Starred by 2 users

Owner:	michaelludwig@google.com
CC:	michaelludwig@google.com backer@chromium.org pbomm...@chromium.org kylec...@chromium.org bsalo...@google.com
Status:	Fixed *(Closed)*
Components:	Internals>Compositing
Modified:	Aug 23, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Windows
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
Security_Impact-Stable
Security_Severity-Medium
allpublic
CVE_description-submitted
Merge-Rejected-91
M-92
Target-91
Target-92
merge-merged-m90
LTS-Merged-90
LTS-Security-90
Release-0-M92
CVE-2021-30587
LTS-Size-Small
LTS-Complexity-Trivial

Issue 1204347: Security: 3d css can still glitch onto native browser UI

Reported by abalq...@microsoft.com on Thu, Apr 29, 2021, 10:24 PM EDT

Code

VULNERABILITY DETAILS
Still seeing ~~bug 1162042~~ and ~~bug 1177833~~ reproduce on latest canary

VERSION
Chrome Version: 92.0.4491.6 (Official Build) canary (64-bit)
Operating System: (using windows 10)

REPRODUCTION CASE
1. go to https://threejs.org/examples/?q=css#css3d_molecules
2. click on 'aluminum oxide' at the bottom
3. click on 'bonds' at the top to show only white bonds
4. scroll up to zoom in maximum
5. Drag the molecule and quickly move in different directions

As you may see in the screenshot, the entire browser UI turned white. Other times it covers parts of it.

Related: ~~Bug 1177833~~, ~~Bug 1162042~~

CREDIT INFORMATION
Reporter credit: Abdulrahman Alqabandi, Microsoft Browser Vulnerability Research

Screenshot 2021-04-30 045659.png
72.6 KB View Download

Comment 1 by ajgo@google.com on Thu, Apr 29, 2021, 11:52 PM EDT Project Member

Cc: michaelludwig@google.com bsalo...@google.com backer@chromium.org kylec...@chromium.org
Components: Internals>Compositing

Adding some folks from previous bugs.

Comment 2 by ajgo@google.com on Fri, Apr 30, 2021, 12:04 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Owner: bsalo...@google.com
Labels: Security_Severity-Medium Security_Impact-Stable OS-Windows Pri-2

I can repro on Windows Canary - initially the browser UI flickers but later is filled in. Not sure how controllable this is. Nothing draws outside the browser Window.

Assigning to bsaloman - feel free to assign to someone else if they are better placed to investigate this security bug.

Comment 3 by bsalo...@google.com on Fri, Apr 30, 2021, 12:39 PM EDT Project Member

Owner: michaelludwig@google.com

I think Michael is the right owner for this.

Comment 4 by sheriffbot on Fri, Apr 30, 2021, 1:02 PM EDT Project Member

Labels: M-91 Target-91

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

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

Comment 5 by sheriffbot on Fri, Apr 30, 2021, 1:38 PM EDT Project Member

Labels: -Pri-2 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 6 by michaelludwig@google.com on Fri, May 7, 2021, 4:37 PM EDT Project Member

Owner: kylec...@chromium.org

I've investigated a fair amount into what's going and I tentatively think the issues are higher up in chromium, and distinct from the earlier issues that Brian fixed.

In order to test, I forced the edge AA flags to be kNone for every DrawQuad in skia_renderer[1]. This ensures that once passed to Skia using the experimental_DrawEdgeAA APIs, we don't do anything to the vertices other than a matrix multiply. Importantly, it avoids any insetting/outsetting calculations and relies on the GPU to clip against W > 0 instead of doing it manually. Those were the areas that were numerically unstable, and plausibly could still have issues under specific scenarios.

But when I did this, I was still able to reproduce the overdraw into the UI area. If I further hacked the compositor to draw perspective quads under a different color, the overdrawn areas were not corrupted with the same color. Instead it seemed to be the default gray background color. I would also see occasional flickers of that same color replacing the black background of the page and some of the 3d divs would display. I don't know where this would have been coming from; maybe it was another div grouped into the sorting context and occasionally bleeding through, or maybe a new tile's content never being drawn to? Given the sheer number of draw quads being generated by these scenes, it's difficult to track down the bad ones.

My current hypothesis is that the quad BSP splitter is generating bad results, or something higher up in cc is make bad decisions about the layer sizes. I am more inclined to suspect the BSP. If I remember correctly, the BSP splitter's output polygons are converted to quadrilaterals and it's assumed that the new vertices are all inside the original rectangle. With extreme values from perspective divisions (as well as the particularly complex set of intersecting divs), numerical error could easily create points that violate this assumption. skia_renderer currently makes its scissor decisions (whether or not to call SkCanvas::clipRect) based on the visible rect, so if that is correct but the BSP split points are not contained within it, we'd run into issues like this.

I also noticed that it was much more prevalent on Windows than it was on Linux, but I'm not sure why. Unfortunately, my WFH setup only lets me really build chrome on Linux. Cross-compiling to windows and downloading the binary from my corp workstation has allowed me to test, but it takes 5-10min to download over my home internet connection, which you have to do after every compile :(

I'm passing over to Kyle for investigation into the BSP issues, partially because I believe he develops on Windows(?) so may not run into these cycle delays. I am definitely happy to advise and discuss. Brian and I have also made plans for a more robust way to draw quadrilaterals with per-edge AA that will hopefully put Skia's quadrilateral rendering in a better place (but that would not necessarily solve this bug unless I was wrong about my hypothesis). I've attached a single html file that can be used to reproduce and modify the transform a little more easily.

css3d_molecules.html
11.4 KB View Download

Comment 7 by kylec...@chromium.org on Mon, May 10, 2021, 4:32 PM EDT Project Member

I don't have a Windows workstation anymore unfortunately. I am able to reproduce on Windows pretty consistently and extremely infrequently on Linux.

I've also been unable to reproduce on Windows if I switch back to GLRenderer.

I'll see if I can get a consistent repro with a static page on Linux if I can figure out how to dump the page state at the right time.

Comment 8 by michaelludwig@google.com on Mon, May 10, 2021, 4:42 PM EDT Project Member

Hmm, if it doesn't reproduce on Windows with GLRenderer, then it's less likely to be BSP related. Unless, it does reproduce errors within the page content but does something differently that prevents corrupting the UI (e.g. always has a scissor that's not conditionally turned off, or handles partial swap and damage tracking differently?) If none of those options seem promising, it would circle back to Skia as a likely cause, but it's peculiar that Windows reproduces so much more readily than linux. The main change within Skia between those two platforms would be how our auto-SIMD libraries are compiled and potentially what intrinsics are used.

Comment 9 by kylec...@chromium.org on Tue, May 11, 2021, 3:46 PM EDT Project Member

I have a minimized repro for Linux attached. It should reproduce 100% of the time if you have a maximized 1920x1080 window with Linux defaults (eg. software rasterization).

The BSP tree generates the same quads for both GLRenderer and SkiaRenderer. The split quads make it to SkiaRenderer::DrawColoredQuad() where DrawQuadParams::scissor_rect=0,72 1920x1008 which is the right rect for the tab content area. So I think the right scissor_rect makes it to SkiaRenderer::PrepareCanvas() but that's not actually working?

[1]
https://source.chromium.org/chromium/chromium/src/+/master:components/viz/service/display/skia_renderer.cc;l=1683;drc=0e20d1eb38227949805a4c0e9d5cdeddc8d23637

molecule.html
2.0 KB View Download

Comment 10 by michaelludwig@google.com on Tue, May 11, 2021, 4:47 PM EDT Project Member

Unfortunately, I'm unable to reproduce the exact failure when using gl on swiftshader and CRD to access a linux machine. I get the attached screenshot instead, although this does still show a failure, it's at least not corrupting the UI.

Given you have a reliable repro on your machine, what happens if you
1. Comment out the attemptQuadOptimization() line (just set opt to kCropped) at [1]
2. Set aa = GrAA::kNo and quad->fEdgeFlags = GrQuadAAFlags::kNone right after [1]

This should, at basically the lowest level that's convenient, turn off any aa outsetting and insetting skia would do. If that's the cause of the original issue, you shouldn't see it repro after making the above changes.

If that's the case, then we can undo the above changes and add logging at [2], [3], and [4] to see what the final geometry looks like.

[1]
https://source.chromium.org/chromium/chromium/src/+/main:third_party/skia/src/gpu/GrSurfaceDrawContext.cpp;drc=537293bf155f5688a8d91d609b5231799e77d81b;l=631
[2]
https://source.chromium.org/chromium/chromium/src/+/main:third_party/skia/src/gpu/ops/GrQuadPerEdgeAA.cpp;drc=e713e4b5b963cbaaf643c070273860317b58051b;l=339
[3]
https://source.chromium.org/chromium/chromium/src/+/main:third_party/skia/src/gpu/ops/GrQuadPerEdgeAA.cpp;drc=e713e4b5b963cbaaf643c070273860317b58051b;l=370

[4]
https://source.chromium.org/chromium/chromium/src/+/main:third_party/skia/src/gpu/ops/GrQuadPerEdgeAA.cpp;drc=e713e4b5b963cbaaf643c0702738603l7b58051b;l=388

Comment 11 by michaelludwig@google.com on Tue, May 11, 2021, 4:48 PM EDT Project Member

Here's the different type of rendering glitch I'm seeing.

different-repro.png
123 KB View Download

Comment 12 by kylec...@chromium.org on Wed, May 12, 2021, 10:11 AM EDT Project Member

I tried making the requested changes and while it does change the glitch it still draws over top of the UI. I've attached screenshots of the glitch at ToT before making changes vs the glitch after making those changes. I'm not sure how to interpret this.

before_changes.png
23.6 KB View Download

with_changes.png
15.1 KB View Download

Comment 13 by kylec...@chromium.org on Wed, May 12, 2021, 12:15 PM EDT Project Member

Here is what it looks like with GLRenderer.

gl_renderer.png
23.3 KB View Download

Comment 14 by kylec...@chromium.org on Wed, May 12, 2021, 1:38 PM EDT Project Member

Owner: michaelludwig@google.com

We have narrowed down the cause and michaelludwig@ is working on a fix.

Comment 15 by Git Watcher on Thu, May 13, 2021, 3:38 PM EDT Project Member

The following revision refers to this bug:
https://skia.googlesource.com/skia/+/65299907634abe53e697cef25df72f2c11cdabd8

commit 65299907634abe53e697cef25df72f2c11cdabd8
Author: Michael Ludwig <michaelludwig@google.com>
Date: Thu May 13 16:02:23 2021

Fix overdraw from unstable perspective math

There were two issues leading to the corruption seen in the linked
chromium issue.

1. The draw's bounds were calculated based on the quad being clipped
to w>= epsilon, which is what happens when the AA inset/outset is done.
But for non-aa quads, the fillrect and texture ops did no clipping,
assuming that the GPU would be sufficient. However, this can produce
non-aa draws that exceed the calculated bounds, misleading the clip
stack into incorrectly removing the scissor, etc.
2. Precision issues within CropToRect meant some perspective quads'
barycentric coordinates would become degenerate and compute to (0,0,1),
making it appear as if the render target/scissor were contained within
it. This meant we'd turn it into a rectangular clear.

These changes appear to address the corruption on Linux and Windows, but
there are still rendering artifacts from poor aa inset/outset
calculations. These artifacts are at least limited to the clip properly.
A better rendering method that does not rely on line intersections
will address these artifacts, but this CL is a reasonable temporary
mitigation.

~~Bug: chromium:1204347~~
Change-Id: I3c67d4efe70313ae7c98abc0a57b5b047c83890d
Reviewed-on: https://skia-review.googlesource.com/c/skia/+/407821
Reviewed-by: Brian Salomon <bsalomon@google.com>
Commit-Queue: Michael Ludwig <michaelludwig@google.com>

[modify] https://crrev.com/65299907634abe53e697cef25df72f2c11cdabd8/src/gpu/geometry/GrQuadUtils.cpp
[modify] https://crrev.com/65299907634abe53e697cef25df72f2c11cdabd8/src/gpu/ops/GrFillRectOp.cpp

[modify] https://crrev.com/65299907634abe53e697cef25df72f2c11cdabd8/src/gpu/ops/GrTextureOp.cpp
[modify] https://crrev.com/65299907634abe53e697cef25df72f2c11cdabd8/tests/GrQuadCropTest.cpp

Comment 16 by Git Watcher on Thu, May 13, 2021, 9:31 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/34cd1b712caa92e92c034b420faa0c9936f4cee8

commit 34cd1b712caa92e92c034b420faa0c9936f4cee8
Author: chromium-autoroll <chromium-autoroll@skia-public.iam.gserviceaccount.com>
Date: Fri May 14 01:30:07 2021

Roll Skia from d977fdecc634 to cad48c6868bf (22 revisions)

https://skia.googlesource.com/skia.git/+log/d977fdecc634..cad48c6868bf

2021-05-13 robertphillips@google.com Revert "Reland "Use conics with w=Inf to describe triangles for the tessellator""
2021-05-13 robertphillips@google.com Revert "Reland "Delete the index buffer from middle-out tessellation""
2021-05-13 brianosman@google.com Handle null vertex buffers in Metal
2021-05-13 skia-autoroll@skia-public.iam.gserviceaccount.com Roll ANGLE from 9809122dbd0d to c5e344b1e676 (9 revisions)
2021-05-13 brianosman@google.com Fix const globals in Metal
2021-05-13 csmartdalton@google.com Reland "Delete the index buffer from middle-out tessellation"
2021-05-13 herb@google.com Use shared lock for charsToGlyphs
2021-05-13 fmalita@chromium.org [skottie] Text fill/stroke opacity animators
2021-05-13 egdaniel@google.com Share DMSAA attachments in GL between render targets.
2021-05-13 herb@google.com reject sizes that will overflow in SkSpan
2021-05-13 michaelludwig@google.com Fix overdraw from unstable perspective math
2021-05-13 jmbetancourt@google.com [androidkit] add getter for matrices
2021-05-13 johnstiles@google.com Add support for matrix division to Metal codegen.
2021-05-13 johnstiles@google.com Cleanup operator conversion from assignment to non-assignment form.
2021-05-13 csmartdalton@google.com Add a GrCullTest class
2021-05-13 johnstiles@google.com Avoid repeated failure messages in the dm logs.
2021-05-13 johnstiles@google.com Reland "The Matrices test now verifies its results."
2021-05-13 johnstiles@google.com Add support for matrix + scalar to Metal codegen.
2021-05-13 jmbetancourt@google.com [androidkit] implement ThreadedSurface backed by WindowSurface
2021-05-13 csmartdalton@google.com Reland "Use conics with w=Inf to describe triangles for the tessellator"
2021-05-13 skia-autoroll@skia-public.iam.gserviceaccount.com Roll ANGLE from 25b53ceb65b7 to 9809122dbd0d (1 revision)
2021-05-13 brianosman@google.com Fix implicit signedness change warnings in private includes

If this roll has caused a breakage, revert this CL and stop the roller
using the controls here:
https://autoroll.skia.org/r/skia-autoroll
Please CC robertphillips@google.com on the revert to ensure that a human
is aware of the problem.

To report a problem with the AutoRoller itself, please file a bug:
https://bugs.chromium.org/p/skia/issues/entry?template=Autoroller+Bug

Documentation for the AutoRoller is here:
https://skia.googlesource.com/buildbot/+doc/master/autoroll/README.md

Cq-Include-Trybots: luci.chromium.try:android_optional_gpu_tests_rel;luci.chromium.try:linux-blink-rel;luci.chromium.try:linux-chromeos-compile-dbg;luci.chromium.try:linux_optional_gpu_tests_rel;luci.chromium.try:mac_optional_gpu_tests_rel;luci.chromium.try:win_optional_gpu_tests_rel
Cq-Do-Not-Cancel-Tryjobs: true
Bug: chromium:1202607,~~chromium:1204347~~
Tbr: robertphillips@google.com
Change-Id: Ic86c08835d47eb0e8dfebb18ec3b76b641019a71
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2895286
Commit-Queue: chromium-autoroll <chromium-autoroll@skia-public.iam.gserviceaccount.com>
Bot-Commit: chromium-autoroll <chromium-autoroll@skia-public.iam.gserviceaccount.com>
Cr-Commit-Position: refs/heads/master@{#882800}

[modify] https://crrev.com/34cd1b712caa92e92c034b420faa0c9936f4cee8/DEPS

Comment 17 by michaelludwig@google.com on Fri, May 14, 2021, 12:02 PM EDT Project Member

Labels: Merge-Request-91

The overdraw issues should be fixed. Given the proximity to the m91 stable and m92 beta, and that the overdraw is limited to these complex 3d css scenes, I'm not sure if it meets the merge requirements. Requesting to get more eyes on it.

Comment 18 by sheriffbot on Fri, May 14, 2021, 12:06 PM EDT Project Member

Labels: -Merge-Request-91 Hotlist-Merge-Review Merge-Review-91

This bug requires manual review: We are only 10 days from stable.
Before a merge request will be considered, the following information is required to be added to this bug:

1. Does your merge fit within the Merge Decision Guidelines?
- Chrome: https://chromium.googlesource.com/chromium/src.git/+/master/docs/process/merge_request.md#when-to-request-a-merge
- Chrome OS: https://goto.google.com/cros-release-branch-merge-guidelines
2. Links to the CLs you are requesting to merge.
3. Has the change landed and been verified on ToT?
4. Does this change need to be merged into other active release branches (M-1, M+1)?
5. Why are these changes required in this milestone after branch?
6. Is this a new feature?
7. If it is a new feature, is it behind a flag using finch?

Chrome OS Only:
8. Was the change reviewed and approved by the Eng Prod Representative? See Eng Prod ownership by component: http://go/cros-engprodcomponents

Please contact the milestone owner if you have questions.
Owners: benmason@(Android), bindusuvarna@(iOS), marinakz@(ChromeOS), pbommana@(Desktop)

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

Comment 19 by michaelludwig@google.com on Fri, May 14, 2021, 4:57 PM EDT Project Member

Cc: pbomm...@chromium.org

1. I think so, but arguments could be made to just wait to m92 given the relatively rare occurrence, and low risk when it does happen
2. https://skia-review.googlesource.com/c/skia/+/407821
3. Yes and yes
4. No
5. Draws over the UI, security impact medium
6. No, just a bug fix; disables optimizations that were invalid so is a safe change.
7. N/A

8. N/A

CCing @pbommana for severity guidance based on proximity to stable release, and the most commonly affected platform appears to be windows (linux did not trigger it as frequently).

Comment 20 by kenrb@chromium.org on Fri, May 14, 2021, 5:40 PM EDT Project Member

Status: Fixed (was: Assigned)

Marking as fixed based on comment 17.

Comment 21 by adetaylor@chromium.org on Fri, May 14, 2021, 6:42 PM EDT Project Member

Labels: -Merge-Review-91 Merge-Rejected-91

As a medium severity bug with a non-trivial fix, I'm inclined to leave this to M92. Thanks for all the information.

Comment 22 by sheriffbot on Sat, May 15, 2021, 2:00 PM EDT Project Member

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

Comment 23 by amyressler@chromium.org on Mon, Jul 19, 2021, 3:24 PM EDT Project Member

Labels: Release-0-M92

Comment 24 by amyressler@google.com on Mon, Jul 19, 2021, 7:18 PM EDT Project Member

Labels: CVE-2021-30587 CVE_description-missing

Comment 25 by amyressler@google.com on Tue, Aug 3, 2021, 3:42 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 26 by rzanoni@google.com on Tue, Aug 17, 2021, 3:39 AM EDT Project Member

Labels: LTS-Security-90 LTS-Merge-Request-90

Comment 27 by sheriffbot on Tue, Aug 17, 2021, 12:21 PM EDT Project Member

Labels: -M-91 Target-92 M-92

Comment 28 by rzanoni@google.com on Thu, Aug 19, 2021, 11:25 AM EDT Project Member

Labels: LTS-Size-Small LTS-Complexity-Trivial

Comment 29 by gianluca@google.com on Fri, Aug 20, 2021, 3:35 AM EDT Project Member

Labels: -LTS-Merge-Request-90 LTS-Merge-Approved-90

Comment 30 by Git Watcher on Fri, Aug 20, 2021, 3:18 PM EDT Project Member

Labels: merge-merged-m90

The following revision refers to this bug:
https://skia.googlesource.com/skia/+/c5b74472e754407e9d46f634be4a34b1ee55e7fc

commit c5b74472e754407e9d46f634be4a34b1ee55e7fc
Author: Michael Ludwig <michaelludwig@google.com>
Date: Thu May 13 16:02:23 2021

[M90-LTS] Fix overdraw from unstable perspective math

There were two issues leading to the corruption seen in the linked
chromium issue.

1. The draw's bounds were calculated based on the quad being clipped
to w>= epsilon, which is what happens when the AA inset/outset is done.
But for non-aa quads, the fillrect and texture ops did no clipping,
assuming that the GPU would be sufficient. However, this can produce
non-aa draws that exceed the calculated bounds, misleading the clip
stack into incorrectly removing the scissor, etc.
2. Precision issues within CropToRect meant some perspective quads'
barycentric coordinates would become degenerate and compute to (0,0,1),
making it appear as if the render target/scissor were contained within
it. This meant we'd turn it into a rectangular clear.

These changes appear to address the corruption on Linux and Windows, but
there are still rendering artifacts from poor aa inset/outset
calculations. These artifacts are at least limited to the clip properly.
A better rendering method that does not rely on line intersections
will address these artifacts, but this CL is a reasonable temporary
mitigation.

~~Bug: chromium:1204347~~
Change-Id: I3c67d4efe70313ae7c98abc0a57b5b047c83890d
Reviewed-on: https://skia-review.googlesource.com/c/skia/+/407821
Commit-Queue: Michael Ludwig <michaelludwig@google.com>
(cherry picked from commit 65299907634abe53e697cef25df72f2c11cdabd8)
Reviewed-on: https://skia-review.googlesource.com/c/skia/+/435636
Reviewed-by: Artem Sumaneev <asumaneev@google.com>
Reviewed-by: Brian Salomon <bsalomon@google.com>

[modify] https://crrev.com/c5b74472e754407e9d46f634be4a34b1ee55e7fc/src/gpu/geometry/GrQuadUtils.cpp
[modify] https://crrev.com/c5b74472e754407e9d46f634be4a34b1ee55e7fc/src/gpu/ops/GrFillRectOp.cpp
[modify] https://crrev.com/c5b74472e754407e9d46f634be4a34b1ee55e7fc/src/gpu/ops/GrTextureOp.cpp
[modify] https://crrev.com/c5b74472e754407e9d46f634be4a34b1ee55e7fc/tests/GrQuadCropTest.cpp

Comment 31 by sheriffbot on Sat, Aug 21, 2021, 1:28 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 32 by rzanoni@google.com on Mon, Aug 23, 2021, 3:58 AM EDT Project Member

Labels: -LTS-Merge-Approved-90 LTS-Merged-90