☆ Starred by 3 users

Owner:	arthu...@chromium.org
CC:	mkwst@chromium.org karandeepb@chromium.org rdevl...@chromium.org janag...@google.com chama...@gmail.com arthu...@chromium.org antoniosartori@chromium.org benwells@chromium.org solomonkinard@chromium.org tjudkins@chromium.org
Status:	Fixed *(Closed)*
Components:	Platform>Extensions Blink>SecurityFeature>ContentSecurityPolicy
Modified:	Feb 9, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	10
NextAction:	2020-10-08
OS:	Linux, Windows, Chrome, Mac
Pri:	1
Type:	Bug-Security

reward-3000
Security_Impact-Stable
Security_Severity-Medium
allpublic
reward-inprocess
CVE_description-submitted
Target-88
Target-85
Target-86
M-88
merge-merged-4240
merge-merged-86
LTC-Merged-86
LTS-Security-86
Release-0-M88
CVE-2021-21127

Issue 1115590: CSP Bypass via Chrome Extension

Reported by ghuli...@gmail.com on Wed, Aug 12, 2020, 1:24 PM EDT

Code

VULNERABILITY DETAILS

CSP(Content-Security-Policy) has a header frame-ancestors which blocks click jacking and UI Redressing Vulnerabilities. However, I have discovered a way to bypass the frame-ancestors header via Chrome Extension.

VERSION
Chrome Version: 84.0.4147.125 (Official Build) (64-bit)
Operating System: Windows 10 Home

REPRODUCTION CASE
Bypassing Frame-ancestor header:
1. Create a PHP file which has the frame-ancestors header set.(No matter whats the value of header. self,none,url) .File attached (mysite_csp.php). You can use any other site which has this header set or just upload this file to some url.
2. Download the exploit code, csp_bypass_extension.zip chrome extension . Turn ON Developer Mode at Chrome Extensions page and load the extracted extension folder by using Load Unpacked button.
3. Open the frame-ancestors URL and click the extension button, this extension will bypass the CSP and load the URL in extension popup. Popup.js file has the simple code to load the URL in iframe.

Bypassing X-Frame-Options header
1. Download mysite_xframe.php file and upload somewhere.
2. Open URL and click the extension button and it will load the URL in iframe aswell. Basically if frame-ancestors is present with x-frame-options. X-Frame-Options header will get bypassed.

Video POC with Reproduction Steps : https://youtu.be/bdbFQk4b2po
URL with CSP frame-ancestors set : https://jasminderpalsingh.info/pentest/mysite_csp.php
URL With X-frame-options set : https://jasminderpalsingh.info/pentest/mysite_xframe.php

CREDIT INFORMATION
Reporter credit: Jasminder Pal Singh, Web Services Point WSP, Kotkapura

csp_bypass_extension.zip
28.2 KB Download

mysite_csp.php
262 bytes View Download

mysite_xframe.php
229 bytes View Download

Comment 1 by vakh@chromium.org on Thu, Aug 13, 2020, 7:27 PM EDT Project Member

Status: Available (was: Unconfirmed)
Cc: arthu...@chromium.org rdevl...@chromium.org
Components: Blink>SecurityFeature>ContentSecurityPolicy Platform>Extensions

I'm not sure if this is an issue in Chrome Extensions or CSP handling so adding both components and owners.
Could one of you please help me with the triage here? I'd really appreciate it.

Comment 2 by arthu...@chromium.org on Fri, Aug 14, 2020, 4:17 AM EDT Project Member

Owner: arthu...@chromium.org
Cc: mkwst@chromium.org antoniosartori@chromium.org
Labels: OS-Chrome OS-Linux OS-Mac OS-Windows Pri-3

+mkwst@ and +antoniosartori@
+karandeepb@ FYI, maybe this might have an interest to you relatively to ~~issue 806041~~.

> Basically if frame-ancestors is present with x-frame-options. X-Frame-Options header will get bypassed.

This is expected:
https://www.w3.org/TR/CSP/#frame-ancestors-and-frame-options

```
In order to allow backwards-compatible deployment, the frame-ancestors directive _obsoletes_ the X-Frame-Options header. If a resource is delivered with an policy that includes a directive named frame-ancestors and whose disposition is "enforce", then the X-Frame-Options header MUST be ignored.
```

So, now the real question is: why the CSP:frame-ancestor doesn't apply here in the context of an extension?

Again, I think this is also expected, because the chrome-extension: scheme "bypass" CSP. So frame-ancestor do not block its parent.

```
bool CSPContext::IsAllowedByCsp(mojom::CSPDirectiveName directive_name,
                                const GURL& url,
                                bool has_followed_redirect,
                                bool is_response_check,
                                const mojom::SourceLocationPtr& source_location,
                                CheckCSPDisposition check_csp_disposition,
                                bool is_form_submission) {
  if (SchemeShouldBypassCSP(url.scheme_piece()))
    return true;
```

From what I understood, this is working the way it was designed and implemented. Maybe we should close as Working-As-Intented? (WontFix)
What do you think Mike?

Comment 3 by ghuli...@gmail.com on Fri, Aug 14, 2020, 4:47 AM EDT

Hello,
If that's the case many popular websites are vulnerable to extension based Clickjacking and UI Redressing Vulnerabilities.

Comment 4 by mkwst@chromium.org on Fri, Aug 14, 2020, 5:00 AM EDT Project Member

The goal of the `chrome-extension:` carveout was to allow extension resources to be embeddable despite a page's desire to limit the sources from which it normally loaded resources. The `frame-ancestors` interaction isn't something I explicitly thought about at the time, and I can understand how it's unexpected.

I think we'd ideally carve out only extension resources that have access to the given page. That wasn't possible at the time, but perhaps it's possible today given that the implementation of `frame-ancestors` has shifted up to the browser process.

Comment 5 by ghuli...@gmail.com on Fri, Aug 14, 2020, 5:08 AM EDT

Attack Scenario : In the video POC i mentioned a example website https://tweetdeck.twitter.com/ . Its a twiiter site and victim to this bug. It allows to tweet from logged account in two steps. Its easy to craft a interactive chrome extension by exploiting this bypass.

There are hundreds of other sites that might be vulnerable to this. As per my experience, it should be fixed.

Comment 6 by sheriffbot on Fri, Aug 14, 2020, 3:45 PM EDT Project Member

Status: Assigned (was: Available)

Comment 7 by vakh@chromium.org on Mon, Aug 17, 2020, 6:27 PM EDT Project Member

Labels: Security_Severity-Medium Security_Impact-Stable

Adding Sev-Medium tentatively.

Comment 8 by sheriffbot on Tue, Aug 18, 2020, 2:14 PM EDT Project Member

Labels: Target-85 M-85

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 9 by sheriffbot on Tue, Aug 18, 2020, 2:50 PM EDT 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 10 by ghuli...@gmail.com on Mon, Aug 24, 2020, 3:24 AM EDT

Thanks. Looking forward for a fix and eligibility for reward.

Comment 11 by ghuli...@gmail.com on Thu, Aug 27, 2020, 11:06 PM EDT

Hello Team,

Is it possible to continue the reward process parallel to fix ?
I want to make chromium research full time because i love it. On the other hand i want to make the survival out of it. So, is it possible to reward the researchers(if eligible) as early as possible? Or reward a before-fix amount until bug is fixed and final amount after fix. It will help the researchers like me to keep the motivation strong.

In my case, i want to buy a Chromebook to continue my further research but here i am waiting since two weeks.

I hope you will take care of this for the sake to encourage researchers.

Thanks

Comment 12 by sheriffbot on Fri, Aug 28, 2020, 1:37 PM EDT Project Member arthursonzogni: 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 arthu...@chromium.org on Wed, Sep 2, 2020, 6:10 AM EDT Project Member

Cc: benwells@chromium.org

> I think we'd ideally carve out only extension resources that have access to the given page. That wasn't possible at the time, but perhaps it's possible today given that the implementation of `frame-ancestors` has shifted up to the browser process.

This sounds like good trade-off.

+CC:benwells@ I am not super familiar with extensions/.
Could you point out a function to check if a given extension have access to a URL/Origin/iframe?

Comment 14 by ghuli...@gmail.com on Mon, Sep 7, 2020, 11:58 AM EDT

Looking forward for a fix.

Thanks

Comment 15 by ghuli...@gmail.com on Tue, Sep 15, 2020, 10:27 AM EDT

Looking for an update.

Comment 16 by sheriffbot on Wed, Sep 16, 2020, 1:37 PM EDT Project Member

arthursonzogni: 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 17 by ghuli...@gmail.com on Tue, Sep 22, 2020, 12:37 PM EDT

Looking forward for an update.

Comment 18 by arthu...@chromium.org on Tue, Sep 22, 2020, 1:41 PM EDT Project Member

+benwells@

I am not familiar with extensions/.
Is the sentence: "extension have access to the given page" defined in a given way?
If yes, do you happen to know how I can retrieve this information?

Comment 19 by benwells@chromium.org on Tue, Sep 22, 2020, 11:07 PM EDT Project Member

Cc: karandeepb@chromium.org

I think you need someone from the core extensions team to chime in, either Karan or Devlin should be able to help.

Comment 20 by karandeepb@chromium.org on Wed, Sep 23, 2020, 1:15 AM EDT Project Member

I don't think this is WAI. As Mike mentioned, the chrome-extension scheme bypassing is required for things like web accessible resources to work. This is how an extension can inject a web accessible iframe into a page without it being blocked by the page's CSP.

I am not sure I understand how the check is working correctly? Is it that the frame-ancestors is being checked against each parent frame and when it comes to the extension's frame we skip the check because the extension scheme is a CSP bypassing scheme? If yes, I don't think the notion of a CSP bypassing scheme is being followed correctly here. It's described as:

```
    // Registers a URL scheme whose resources can be loaded regardless of a
    // page's Content Security Policy.
```

at
https://source.chromium.org/chromium/chromium/src/+/master:content/public/common/content_client.h;l=134;drc=c240cfc4b578dc6268356551922ba73762294795;bpv=1;bpt=0.

Hence I think the notion of a csp bypassing scheme should only be applicable to resources being loaded from that scheme and not to the urls of the ancestor frames in this case.

Comment 21 by arthu...@chromium.org on Wed, Sep 23, 2020, 3:46 AM EDT Project Member

> Is it that the frame-ancestors is being checked against each parent frame and when it comes to the extension's frame we skip the check because the extension scheme is a CSP bypassing scheme?

Yes.

--

So you are suggesting to apply CSP for frame-ancestor no matter the extension?
That would be very trivial.

Mike suggested to do it only if the extension have access to the embedded document. I am not sure to see what it really means. Is there any meaningful context were we can say that?

Comment 22 by karandeepb@chromium.org on Wed, Sep 23, 2020, 4:11 PM EDT Project Member

> So you are suggesting to apply CSP for frame-ancestor no matter the extension?
Yes.

> Mike suggested to do it only if the extension have access to the embedded document. I am not sure to see what it really means. Is there any meaningful context were we can say that?

Yes. There is a concept of host permissions which an extension can specify and a user can modify (https://developer.chrome.com/extensions/runtime_host_permissions).
IIUC the suggestion was for the extension to be able to embed a frame if it had access to a frame regardless of X-Frame-Options/frame-ancestors. I think this is a good thing to do given that:
- There are legit use cases for an extension to be able to embed a frame and if it has host permissions to a frame then its reasonable to bypass the frame-src and X-Frame-

Options restriction.
- Currently if an extension has access to a page, it can already modify its X-Frame-Options and CSP header using the web request API to make this possible. However this is not ideal and in Manifest V3, we are hoping to prevent the extension from relaxing the CSP. If we automatically allow the extension to embed frames to which it has permission, it won't need to modify these headers.

TLDR: I think taking host permissions into account would be a net positive change, however I am not sure if it needs to necessarily block fixing this bug anc can probably be tackled separately.

Also do we know the relative prevalence of frame-ancestors vs X-Frame-Options (IIUC the latter is not exempting extension parent frames)? Want to get a sense of whether there might be extensions depending on this behavior.

Comment 23 by ghuli...@gmail.com on Mon, Sep 28, 2020, 1:03 PM EDT

Looking forward for an update.

Comment 24 by arthu...@chromium.org on Tue, Sep 29, 2020, 9:03 AM EDT Project Member

EstimatedDays: 10
NextAction: 2020-10-08

Thanks karandeepb@ (comment 22)

I will make Chrome enforce CSP:frame-ancestor no matter the scheme of its parent (even extension).
I am a bit busy now, so I am expecting a 10 days delay before landing this.

Comment 25 by sheriffbot on Wed, Oct 7, 2020, 1:37 PM EDT Project Member

Labels: -M-85 M-86 Target-86

Comment 26 by ghuli...@gmail.com on Wed, Oct 7, 2020, 8:12 PM EDT

Looking forward for an update.

Thanks.

Comment 27 by monor...@bugs.chromium.org on Thu, Oct 8, 2020, 8:00 AM EDT

The NextAction date has arrived: 2020-10-08

Comment 28 by ghuli...@gmail.com on Sun, Oct 11, 2020, 11:37 AM EDT

Looking forward for a fix this week.

Thanks

Comment 29 by arthu...@chromium.org on Tue, Oct 13, 2020, 3:11 AM EDT Project Member

Status: Started (was: Assigned)

(I'm back from vacation)

Let's fix this for today.

Comment 30 by arthu...@chromium.org on Tue, Oct 13, 2020, 8:20 AM EDT Project Member

> (I'm back from vacation)
>
> Let's fix this for today.

Patch:
https://chromium-review.googlesource.com/c/chromium/src/+/2467897

Comment 31 by ghuli...@gmail.com on Wed, Oct 14, 2020, 1:21 PM EDT

Thanks for the patch.
Looking forward to get the patch reviewed.

Comment 32 by karandeepb@chromium.org on Thu, Oct 15, 2020, 12:16 PM EDT Project Member

One point regarding the security severity: I am not sure if it is medium. The extension can still (even after the patch) modify the CSP header using the web request API to relax frame-ancestors and embed iframes. However doing so would require host permissions to the iframe url.

Comment 33 by bugdroid on Fri, Oct 16, 2020, 6:01 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/fd406771ba8565e5c58563b5492d45fe2ba5a3c8

commit fd406771ba8565e5c58563b5492d45fe2ba5a3c8
Author: arthursonzogni <arthursonzogni@chromium.org>
Date: Fri Oct 16 09:59:28 2020

[CSP] Do not bypass CSP:frame-ancestors

Extensions can load their own internal content into the document. They shouldn't be blocked by the document's CSP.

There is an exception: CSP:frame-ancestors. This one is not about allowing a document to embed other resources. This is about being embedded. As such this shouldn't be bypassed. A document should be able to deny being embedded inside an extension.

~~Bug: 1115590~~
Change-Id: I2176a25e67cd0d637ecb3b13a39de30259d9d7a1
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2467897
Reviewed-by: Arthur Sonzogni <arthursonzogni@chromium.org>
Reviewed-by: Karan Bhatia <karandeepb@chromium.org>
Reviewed-by: Antonio Sartori <antoniosartori@chromium.org>
Commit-Queue: Arthur Sonzogni <arthursonzogni@chromium.org>
Cr-Commit-Position: refs/heads/master@{#817859}

[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/browser/extensions/extension_csp_bypass_browsertest.cc
[add] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/test/data/extensions/csp/frame-ancestors-none.html
[add] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/test/data/extensions/csp/frame-ancestors-none.html.mock-http-headers
[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/services/network/public/cpp/content_security_policy/content_security_policy.cc
[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/services/network/public/cpp/content_security_policy/csp_context.cc

Comment 34 by arthu...@chromium.org on Fri, Oct 16, 2020, 6:12 AM EDT Project Member

Status: Fixed (was: Started)

Do you think we should attempt merging this to M87 beta? (release date 2020-11-10)
It would be released on M88 otherwise (release date 2021-01-19)

I think this update can potentially break some extensions. So I think not cherry-picking and let it go through canary/dev to be better.
WDYT?

> One point regarding the security severity: I am not sure if it is medium.

I don't know how to judge severity of this myself when this is about Extensions.
Extensions have an enormous amount of privileges already, this is removing tiny one. I wouldn't consider users safe if they install a malicious extension. The web store checking/banning extensions is the only reasonable barrier I can think of.

(I will let others judge about the Security_Severity of this.)

Comment 35 by karandeepb@chromium.org on Fri, Oct 16, 2020, 6:49 AM EDT Project Member

> I think this update can potentially break some extensions. So I think not cherry-picking and let it go through canary/dev to be better.
WDYT?

I agree, especially since this is not a regression.

Comment 36 by sheriffbot on Fri, Oct 16, 2020, 3:08 PM EDT Project Member

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

Comment 37 by ghuli...@gmail.com on Sun, Oct 18, 2020, 3:12 PM EDT

Thanks for the fix. Looking forward for the release and reward process.

Comment 38 by adetaylor@google.com on Sun, Oct 18, 2020, 4:52 PM EDT Project Member

Labels: reward-topanel

Comment 39 by sheriffbot on Mon, Oct 19, 2020, 2:20 PM EDT Project Member

Labels: Merge-Request-87

Requesting merge to beta M87 because latest trunk commit (817859) appears to be after beta branch point (812852).

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

Comment 40 by sheriffbot on Mon, Oct 19, 2020, 2:23 PM EDT Project Member

Labels: -Merge-Request-87 Merge-Review-87 Hotlist-Merge-Review

This bug requires manual review: M87's targeted beta branch promotion date has already passed, so this requires manual review
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), cindyb@(ChromeOS), lakpamarthy@(Desktop)

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

Comment 41 by lakpamarthy@google.com on Tue, Oct 20, 2020, 2:39 AM EDT Project Member

arthursonzogni@ - please respond to the merge questionnaire in c#40 to consider the merge request

Comment 42 by arthursonzogni@google.com on Tue, Oct 20, 2020, 4:36 AM EDT Project Member

Labels: -Hotlist-Merge-Review -Merge-Review-87

> arthursonzogni@ - please respond to the merge questionnaire in c#40 to consider the merge request

See comment 34 and comment 35. We don't want to merge.

Comment 43 by adetaylor@google.com on Wed, Oct 21, 2020, 7:12 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-3000

*** Boilerplate reminders! ***
Please do NOT publicly disclose details until a fix has been released to all our users. Early public disclosure may cancel the provisional reward. Also, please be considerate about disclosure when the bug affects a core library that may be used by other products. Please do NOT share this information with third parties who are not directly involved in fixing the bug. Doing so may cancel the provisional reward. Please be honest if you have already disclosed anything publicly or to third parties. Lastly, we understand that some of you are not interested in money. We offer the option to donate your reward to an eligible charity. If you prefer this option, let us know and we will also match your donation - subject to our discretion. Any rewards that are unclaimed after 12 months will be donated to a charity of our choosing.

Please contact security-vrp@chromium.org with any questions.
****

Comment 44 by adetaylor@google.com on Wed, Oct 21, 2020, 7:32 PM EDT Project Member

ghulianisikh@gmail.com - Congratulations, the VRP panel has awarded $3000 for this report. Someone from our finance team will be in touch.

Comment 45 by ghuli...@gmail.com on Wed, Oct 21, 2020, 11:43 PM EDT

Thank you so much for the reward. I really appreciate 🙏
Looking forward to be contacted by finance team.

Comment 46 by ghuli...@gmail.com on Thu, Oct 22, 2020, 1:36 AM EDT

Please, also check my comment #11 if that's possible.

Comment 47 by adetaylor@google.com on Thu, Oct 22, 2020, 12:27 PM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 48 by adetaylor@google.com on Thu, Oct 22, 2020, 12:33 PM EDT Project Member

Aha, I hadn't spotted #c11. Yes, I understand where you're coming from, but I'm afraid our rules are pretty firm here. It's often hard for the VRP panel to determine the correct reward amount until the fix has actually landed, so we can't and don't give early payouts. Sorry!

This reward is now submitted to our finance team and they will get in touch. I should warn you... that process can sometimes take a few weeks as well, for initial enrollment. Assuming you find lots more valid bugs, subsequent rewards should be much more efficient :)

Comment 49 by ghuli...@gmail.com on Thu, Oct 22, 2020, 2:04 PM EDT

No worries. I am working on converting assumptions to reality. :)
Thanks :)

Comment 50 by adetaylor@google.com on Sun, Nov 22, 2020, 8:33 PM EST Project Member

Cc: chama...@gmail.com

cc chamal.desilva@gmail.com as requested by e-mail, given that this bug is cited from ~~issue 1134338~~.

Comment 51 by ghuli...@gmail.com on Tue, Dec 22, 2020, 11:31 PM EST

Wondering, when its getting patched in stable ?

Comment 52 by rdevl...@chromium.org on Wed, Dec 23, 2020, 5:37 PM EST Project Member

The fix landed in M88, which is currently slated to reach stable on Jan 19.

Comment 53 by adetaylor@google.com on Wed, Jan 13, 2021, 5:48 PM EST Project Member

Labels: Release-0-M88

Comment 54 by amyressler@google.com on Tue, Jan 19, 2021, 1:56 PM EST Project Member

Labels: CVE-2021-21127 CVE_description-missing

Comment 55 by janag...@google.com on Wed, Jan 20, 2021, 7:23 AM EST Project Member

Cc: janag...@google.com
Labels: LTS-Security-86 Merge-Request-86-LTS

Comment 56 by gianluca@google.com on Wed, Jan 20, 2021, 12:01 PM EST Project Member

Labels: Merge-Approved-86-LTS

Comment 57 by sheriffbot on Wed, Jan 20, 2021, 12:22 PM EST Project Member

Labels: -M-86 Target-88 M-88

Comment 58 by bugdroid on Thu, Jan 21, 2021, 1:38 PM EST Project Member

Labels: merge-merged-4240 merge-merged-86

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/5eaa1618eb73c224ad154f7297bb200698e44472

commit 5eaa1618eb73c224ad154f7297bb200698e44472
Author: arthursonzogni <arthursonzogni@chromium.org>
Date: Thu Jan 21 18:38:43 2021

[CSP] Do not bypass CSP:frame-ancestors

Extensions can load their own internal content into the document. They
shouldn't be blocked by the document's CSP.

There is an exception: CSP:frame-ancestors. This one is not about
allowing a document to embed other resources. This is about being
embedded. As such this shouldn't be bypassed. A document should be able
to deny being embedded inside an extension.

(cherry picked from commit fd406771ba8565e5c58563b5492d45fe2ba5a3c8)

~~Bug: 1115590~~
Change-Id: I2176a25e67cd0d637ecb3b13a39de30259d9d7a1
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2467897
Reviewed-by: Arthur Sonzogni <arthursonzogni@chromium.org>
Reviewed-by: Karan Bhatia <karandeepb@chromium.org>
Reviewed-by: Antonio Sartori <antoniosartori@chromium.org>
Commit-Queue: Arthur Sonzogni <arthursonzogni@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#817859}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2639764
Reviewed-by: Victor-Gabriel Savu <vsavu@google.com>
Commit-Queue: Jana Grill <janagrill@chromium.org>
Cr-Commit-Position: refs/branch-heads/4240@{#1526}
Cr-Branched-From: f297677702651916bbf65e59c0d4bbd4ce57d1ee-refs/heads/master@{#800218}

[modify] https://crrev.com/5eaa1618eb73c224ad154f7297bb200698e44472/services/network/public/cpp/content_security_policy/csp_context.cc
[add] https://crrev.com/5eaa1618eb73c224ad154f7297bb200698e44472/chrome/test/data/extensions/csp/frame-ancestors-none.html
[add] https://crrev.com/5eaa1618eb73c224ad154f7297bb200698e44472/chrome/test/data/extensions/csp/frame-ancestors-none.html.mock-http-headers
[modify] https://crrev.com/5eaa1618eb73c224ad154f7297bb200698e44472/chrome/browser/extensions/extension_csp_bypass_browsertest.cc
[modify] https://crrev.com/5eaa1618eb73c224ad154f7297bb200698e44472/services/network/public/cpp/content_security_policy/content_security_policy.cc

Comment 59 by bugdroid on Fri, Jan 22, 2021, 1:11 AM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/fd406771ba8565e5c58563b5492d45fe2ba5a3c8

commit fd406771ba8565e5c58563b5492d45fe2ba5a3c8
Author: arthursonzogni <arthursonzogni@chromium.org>
Date: Fri Oct 16 09:59:28 2020

[CSP] Do not bypass CSP:frame-ancestors

Extensions can load their own internal content into the document. They
shouldn't be blocked by the document's CSP.

There is an exception: CSP:frame-ancestors. This one is not about
allowing a document to embed other resources. This is about being
embedded. As such this shouldn't be bypassed. A document should be able
to deny being embedded inside an extension.

~~Bug: 1115590~~

Change-Id: I2176a25e67cd0d637ecb3b13a39de30259d9d7a1
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2467897
Reviewed-by: Arthur Sonzogni <arthursonzogni@chromium.org>
Reviewed-by: Karan Bhatia <karandeepb@chromium.org>
Reviewed-by: Antonio Sartori <antoniosartori@chromium.org>
Commit-Queue: Arthur Sonzogni <arthursonzogni@chromium.org>
Cr-Commit-Position: refs/heads/master@{#817859}

[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/services/network/public/cpp/content_security_policy/csp_context.cc
[add] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/test/data/extensions/csp/frame-ancestors-none.html
[add] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/test/data/extensions/csp/frame-ancestors-none.html.mock-http-headers
[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/chrome/browser/extensions/extension_csp_bypass_browsertest.cc
[modify] https://crrev.com/fd406771ba8565e5c58563b5492d45fe2ba5a3c8/services/network/public/cpp/content_security_policy/content_security_policy.cc

Comment 60 by janag...@google.com on Fri, Jan 22, 2021, 3:55 AM EST Project Member

Labels: -Merge-Request-86-LTS -Merge-Approved-86-LTS LTC-Merged-86

Comment 61 by sheriffbot on Fri, Jan 22, 2021, 1:52 PM EST 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 62 by amyressler@google.com on Tue, Feb 9, 2021, 9:27 AM EST Project Member

Labels: -CVE_description-missing CVE_description-submitted