☆ Starred by 4 users

Owner:	acolwell@chromium.org **Last visit > 30 days ago**
CC:	ajha@chromium.org adetaylor@chromium.org wfh@chromium.org creis@chromium.org awhalley@google.com lukasza@chromium.org mek@chromium.org alex...@chromium.org achuith@chromium.org nasko@chromium.org ajgo@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>Storage Internals>Sandbox>SiteIsolation Blink>Storage>FileAPI
Modified:	Apr 30, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac
Pri:	1
Type:	Bug-Security

Security_Impact-Stable
Security_Severity-High
allpublic
CVE_description-submitted
Target-79
M-79
VulnerabilityAnalysis-Requested
merge-merged-3987
merge-merged-80
Release-0-M80
CVE-2020-6385

Issue 1035399: Security: Site Isolation bypass in BlobURLStoreImpl::Register

Reported by glazunov@google.com on Wed, Dec 18, 2019, 7:36 AM EST Project Member

Code

VULNERABILITY DETAILS
storage/browser/blob/blob_url_store_impl.cc:66:

```
BlobURLStoreImpl::~BlobURLStoreImpl() {
  if (context_) {
    for (const auto& url : urls_)
      context_->RevokePublicBlobURL(url); // ***1***
  }
}

void BlobURLStoreImpl::Register(mojo::PendingRemote<blink::mojom::Blob> blob,
                                const GURL& url,
                                RegisterCallback callback) {
  if (!url.SchemeIsBlob()) {
    mojo::ReportBadMessage("Invalid scheme passed to BlobURLStore::Register");
    std::move(callback).Run();
    return;
  }
  // Only report errors when we don't have permission to commit and
  // the process is valid. The process check is a temporary solution to
  // handle cases where this method is run after the
  // process associated with |delegate_| has been destroyed.
  // See https://crbug.com/933089 for details.
  if (!delegate_->CanCommitURL(url) && delegate_->IsProcessValid()) { // ***2***
    mojo::ReportBadMessage(
        "Non committable URL passed to BlobURLStore::Register");
    std::move(callback).Run();
    return;
  }
  if (BlobUrlUtils::UrlHasFragment(url)) {
    mojo::ReportBadMessage(
        "URL with fragment passed to BlobURLStore::Register");
    std::move(callback).Run();
    return;
  }

  if (context_)
    context_->RegisterPublicBlobURL(url, std::move(blob));
  urls_.insert(url);
  std::move(callback).Run();
}
```

The patch for https://crbug.com/933089, as the comment above says, is supposed to suppress errors when access checks fail because the evaluated child process is being closed and some of the associated data structures have already been deleted[2]. Unfortunately, it actually skips the checks completely for shutting down processes.

Therefore, a compromised renderer can intentionally send a request when it's about to get closed and, for example, register a blob URL for another website. If an attacker then tries to navigate to the URL, it will be loaded by the other website's process. Since the contents of the blob is attacker-controlled, this allows executing arbitrary JavaScript in the context of the other website.

Note that `BlobURLStoreImpl` unregisters all of its URLs in the destructor[1], so it's important that the page load happens before the destructor call.

The other functions modified by the patch might be affected in a similar way.

VERSION
Google Chrome 79.0.3945.79 (Official Build) (64-bit)
Chromium 81.0.3991.0 (Developer Build) (64-bit)

REPRODUCTION CASE
The patch that simulates a compromised renderer does quite a few things to make the issue reproduce more reliably:
- Modifies the URL in the blob registration request to have the "https://www.google.com/" origin.
- Disables the renderer-side URL check that prevents other websites' blob URLs from loading. It
- probably makes sense to add a similar browser-side check as well.
- Sends a frame detachment notification to the browser while keeping the corresponding objects alive
- in the renderer.
- Makes the renderer ignore the SIGTERM signal.
- Disables the process termination on a message channel disconnects.
- Prevents a null pointer dereference caused by an invalid IPC message from the browser.

A compromised renderer process is capable of performing all of the above.

The test case should be accessible from both "localhost" and "127.0.0.1" (to allow using multiple processes) and should be loaded from "localhost". It takes some time to succeed, but it's way below a minute on my workstation.

CREDIT INFORMATION
Sergei Glazunov of Google Project Zero

This bug is subject to a 90 day disclosure deadline. After 90 days elapse or a patch has been made broadly available (whichever is earlier), the bug report will become visible to the public.

repro.html
1.8 KB View Download

renderer-patch.diff
6.4 KB View Download

Comment 1 by mmoroz@chromium.org on Wed, Dec 18, 2019, 10:48 AM EST Project Member

Cc: creis@chromium.org nasko@chromium.org

Comment 2 by mmoroz@chromium.org on Wed, Dec 18, 2019, 10:51 AM EST Project Member

Cc: acolwell@chromium.org ajha@chromium.org mek@chromium.org alex...@chromium.org

Comment 3 by wfh@chromium.org on Wed, Dec 18, 2019, 12:47 PM EST Project Member

Status: Assigned (was: Unconfirmed)
Owner: acolwell@chromium.org
Cc: -acolwell@chromium.org wfh@chromium.org
Labels: Security_Severity-High Security_Impact-Stable OS-Android OS-Chrome OS-Linux OS-Mac OS-Windows Pri-1
Components: Blink>Storage>FileAPI Internals>Sandbox>SiteIsolation Blink>Storage

This seems like a serious SOP bypass. acolwell@chromium.org can you take a look at this as you handled the previous issue referenced in this bug?

Comment 4 by acolwell@chromium.org on Wed, Dec 18, 2019, 1:41 PM EST Project Member

Status: Started (was: Assigned)

Comment 5 by sheriffbot@chromium.org on Thu, Dec 19, 2019, 9:28 AM EST Project Member

Labels: Target-79 M-79

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

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

Comment 6 by alexmos@google.com on Fri, Dec 20, 2019, 2:08 PM EST Project Member

Cc: lukasza@chromium.org

Thanks a lot for this report! Quick status update here before the holidays. Aaron has a draft CL (https://chromium-review.googlesource.com/c/chromium/src/+/1975165) in progress to fix the main aspect of the problem and avoid skipping ChildProcessSecurityPolicy checks for processes that are shutting down. The idea is to extend the lifetime of ChildProcessSecurityPolicyImpl::SecurityState in cases like blob storage by handing out handle objects. We don't want to land this before the holidays, as we won't be around to monitor stability, so we'll come back to this in the new year.

Separately, another area we hadn't considered before is a malicious renderer's ability to stay alive after the browser process has destroyed the corresponding RenderProcessHost, and how we can prevent that going forward. I'm not familiar with the shutdown logic, but one question we've discussed is why we use SIGTERM and whether it's possible to use SIGKILL instead, which a malicious renderer won't be able to ignore. Apparently, base::Process::Terminate has logic to use both (https://cs.chromium.org/chromium/src/base/process/process_posix.cc?l=322-330&rcl=04269f0e1f58afc421b332ad5dd6e1ae35a18499), but (1) RenderProcessHostImpl::Shutdown doesn't appear to ever pass the |wait| flag as true, which prevents ever using SIGKILL, and (2) even if it did, the 60-second delay seems too high.

Comment 7 by livvielin@chromium.org on Thu, Jan 9, 2020, 2:26 PM EST Project Member

Security marshal here, just want to make sure there's progress being made on this? Thanks for your help!

Comment 8 by bugdroid on Thu, Jan 16, 2020, 2:12 AM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/4fcbe415172be634fee82ecb300e50f67b27f0b1

commit 4fcbe415172be634fee82ecb300e50f67b27f0b1

Author: Aaron Colwell <acolwell@chromium.org>
Date: Thu Jan 16 07:10:59 2020

Introduce ChildProcessSecurityPolicyImpl::Handle.

This change introduces a Handle object so that Mojo services can
preserve the security state beyond the lifetime of the
RenderProcessHostImpl object. This allows consistent security
checks to occur even during the period when the renderer process is
shutting down and there are still pending Mojo operations in flight.
This will be used to remove all remaining uses of
ChildProcessSecurityPolicyImpl::HasSecurityState() in follow-up CLs.

- Implements new Handle object that allows security checks to provide
  consistent results after ChildProcessSecurityPolicyImpl::Remove() is
  called.
- Convert blob code to use Handle instead of the HasSecurityState()
  workaround.

This is an updated version of https://crrev.com/c/1534368 . Further
discussion of the history and reasons for this CL can be found there.

~~Bug: 1035399~~, ~~943887~~
Change-Id: I6165fad4308643a1ddc845690443e8efceac65f4
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1975165
Reviewed-by: Aaron Colwell <acolwell@chromium.org>
Reviewed-by: Alex Moshchuk <alexmos@chromium.org>
Reviewed-by: Łukasz Anforowicz <lukasza@chromium.org>
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Aaron Colwell <acolwell@chromium.org>
Cr-Commit-Position: refs/heads/master@{#732296}

[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/blob_storage/blob_registry_wrapper.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/browser_context.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/child_process_security_policy_impl.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/child_process_security_policy_impl.h
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/child_process_security_policy_unittest.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/content/browser/site_instance_impl_unittest.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/storage/browser/blob/blob_registry_impl.h
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/storage/browser/blob/blob_url_store_impl.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/storage/browser/blob/blob_url_store_impl_unittest.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/storage/browser/test/mock_blob_registry_delegate.cc
[modify] https://crrev.com/4fcbe415172be634fee82ecb300e50f67b27f0b1/storage/browser/test/mock_blob_registry_delegate.h

Comment 9 by dominickn@chromium.org on Wed, Jan 22, 2020, 12:33 PM EST Project Member

Friendly security marshall ping: is #8 a sufficient mitigation to mark this bug Fixed?

Comment 10 by acolwell@chromium.org on Wed, Jan 22, 2020, 12:47 PM EST Project Member

Status: Fixed (was: Started)

Yes. I believe so. Marking as fixed.

Comment 11 by sheriffbot@chromium.org on Wed, Jan 22, 2020, 12:50 PM EST Project Member

Labels: Merge-Request-80

Requesting merge to beta M80 because latest trunk commit (732296) appears to be after beta branch point (722274).

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

Comment 12 by sheriffbot@chromium.org on Wed, Jan 22, 2020, 12:52 PM EST Project Member

Labels: -Merge-Request-80 Merge-Review-80 Hotlist-Merge-Review

This bug requires manual review: We are only 12 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://goto.google.com/chrome-release-branch-merge-guidelines
- 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 master/ToT?
4. Why are these changes required in this milestone after branch?
5. Is this a new feature?
6. If it is a new feature, is it behind a flag using finch?

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

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

Comment 13 by srinivassista@google.com on Wed, Jan 22, 2020, 12:58 PM EST Project Member

acolwell@ can you pls answer the questions in comment #12 for merge review.

Comment 14 by acolwell@chromium.org on Wed, Jan 22, 2020, 2:04 PM EST Project Member

Why did this automatically get a merge request? I don't really feel confident that this would be safe to merge so close to the stable release.

1. I believe this meets merge criteria because it is a security fix. I do wonder if this really should be high priority for merging to beta since the exploit requires disabling several different mechanisms in the renderer to actually allow the exploit to be successful.
2. https://chromium-review.googlesource.com/c/chromium/src/+/1975165
3. Yes
4. This is a security fix.
5. No
6. No

Comment 15 by creis@chromium.org on Wed, Jan 22, 2020, 3:12 PM EST Project Member

Cc: awhalley@google.com

Just chiming in on the merge decision. The auto-merge-request was likely triggered because this is a High severity issue, and I think that classification is accurate. We have upgraded Site Isolation bypasses to High in https://chromium.googlesource.com/chromium/src/+/master/docs/security/severity-guidelines.md, and this is a case where a compromised renderer process can run JavaScript in an arbitrary victim origin (essentially a UXSS). It may be a long-standing issue, but it's worth fixing quickly if we can.

There's still room to debate whether the fix is too risky to merge. It sounds like it has been baking for 6 days without problems, which is a good sign. Still, there is a chance that the additional restrictions will lead to some unexpected renderer kills. We could try to monitor for that if we move forward with the merge.

I'm inclined to proceed; awhalley@, do you want to weigh in?

Comment 16 by srinivassista@google.com on Wed, Jan 22, 2020, 4:36 PM EST Project Member

Cc: adetaylor@chromium.org

adetaylor@ to chime in his thoughts too.

Comment 17 by adetaylor@chromium.org on Wed, Jan 22, 2020, 5:53 PM EST Project Member

I'm happy to go with Charlie's judgement in #c15 - which is on balance that we should merge this, and keep an eye out for renderer kills.

Comment 18 by acolwell@chromium.org on Wed, Jan 22, 2020, 5:57 PM EST Project Member

ok. Sounds good. I'll start prepping the merge CL.

Comment 19 by srinivassista@google.com on Thu, Jan 23, 2020, 11:48 AM EST Project Member

adetaylor@ what would be our mitigation plan if we see renderer kills?

Comment 20 by sheriffbot@chromium.org on Thu, Jan 23, 2020, 12:25 PM EST Project Member

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

Comment 21 by adetaylor@chromium.org on Thu, Jan 23, 2020, 2:37 PM EST Project Member

Re #c19 I'm assuming we will be on the lookout for such renderer kills *before* M80 actually releases, in which case we could do a sudden revert. By the sounds of it, this is unlikely though.

Comment 22 by alex...@chromium.org on Thu, Jan 23, 2020, 3:59 PM EST Project Member

This is quite a large change to merge, but it also looks like it's a clean merge, so I'm also ok with merging and monitoring for kills, given the security implications.
srinivassista@: Will there be an additional M80 beta release that would include this merge, before the stable cut?

Comment 23 by srinivassista@google.com on Thu, Jan 23, 2020, 4:23 PM EST Project Member

alexmos@ yes last beta release is next week wednesday, ideally we would take that build to stable the foillowing week but if needed we might do a new build to fix critical issues. Net/Net you have 2 days to monitor beta and see if anything would be broken.

Can we monitor for these on early channels dev/canary?

Comment 24 by acolwell@chromium.org on Thu, Jan 23, 2020, 4:44 PM EST Project Member

I have been monitoring canary & dev channels since the patch landed. I haven't seen any renderer kills related to this change. I'd expect to see a spike in "[Renderer kill 123] mojo::`anonymous namespace\'::RunErrorCallback - Non committable URL passed to BlobURLStore::" crashes.

We could monitor it with a query like https://crash.corp.google.com/browse?q=expanded_custom_data.ChromeCrashProto.magic_signature_1.name+LIKE+%27%5BRenderer+kill+123%5D+mojo%3A%3A%60anonymous+namespace%5C%27%3A%3ARunErrorCallback+-+Non+committable+URL+passed+to+BlobURLStore%3A%3A%25%27+AND+product.Version%3E%3D%2780.0.0.0%27

Comment 25 by srinivassista@google.com on Fri, Jan 24, 2020, 11:41 AM EST Project Member

Labels: -Merge-Review-80 Merge-Approved-80

Thank you for the confirmation,

Merge approved for M80 branch:3987 , pls merge your changes asap. Lets closely monitor beta channel next week . beta release is scheduled for wednesday morning next week.

Comment 26 by creis@chromium.org on Fri, Jan 24, 2020, 12:42 PM EST Project Member

Great-- I'll try to land the merge CL momentarily and our team will monitor it, since acolwell is OOO.

Comment 27 by creis@chromium.org on Fri, Jan 24, 2020, 1:05 PM EST Project Member

I reviewed the merge as well and put it in the merge CQ, after confirming there are still no new crashes shown in Aaron's query from comment 24 since r732296 landed in 81.0.4030.0.

Comment 28 by bugdroid on Fri, Jan 24, 2020, 2:03 PM EST Project Member

Labels: -merge-approved-80 merge-merged-3987 merge-merged-80

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3

commit d6c8e1d8d9b93c099bf11796ddcf6183beef65c3
Author: Aaron Colwell <acolwell@chromium.org>
Date: Fri Jan 24 19:02:28 2020

Introduce ChildProcessSecurityPolicyImpl::Handle.

Merging to M80 branch

This change introduces a Handle object so that Mojo services can preserve the security state beyond the lifetime of the RenderProcessHostImpl object. This allows consistent security checks to occur even during the period when the renderer process is shutting down and there are still pending Mojo operations in flight. This will be used to remove all remaining uses of ChildProcessSecurityPolicyImpl::HasSecurityState() in follow-up CLs.

- Implements new Handle object that allows security checks to provide consistent results after ChildProcessSecurityPolicyImpl::Remove() is called.
- Convert blob code to use Handle instead of the HasSecurityState() workaround.

This is an updated version of https://crrev.com/c/1534368 . Further discussion of the history and reasons for this CL can be found there.

(cherry picked from commit 4fcbe415172be634fee82ecb300e50f67b27f0b1)

~~Bug: 1035399~~, ~~943887~~
Change-Id: I6165fad4308643a1ddc845690443e8efceac65f4
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1975165
Reviewed-by: Aaron Colwell <acolwell@chromium.org>
Reviewed-by: Alex Moshchuk <alexmos@chromium.org>
Reviewed-by: Łukasz Anforowicz <lukasza@chromium.org>
Reviewed-by: Marijn Kruisselbrink <mek@chromium.org>
Commit-Queue: Aaron Colwell <acolwell@chromium.org>

Cr-Original-Commit-Position: refs/heads/master@{#732296}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2015725
Commit-Queue: Charlie Reis <creis@chromium.org>
Cr-Commit-Position: refs/branch-heads/3987@{#702}
Cr-Branched-From: c4e8da9871cc266be74481e212f3a5252972509d-refs/heads/master@{#722274}

[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/blob_storage/blob_registry_wrapper.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/browser_context.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/child_process_security_policy_impl.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/child_process_security_policy_impl.h
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/child_process_security_policy_unittest.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/content/browser/site_instance_impl_unittest.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/storage/browser/blob/blob_registry_impl.h
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/storage/browser/blob/blob_url_store_impl.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/storage/browser/blob/blob_url_store_impl_unittest.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/storage/browser/test/mock_blob_registry_delegate.cc
[modify] https://crrev.com/d6c8e1d8d9b93c099bf11796ddcf6183beef65c3/storage/browser/test/mock_blob_registry_delegate.h

Comment 29 by creis@chromium.org on Mon, Jan 27, 2020, 5:44 PM EST Project Member

I'm continuing to monitor the crash reports, and I think things still look good to proceed.

More specifically, there have been previous crash reports with the "[Renderer kill 123] mojo::`anonymous namespace\'::RunErrorCallback - Non committable URL passed to BlobURLStore::" signature [1], before Aaron's r732296 landedin 81.0.4030.0. His CL makes things stricter and isn't designed to fix the corner cases where those crashes were happening (e.g., if the BrowserContext goes away before all incoming Mojo messages are handled [2]), so it's not surprising if we continue to see a few. We will mainly be concerned if there's a spike in crashes from the new restrictions, and that doesn't seem to be happening.

Since Friday, there have been 2 new reports, one on 80.0.3987.18 without Aaron's CL present (crash/598d9fe88037766a), and one on 81.0.4038.2 with Aaron's CL present (crash/d57a662b45171747). That isn't an indication of a spike or that things are any worse, so I think we're still safe to proceed with the new security restrictions. I'll continue to watch as the new beta goes out on Wednesday.

[1] https://crash.corp.google.com/browse?q=expanded_custom_data.ChromeCrashProto.magic_signature_1.name+LIKE+%27%5BRenderer+kill+123%5D+mojo%3A%3A%60anonymous+namespace%5C%27%3A%3ARunErrorCallback+-+Non+committable+URL+passed+to+BlobURLStore%3A%3A%25%27
[2] https://chromium-review.googlesource.com/c/chromium/src/+/1975165/5/content/browser/child_process_security_policy_impl.cc#1733

Comment 30 by mmoroz@chromium.org on Tue, Jan 28, 2020, 11:41 AM EST Project Member

Labels: VulnerabilityAnalysis-Requested

acolwell@, thank you for fixing this issue. Chrome Security team needs your knowledge to prevent that whole class of bugs from happening elsewhere. We would greatly appreciate if you could tell us more about the issue by filling out the following form: https://forms.gle/VWKDUv9a8GXCCRWm7

Comment 31 by creis@chromium.org on Fri, Jan 31, 2020, 6:42 PM EST Project Member

Labels: -Hotlist-Merge-Review

For crashes, this is still looking good. No crashes on 80.0.3987.78 Beta with Aaron's CL, and only one new report since the 27th (on 81.0.4044.0 Canary). (Also, I'll remove the Hotlist-Merge-Review label, since this has already been merged.)

#c30: Good idea! I'll try to help fill out the form, since acolwell@ is OOO for a bit. I'll also note that alexmos@ is landing followup fixes for the other GetSecurityState cases in ~~issue 943887~~.

Comment 32 by adetaylor@google.com on Sat, Feb 1, 2020, 8:13 PM EST Project Member

Labels: Release-0-M80

Comment 33 by adetaylor@chromium.org on Mon, Feb 3, 2020, 6:46 PM EST Project Member

Labels: CVE-2020-6385 CVE_description-missing

Comment 34 by adetaylor@chromium.org on Mon, Feb 10, 2020, 4:36 PM EST Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 35 by adetaylor@google.com on Wed, Mar 4, 2020, 1:44 PM EST Project Member

Cc: achuith@chromium.org

Comment 36 by sheriffbot on Thu, Apr 30, 2020, 2:54 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