/renderer_host/render_widget_host_impl.cc;l=2548

See the code for ConvertToJavaBitmap. This calls CreateJavaBitmap with some parameters taken from the provided skbitmap, but it does not correctly handle all possible input skbitmaps. eg.

```
static int SkColorTypeToBitmapFormat(SkColorType color_type) {
switch (color_type) {
case kAlpha_8_SkColorType:
return BITMAP_FORMAT_ALPHA_8;
case kARGB_4444_SkColorType:
return BITMAP_FORMAT_ARGB_4444;
case kN32_SkColorType:
return BITMAP_FORMAT_ARGB_8888;
case kRGB_565_SkColorType:
return BITMAP_FORMAT_RGB_565;
case kUnknown_SkColorType:
default:
NOTREACHED();
return BITMAP_FORMAT_NO_CONFIG;
}
}
```

Uses NOTREACHED to handle unknown color types, which is a not an assertion on release builds. As you can see below, the only further color type constraints are also inside DCHECKs, so an attacker that can supply a malicious (but valid) skbitmap can cause a CreateJavaBitmap to create an output JavaBitmap bitmap with a smaller backing store than the input SkBitmap bitmap. This will lead to a heap-buffer-overflow in the memcpy below.

https://source.chromium.org/chromium/chromium/src/+/master:ui/gfx/android/java_bitmap.cc;l=89

```
ScopedJavaLocalRef<jobject> ConvertToJavaBitmap(const SkBitmap* skbitmap,
OomBehavior reaction) {
DCHECK(skbitmap);
DCHECK(!skbitmap->isNull());
SkColorType color_type = skbitmap->colorType();
DCHECK((color_type == kRGB_565_SkColorType) ||
(color_type == kN32_SkColorType));
ScopedJavaLocalRef<jobject> jbitmap = CreateJavaBitmap(
skbitmap->width(), skbitmap->height(), color_type, reaction);
if (!jbitmap) {
DCHECK_EQ(OomBehavior::kReturnNullOnOom, reaction);
return jbitmap;
}
JavaBitmap dst_lock(jbitmap);
void* src_pixels = skbitmap->getPixels();
void* dst_pixels = dst_lock.pixels();
memcpy(dst_pixels, src_pixels, skbitmap->computeByteSize()); // <-- we use skbitmap size here, which may be larger than allocated size.

return jbitmap;
}
```

VERSION
Chrome Version: [stable]
Operating System: [Android]

REPRODUCTION CASE
Apply the attached patch that simulates a compromised renderer state on top of Chrome 86.0.4240.99.

Build for android, launch it, see crash in logcat output.

Recently, the drag-and-drop feature has been reimplemented using Mojo; therefore, the patch will require modifications to work on ToT. However, the vulnerability is still present in ToT.

Reporter	ma...@google.com
Type	Vulnerability
Priority	P1
Severity	S1
Status	Fixed
Story points	--
Access	Default access View
Expanded Access	
Assignee	sk...@chromium.org
Verifier	
Collaborators	
CC	ad...@chromium.org ad...@google.com av...@chromium.org aw...@google.com bo...@chromium.org bs...@chromium.org bs...@google.com cb...@chromium.org cc...@chromium.org cr...@chromium.org cr...@google.com fm...@chromium.org ha...@google.com hc...@chromium.org hu...@chromium.org kb...@chromium.org kh...@chromium.org ma...@google.com re...@google.com te...@chromium.org te...@google.com yf...@chromium.org yf...@google.com
Code Changes	--
Pending Code Changes	--
Backlog-Rank	--
BuildNumber	--
Chromium Labels	allpublic
Component Tags	UI>Browser
CVE	2020-16010
CWE ID	--
Design-Doc (Deprecated)	--
Design-TLDR-Summary (Deprecated)	--
EstimatedDays	--
Flaky-Test	--
Merge	Merged-4307 Merged-4280

FOR CRASHES, PLEASE INCLUDE THE FOLLOWING ADDITIONAL INFORMATION
Type of crash: [tab, browser, etc.]
Crash State: [see link above: stack trace *with symbols*, registers, exception record]
Client ID (if relevant): [see link above]

CREDIT INFORMATION
Externally reported security bugs may appear in Chrome release notes. If this bug is included, how would you like to be credited?
Reporter credit: [goes here]

renderer_patch.diff

983 B View Download

crash.log

17 KB View Download

COMMENTS

All comments | ↓ Oldest first

ad...@google.com <ad...@google.com> #2 Oct 31, 2020 12:45AM

From discussion offline with Mark, this affects M86. I have not personally reproduced it but I am setting the Security_Impact label. This is a sandbox escape from a compromised renderer, so is "high" severity (though at the upper end of that range).

ad...@google.com <ad...@google.com> #3 Oct 31, 2020 12:48AM

Assigned to sk...@chromium.org.

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #4 Oct 31, 2020 12:49AM

This only affects Android so deselecting ChromeOS.

[Monorail components: UI>Browser]

go...@google.com <go...@google.com> #5 Oct 31, 2020 12:54AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #6 Oct 31, 2020 12:59AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #7 Oct 31, 2020 01:25AM

Quick fix here, as agreed with Project Zero as being the best way forward *if* we want an emergency fix:

https://chromium-review.googlesource.com/c/chromium/src/+/2511859

(An alternative would have been to turn the format-check DCHECK into a CHECK, but it was felt best to eliminate any other possible ways that the wrong sizes could be passed to the memcpy in case there was some other flow we didn't spot).

I've e-mailed the relevant OWNERS etc. suggesting that we might want to land this (or something like it) over the weekend so we can get it into a Canary, and stand the possibility of including a fix here in the respin we're already making towards the beginning of next week. However, we also want to do a more thorough check of whether we could have made similar mistakes elsewhere.

ad...@google.com <ad...@google.com> #8 Oct 31, 2020 01:26AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #9 Oct 31, 2020 01:27AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #10 Oct 31, 2020 02:36PM

[Empty comment from Monorail migration]

	Merged-87
	... and 3 more (show all)
Merge-Request	--
Milestone	86 87
NextAction	--
Notice	--
OS	Android
ReleaseBlock	Stable
Respin	--
IRM Link	--
Security_Release	4-M86
vrp-reward	--
Fixed By Code Changes	--
Introduced In	--
Found In	--
Targeted To	--
Verified In	--
In Prod	

Show 1 additional field

ad...@google.com <ad...@google.com> #11 Oct 31, 2020 02:43PM

[Empty comment from Monorail migration]

hc...@google.com <hc...@google.com> #12 Oct 31, 2020 02:51PM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #13 Oct 31, 2020 03:00PM

[Empty comment from Monorail migration]

sk...@chromium.org <sk...@chromium.org> #14 Oct 31, 2020 03:39PM

I've +1'd the initial fix from #6 -- thanks for putting it together! I think it'd be worth also checking CreateSkBitmapFromJavaBitmap() and any other place where we convert between Java and Skia bitmap types for similar errors. These APIs are complex enough that it's not unlikely to end up with bugs like these :\

ad...@google.com <ad...@google.com> #15 Oct 31, 2020 03:56PM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #16 Oct 31, 2020 03:57PM

The situation is this:

I've worked with skyostiil@ and fmalita@ (thanks both!) and we've hit submit on the emergency fix which Project Zero proposed (with some minor changes). It's going through CQ now.

We have a faint concern that the code might have been _relying on_ buffer overflow in some rare cases, for bitmaps with padding. Therefore, even though the emergency fix looks trivial and highly targeted towards stopping this specific exploitation, there is a non-zero chance that this could cause stability problems, and we will want to get this into a Canary as soon as we can, prior to merging to M86 for release on Monday or Tuesday.

ad...@google.com <ad...@google.com> #17 Oct 31, 2020 04:11PM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #18 Oct 31, 2020 04:42PM

Pre-emptively approving merge to M86 and M87, but we're going to wait for a day of Canary coverage first. govind@ is kindly going to set a new Android Canary baking as soon as the CL lands in trunk.

I am expecting Sheriffbot to get cross with me because this is not yet marked as Fixed.

ad...@google.com <ad...@google.com> #19 Oct 31, 2020 04:43PM

[Empty comment from Monorail migration]

bu...@chops-service-accounts.iam.gserviceaccount.com <bu...@chops-service-accounts.iam.gserviceaccount.com> #20 Oct 31, 2020 05:21PM

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/e598fc599bd920392256d05c61826466c73c8e89

commit e598fc599bd920392256d05c61826466c73c8e89
Author: Adrian Taylor <adetaylor@chromium.org>
Date: Sat Oct 31 17:18:47 2020

Avoid bitmap overflow.

This ensures there are no circumstances under which the
following memcpy could write beyond the end of the bitmap.

Bug: 1144368
Change-Id: I2d41d9f059445c936387a25d9fe9b45818a3e649
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2511859
Commit-Queue: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Avi Drissman <avi@chromium.org>
Reviewed-by: Sami Kyöstilä <skyostil@chromium.org>
Cr-Commit-Position: refs/heads/master@{#822974}

[modify] https://crrev.com/e598fc599bd920392256d05c61826466c73c8e89/ui/gfx/android/java_bitmap.cc

[Deleted User] <[Deleted User]> #21 Oct 31, 2020 05:29PM

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

---

bu...@chops-service-accounts.iam.gserviceaccount.com <bu...@chops-service-accounts.iam.gserviceaccount.com> #22 Oct 31, 2020 05:34PM

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/710a6ff5f410ebfd47d9fb8919fd49309bf68480

commit 710a6ff5f410ebfd47d9fb8919fd49309bf68480
Author: Adrian Taylor <adetaylor@chromium.org>
Date: Sat Oct 31 17:29:49 2020

Avoid bitmap overflow.

This ensures there are no circumstances under which the
following memcpy could write beyond the end of the bitmap.

Bug: 1144368
Change-Id: I2d41d9f059445c936387a25d9fe9b45818a3e649
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2512796
Reviewed-by: Krishna Govind <govind@chromium.org>
Reviewed-by: Adrian Taylor <adetaylor@chromium.org>
Commit-Queue: Krishna Govind <govind@chromium.org>
Cr-Commit-Position: refs/branch-heads/4307@{#6}
Cr-Branched-From: 49634b02e0b1b5a048d58877f2c43d798b84b3b4-refs/heads/master@{#821981}

[modify] https://crrev.com/710a6ff5f410ebfd47d9fb8919fd49309bf68480/ui/gfx/android/java_bitmap.cc

---

go...@google.com <go...@google.com> #23 Oct 31, 2020 05:43PM

Canary #88.0.4307.4 (currently building) includes this fix. Please verify when available in ~10 hrs. Thank you.

---

ad...@google.com <ad...@google.com> #24 Oct 31, 2020 07:50PM

Marked as fixed.

I am marking this as Fixed so it can go through the normal merge/release notes/CVE processes. I've raised https://crbug.com/chromium/1144462 for follow-up work including checking other code paths and perhaps moving to the readPixels() API.

---

go...@google.com <go...@google.com> #25 Oct 31, 2020 07:52PM

CQ dry run passed for M86 & M87, will submit after canary coverage and verification.

M86: https://chromium-review.googlesource.com/c/chromium/src/+/2513107
M87: https://chromium-review.googlesource.com/c/chromium/src/+/2513108

---

ad...@google.com <ad...@google.com> #26 Oct 31, 2020 10:16PM

I won't be able to verify this on the Canary build because this is just one step in an exploit chain. To exploit this bug, we first need _another_ bug to execute malicious code in the compromised renderer. Short of deliberately introducing a new security vulnerability in Canary, there's no way to test the exploitability of this bug in isolation.

However, we want the Canary build to go out to ensure there aren't a spike in crashes from this new CHECK. It's very unlikely. If there are, we have a serious problem because we were relying on the previous exploitable buffer overflow.

So here's what I plan to do to verify this.

Testing that the fix itself works:
* Build a Chrome for Android build from branch 4240. Ensure it works.
* Apply the renderer_patch.diff from this bug to simulate a compromised renderer.
* Build and run, whilst monitoring logcat.
* We may or may not see a crash. If we do, it should be a segmentation fault.
* Then apply the fix from the above CL.
* Build and run, whilst monitoring logcat.
* We should definitely see a crash, and logcat should show it's from the newly added CHECK. If so, this is fixed, as the bug is no longer exploitable.
* And of course, confirm that the code for the canary build has an identical fix (i.e. nothing went wrong with merging).

Testing that the fix hasn't broken anything else:
* Use the Canary build.
* Generally browse around.
* Try to think of any circumstances where the renderer process will be sending images to the browser process, and test some of those cases. (skyostil@ I could use some help here!)

ad...@google.com <ad...@google.com> #27 Oct 31, 2020 10:24PM

OK. Crash received when using 4240 + renderer_patch.diff:

--------- beginning of crash
10-31 15:23:52.167 5072 5072 F libc : Fatal signal 11 (SIGSEGV), code 2, fault addr 0x99b89000 in tid 5072 (chromium.chrome)
10-31 15:23:52.272 200 200 F DEBUG : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
10-31 15:23:52.272 200 200 F DEBUG : Build fingerprint: 'google/hammerhead/hammerhead:6.0.1/M4B30Z/3437181:user/release-keys'
10-31 15:23:52.272 200 200 F DEBUG : Revision: '11'
10-31 15:23:52.272 200 200 F DEBUG : ABI: 'arm'
10-31 15:23:52.272 200 200 F DEBUG : pid: 5072, tid: 5072, name: chromium.chrome >>> org.chromium.chrome <<<
10-31 15:23:52.272 200 200 F DEBUG : signal 11 (SIGSEGV), code 2 (SEGV_ACCERR), fault addr 0x99b89000
10-31 15:23:52.287 200 200 F DEBUG : r0 99b88ff0 r1 998c3004 r2 0007cfbc r3 00000004
10-31 15:23:52.287 200 200 F DEBUG : r4 bef038b8 r5 9c76da58 r6 99840000 r7 99b0600c
10-31 15:23:52.287 200 200 F DEBUG : r8 9e37dcdc r9 aa6d7690 sl bef03c1c fp 00000000
10-31 15:23:52.287 200 200 F DEBUG : ip 80000000 sp bef038a8 lr 95b30fa5 pc b6c5e668 cpsr 20010030
10-31 15:23:52.308 200 200 F DEBUG :
10-31 15:23:52.308 200 200 F DEBUG : backtrace:
10-31 15:23:52.308 200 200 F DEBUG : #00 pc 00017668 /system/lib/libc.so (__memcpy_base+95)
10-31 15:23:52.308 200 200 F DEBUG : #01 pc 01d46fa1 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #02 pc 012eadcd /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #03 pc 012eae61 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #04 pc 01223fb7 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #05 pc 01228c71 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #06 pc 01228c41 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #07 pc 01223dc9 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #08 pc 01223be3 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #09 pc 02024869 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #10 pc 01c7d609 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.308 200 200 F DEBUG : #11 pc 01c87f0f /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.309 200 200 F DEBUG : #12 pc 01c87c6f /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.309 200 200 F DEBUG : #13 pc 01c881b1 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.309 200 200 F DEBUG : #14 pc 01cae46b /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.309 200 200 F DEBUG : #15 pc 01cae301 /data/app/org.chromium.chrome-2/base.apk (offset 0xdc6000)
10-31 15:23:52.309 200 200 F DEBUG : #16 pc 00012e93 /system/lib/libutils.so (_ZN7android6Looper9pollInnerEi+530)
10-31 15:23:52.309 200 200 F DEBUG : #17 pc 00012f63 /system/lib/libutils.so (_ZN7android6Looper8pollOnceEiPiS1_PPv+130)
10-31 15:23:52.309 200 200 F DEBUG : #18 pc 00081d05 /system/lib/libandroid_runtime.so (_ZN7android18NativeMessageQueue8pollOnceEP7_JNIEnvP8_jobjecti+22)
10-31 15:23:52.309 200 200 F DEBUG : #19 pc 7220556d /data/dalvik-cache/arm/system@framework@boot.oat (offset 0x1ed6000)
10-31 15:23:53.080 200 200 F DEBUG :

gl...@google.com <gl...@google.com> #28 Oct 31, 2020 10:27PM

Attached a reproducer for ToT. Interestingly, the new Mojo-based API limits the the supported color types to ones that use at most 32 bits per pixel (which is the default value used by |CreateJavaBitmap| to calculate the allocation size):

https://source.chromium.org/chromium/chromium/src/+/master:skia/public/mojom/image_info.mojom;drc=5b8933e94139a0ab5be46141666fdfcce0f624f6;l=10

Therefore, it's impossible to trigger the overflow with a non-standard color type. However, it's still vulnerable because |CreateJavaBitmap| doesn't take into account the |row_bytes| parameter, which similarly affects the allocation size:

https://source.chromium.org/chromium/chromium/src/+/master:skia/public/mojom/bitmap.mojom;drc=af69de90371db4b8dd74eb22410d556f553cdcb4;l=14

crash_trunk.log

16 KB View Download

renderer_patch_trunk.diff

1.2 KB View Download

ad...@google.com <ad...@google.com> #29 Oct 31, 2020 10:28PM

Crash received when using 4240 + renderer_patch.diff + commit 87ef01055ad05a8f506243d2a674045fb50d86ad (from https://chromium-review.googlesource.com/c/chromium/src/+/2513107):

```
10-31 15:26:31.189   200   200 F DEBUG   : Abort message: '[FATAL:java_bitmap.cc(91)] Check failed:
base::checked_cast<size_t>(dst_lock.byte_count()) >= skbitmap->computeByteSize() (524288 vs. 1048576)
10-31 15:26:31.189   200   200 F DEBUG   : Task trace:
10-31 15:26:31.189   200   200 F DEBUG   : #00 pc 0x02024872 /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.189   200   200 F DEBUG   : #01 pc 0x0202bc14 /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.189   200   200 F DEBUG   : IPC message handler context: 0x9EDDC710
10-31 15:26:31.189   200   200 F DEBUG   :
10-31 15:26:31.189   200   200 F DEBUG   : '
10-31 15:26:31.189   200   200 F DEBUG   :    r0 00000000  r1 0000159a  r2 00000006  r3 b6f17b7c
10-31 15:26:31.189   200   200 F DEBUG   :    r4 b6f17b84  r5 b6f17b34  r6 00000000  r7 0000010c
10-31 15:26:31.189   200   200 F DEBUG   :    r8 bef0335c  r9 b6cc0ec0  sl 9b77e448  fp 9b77e440
10-31 15:26:31.189   200   200 F DEBUG   :    ip 00000006  sp bef032b0  lr b6c86b61  pc b6c88f50  cpsr 400d0010
10-31 15:26:31.210   200   200 F DEBUG   :
10-31 15:26:31.210   200   200 F DEBUG   : backtrace:
10-31 15:26:31.210   200   200 F DEBUG   :    #00 pc 00041f50  /system/lib/libc.so (tgkill+12)
10-31 15:26:31.211   200   200 F DEBUG   :    #01 pc 0003fb5d  /system/lib/libc.so (pthread_kill+32)
10-31 15:26:31.211   200   200 F DEBUG   :    #02 pc 0001c30f  /system/lib/libc.so (raise+10)
10-31 15:26:31.211   200   200 F DEBUG   :    #03 pc 000194c1  /system/lib/libc.so (__libc_android_abort+34)
10-31 15:26:31.211   200   200 F DEBUG   :    #04 pc 000174ac  /system/lib/libc.so (abort+4)
10-31 15:26:31.211   200   200 F DEBUG   :    #05 pc 01ca4bb3  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.211   200   200 F DEBUG   :    #06 pc 01c54def  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.211   200   200 F DEBUG   :    #07 pc 01c550a3  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.211   200   200 F DEBUG   :    #08 pc 01c471f7  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #09 pc 01d46fe9  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #10 pc 012eadcd  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #11 pc 012eae61  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #12 pc 01223fb7  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #13 pc 01228c71  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #14 pc 01228c41  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #15 pc 01223dc9  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #16 pc 01223be3  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.212   200   200 F DEBUG   :    #17 pc 020248d9  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #18 pc 01c7d609  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #19 pc 01c87f0f  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #20 pc 01c87c6f  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #21 pc 01c881b1  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #22 pc 01cae46b  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #23 pc 01cae301  /data/app/org.chromium.chrome-1/base.apk (offset
0xdc6000)
10-31 15:26:31.213   200   200 F DEBUG   :    #24 pc 00012e93  /system/lib/libutils.so (_ZN7android6Looper9pollInnerEi+530)
10-31 15:26:31.213   200   200 F DEBUG   :    #25 pc 00012f63  /system/lib/libutils.so
(_ZN7android6Looper8pollOnceEiPiS1_PPv+130)
10-31 15:26:31.213   200   200 F DEBUG   :    #26 pc 00081d05  /system/lib/libandroid_runtime.so
(_ZN7android18NativeMessageQueue8pollOnceEP7_JNIEnvP8_jobjecti+22)
10-31 15:26:31.213   200   200 F DEBUG   :    #27 pc 7220556d  /data/dalvik-cache/arm/system@framework@boot.oat
(offset 0x1ed6000)
10-31 15:26:31.890   778  5671 W ActivityManager:   Force finishing activity
org.chromium.chrome/com.google.android.apps.chrome.Main
```

This is exactly the result I expected to see.

I consider verification of the actual fix is complete, albeit on branch 4240 + https://chromium-review.googlesource.com/c/chromium/src/+/2513107, rather than on the Canary build.

ad...@google.com <ad...@google.com> #30 Oct 31, 2020 10:43PM

As to more general testing of Canary:
I can't figure out what of https://pantheon.corp.google.com/storage/browser/chrome-signed/android-

B0urB0N/88.0.4307.4/arm?pageState=(%22StorageObjectListTable%22:(%22f%22:%22%255B%255D%22))&forceOnObjectsSortingFiltering=false I need to install, so I think I'll need to wait for it to get to the play store.

However, I have confirmed that refs/tags/88.0.4307.4 contains the fix as expected.

aw...@google.com <aw...@google.com> #31 Oct 31, 2020 10:44PM

[Empty comment from Monorail migration]

gl...@google.com <gl...@google.com> #32 Nov 1, 2020 03:19AM

While we've only seen this vulnerability used against Android devices, an identical bug exists in Chrome on Windows.

https://source.chromium.org/chromium/chromium/src/+/master:ui/base/dragdrop/os_exchange_data_provider_win.cc;drc=2928283f0a7d52e68f326e3befc5568edccc9570;l=724

```
void OSExchangeDataProviderWin::SetDragImage(
    const gfx::ImageSkia& image_skia,
    const gfx::Vector2d& cursor_offset) {
  DCHECK(!image_skia.size().IsEmpty());

  // InitializeFromBitmap() doesn't expect an alpha channel and is confused
  // by premultiplied colors, so unpremultiply the bitmap.
  SkBitmap unpremul_bitmap =
      SkBitmapOperations::UnPreMultiply(*image_skia.bitmap());
  int width = unpremul_bitmap.width();
  int height = unpremul_bitmap.height();
  size_t rowbytes = unpremul_bitmap.rowBytes();
  DCHECK_EQ(rowbytes, static_cast<size_t>(width) * 4u);

  void* bits;
  HBITMAP hbitmap;
  {
    BITMAPINFOHEADER header;
    skia::CreateBitmapHeader(width, height, &header);

    base::win::ScopedGetDC screen_dc(NULL);
    // By giving a null hSection, the |bits| will be destroyed when the
    // |hbitmap| is destroyed.
    hbitmap =
        CreateDIBSection(screen_dc, reinterpret_cast<BITMAPINFO*>(&header),
                         DIB_RGB_COLORS, &bits, NULL, 0);
  }
  if (!hbitmap)
    return;

  memcpy(bits, unpremul_bitmap.getPixels(), height * rowbytes);
```

The only modification needed to make the original reproducer trigger the bug on Windows is to set the alpha type to |kUnpremul_SkAlphaType|.

renderer_patch_windows.diff

981 B View Download

asan_win.log

4.9 KB View Download

ad...@google.com <ad...@google.com> #33 Nov 1, 2020 03:23AM

Well, that's disappointing.

I'd like to keep the current crbug for the Android vulnerability. I'm going to raise a new crbug for the equivalent Windows vulnerability right now.

ad...@google.com <ad...@google.com> #34 Nov 1, 2020 03:26AM

Windows bug raised as https://crbug.com/chromium/1144489.

ad...@google.com <ad...@google.com> #35 Nov 1, 2020 03:27AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #36 Nov 1, 2020 04:41AM

[Empty comment from Monorail migration]

go...@google.com <go...@google.com> #37 Nov 1, 2020 05:30AM

Android Canary version #88.0.4307.5 includes this fix.

[Deleted User] <[Deleted User]> #38 Nov 1, 2020 06:56PM

[Empty comment from Monorail migration]

go...@google.com <go...@google.com> #39 Nov 1, 2020 07:01PM

This change also made it to latest canary #88.0.4312.0 (currently building) .

bu...@chops-service-accounts.iam.gserviceaccount.com <bu...@chops-service-accounts.iam.gserviceaccount.com> #40
Nov 2, 2020 01:03AM

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/65362c9cc60a5febeb3c013a9a23ea5aad41fd37

commit 65362c9cc60a5febeb3c013a9a23ea5aad41fd37
Author: Adrian Taylor <adetaylor@chromium.org>
Date: Mon Nov 02 01:00:38 2020

Avoid bitmap overflow.

This ensures there are no circumstances under which the
following memcpy could write beyond the end of the bitmap.

(cherry picked from commit e598fc599bd920392256d05c61826466c73c8e89)

Bug: 1144368
Change-Id: I2d41d9f059445c936387a25d9fe9b45818a3e649
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2511859
Commit-Queue: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Avi Drissman <avi@chromium.org>
Reviewed-by: Sami Kyöstilä <skyostil@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#822974}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2513108
Reviewed-by: Krishna Govind <govind@chromium.org>
Reviewed-by: Adrian Taylor <adetaylor@chromium.org>
Commit-Queue: Krishna Govind <govind@chromium.org>
Cr-Commit-Position: refs/branch-heads/4280@{#999}
Cr-Branched-From: ea420fb963f9658c9969b6513c56b8f47efa1a2a-refs/heads/master@{#812852}

[modify] https://crrev.com/65362c9cc60a5febeb3c013a9a23ea5aad41fd37/ui/gfx/android/java_bitmap.cc

bu...@chops-service-accounts.iam.gserviceaccount.com <bu...@chops-service-accounts.iam.gserviceaccount.com> #41
Nov 2, 2020 01:08AM

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/c8b05f21a27fcc0287256d991d562d219551eb2c

commit c8b05f21a27fcc0287256d991d562d219551eb2c
Author: Adrian Taylor <adetaylor@chromium.org>
Date: Mon Nov 02 01:07:29 2020

Avoid bitmap overflow.

This ensures there are no circumstances under which the
following memcpy could write beyond the end of the bitmap.

(cherry picked from commit e598fc599bd920392256d05c61826466c73c8e89)

Bug: 1144368
Change-Id: I2d41d9f059445c936387a25d9fe9b45818a3e649
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2511859
Commit-Queue: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Avi Drissman <avi@chromium.org>
Reviewed-by: Sami Kyöstilä <skyostil@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#822974}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2513107
Reviewed-by: Krishna Govind <govind@chromium.org>
Reviewed-by: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Ben Mason <benmason@chromium.org>
Commit-Queue: Krishna Govind <govind@chromium.org>
Cr-Commit-Position: refs/branch-heads/4240@{#1375}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/c8b05f21a27fcc0287256d991d562d219551eb2c/ui/gfx/android/java_bitmap.cc

ad...@google.com <ad...@google.com> #42 Nov 2, 2020 02:06AM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #43 Nov 2, 2020 08:32PM

[Empty comment from Monorail migration]

kb...@chromium.org <kb...@chromium.org> #44 Nov 2, 2020 11:52PM

[Empty comment from Monorail migration]

ad...@google.com <ad...@google.com> #45 Nov 3, 2020 02:16AM

[Empty comment from Monorail migration]

bu...@chops-service-accounts.iam.gserviceaccount.com <bu...@chops-service-accounts.iam.gserviceaccount.com> #46 Nov 5, 2020 08:51AM

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/66b30ce28cd8db967750530563f164a7380501bc

commit 66b30ce28cd8db967750530563f164a7380501bc
Author: Adrian Taylor <adetaylor@chromium.org>
Date: Thu Nov 05 08:50:18 2020

Avoid bitmap overflow.

This ensures there are no circumstances under which the
following memcpy could write beyond the end of the bitmap.

(cherry picked from commit e598fc599bd920392256d05c61826466c73c8e89)

(cherry picked from commit c8b05f21a27fcc0287256d991d562d219551eb2c)

Bug: 1144368
Change-Id: I2d41d9f059445c936387a25d9fe9b45818a3e649
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2511859
Commit-Queue: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Avi Drissman <avi@chromium.org>
Reviewed-by: Sami Kyöstilä <skyostil@chromium.org>
Cr-Original-Original-Commit-Position: refs/heads/master@{#822974}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2513107
Reviewed-by: Krishna Govind <govind@chromium.org>
Reviewed-by: Adrian Taylor <adetaylor@chromium.org>
Reviewed-by: Ben Mason <benmason@chromium.org>
Commit-Queue: Krishna Govind <govind@chromium.org>
Cr-Original-Commit-Position: refs/branch-heads/4240@{#1375}
Cr-Original-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2518054
Reviewed-by: Achuith Bhandarkar <achuith@chromium.org>
Commit-Queue: Victor-Gabriel Savu <vsavu@google.com>
Cr-Commit-Position: refs/branch-heads/4240_112@{#17}
Cr-Branched-From: 427c00d3874b6abcf4c4c2719768835fc3ef26d6-refs/branch-heads/4240@{#1291}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/66b30ce28cd8db967750530563f164a7380501bc/ui/gfx/android/java_bitmap.cc

kb...@chromium.org <kb...@chromium.org> #47 Nov 5, 2020 08:06PM

[Empty comment from Monorail migration]

hu...@chromium.org <hu...@chromium.org> #48 Nov 5, 2020 08:50PM

[Empty comment from Monorail migration]

[Deleted User] <[Deleted User]> #49 Feb 7, 2021 06:52PM

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

ha...@google.com <ha...@google.com> #50 Jan 9, 2024 02:30AM

[Empty comment from Monorail migration]

ha...@google.com <ha...@google.com> #51 Jan 9, 2024 02:39AM

[Empty comment from Monorail migration]

is...@google.com <is...@google.com> #52 Jan 9, 2024 02:39AM

This issue was migrated from crbug.com/chromium/1144368?no_tracker_redirect=1

[Monorail components added to Component Tags custom field.]