Starred by 3 users

Owner:	xiaoc...@chromium.org
CC:	adetaylor@chromium.org mic...@bentkowski.info vogelheim@chromium.org ikilpatrick@chromium.org benmason@chromium.org pbomm...@chromium.org yosin@chromium.org futhark@chromium.org dcheng@chromium.org achuith@chromium.org style-bugs@google.com mas...@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>CSS Blink>Editing
Modified:	Apr 29, 2020
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Fuchsia
Pri:	1
Type:	Bug-Security

Hotlist-Merge-Review
reward-10000
Security_Impact-Stable
Security_Severity-Medium
allpublic
reward-inprocess
CVE_description-submitted
M-79
Merge-Rejected-79
reward_to-michal_at_bentkowski.info
Release-0-M80
CVE-2020-6391

Issue 1017871: Security: Injecting styles via copy-and-paste

Reported by adetaylor@chromium.org on Thu, Oct 24, 2019, 3:12 PM EDT Project Member

Code

VULNERABILITY DETAILS
An issue in sanitizer can (perhaps) lead to XSS via copy&paste. The destination element needs to be contenteditable.

VERSION
Chrome Version: 77.0.3865.90 stable
Operating System: macOS

REPRODUCTION CASE
See ~~issue 1011959~~ which was about injecting scripts via copy-and-paste, and has now been blocked.

Comment 2 of that bug states that CSS is also potentially injectable, and was not covered by that fix. We need to work out whether injection of CSS has any security risk, so I'm raising this for that sort of triage/consideration.

Comment 1 by adetaylor@chromium.org on Thu, Oct 24, 2019, 3:14 PM EDT Project Member

Cc: mic...@bentkowski.info yosin@chromium.org
Labels: reward_to-michal_at_bentkowski.info

I'm adding reward_to-michal_at_bentkowski.info, but this is probably not eligible for an extra reward as it's already covered by ~~issue 1011959~~. However if we fix this separately in a different release I'd probably expect to credit it in the release notes.

Comment 2 by mic...@bentkowski.info on Thu, Oct 24, 2019, 3:21 PM EDT

Injecting CSS might also have security implications. It is mainly used for exfiltrating data.

Ways to abuse CSS injection have been recently nicely covered in this blog post: https://x-c3ll.github.io/posts/CSS-Injection-Primitives/

Comment 3 by jdeblasio@chromium.org on Thu, Oct 24, 2019, 7:03 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Labels: Security_Needs_Attention-Severity Security_Severity-Medium M-79 Pri-1
Components: Blink>CSS

Tentatively setting sev-medium, but this should be re-assessed once we've thought about it a bit more.

Comment 4 by xiaoc...@chromium.org on Fri, Oct 25, 2019, 5:39 AM EDT Project Member

Tested pasting "foo<style>*{color:red}</style>bar" with https://jsbin.com/mozidoxegi/edit?html,output in other browsers:

- Chrome: No style sanitization at all. everything else turns red.
- Firefox: Same as Chrome
- Safari: Sanitized as following. Basically <style> is computed and then moved as inline styles to the pasted elements

"foobar"

Safari's approach looks reasonable to me, as it applies the style properly to pasted elements while prevents it from being applied on the original content

Comment 5 by xiaoc...@chromium.org on Fri, Oct 25, 2019, 5:50 AM EDT Project Member

Cc: adetaylor@chromium.org

+adetaylor@

As you reviewed ~~issue 1011959~~, could you also take a look at the idea in #4? Thank you!

Comment 6 by adetaylor@google.com on Fri, Oct 25, 2019, 1:46 PM EDT Project Member

xiaochengh@ Thanks for the testing and the proposed solution sounds good to me.

Comment 7 by dcheng@chromium.org on Fri, Oct 25, 2019, 2:07 PM EDT Project Member

Cc: dcheng@chromium.org

Comment 8 by dcheng@chromium.org on Fri, Oct 25, 2019, 2:19 PM EDT Project Member

We used to strip style tags but it looks like this was intentionally changed in https://bugs.chromium.org/p/chromium/issues/detail?id=121163#c31.

While computing the styles before inserting the pasted fragment would be the ideal approach, this was previously quite hard: the problem is we couldn't compute the style without doing layout. This required inserting the fragment into a dummy page, or other complicated solutions.

If there's a better way to do that now, then computing the style and including it inline is the best solution. I'm just worried about how complex it will be…

Comment 9 by mic...@bentkowski.info on Sun, Oct 27, 2019, 11:30 AM EDT

As some sort of exercise, I decided to create a proof of concept of exfiltrating data using <style> and copy-and-paste. In the example, I was able to exfiltrate email address of currently logged in user of Gmail. The code is attached. I also attached a video, showcasing the exploit (and yes, rtuspzmvjsofpm92992@gmail.com is actually one of my throw-away accounts ;)).

main.js
2.9 KB View Download

css-leak.mp4
802 KB View Download

Comment 10 by sheriffbot@chromium.org on Fri, Nov 8, 2019, 9:10 AM EST Project Member

xiaochengh: 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 11 by yosin@chromium.org on Fri, Nov 15, 2019, 6:46 PM EST Project Member

Cc: ikilpatrick@chromium.org

For short time solution, let's simply remove <style> elements from Document Fragment to be pasted in ClipboardCommands::GetFragmentFromClipboard() returned from CreateFragmentFromMarkupWithContext(). This can be done by Document::getElementsByTagName().

For long time solution, in addition to removing <style> tag, we set inline style to elements in fragment.

Let's do short time solution to fix this security issue.

Comment 12 by yosin@chromium.org on Fri, Nov 15, 2019, 7:51 PM EST Project Member

Discuss more with xiaochenghu@, simply remove <style> elements causes issue [1].
Before fixing [1], Blink excluded <style> elements when pasting. After fixing [1], Blink inserts <style> elements.

It seems Safari does[2[:
1. Creating Page
2. Parse markup text into Page
3. Serialize into document fragment from the Page with inserting inline style
4. Inserting fragment at selection excluding <style> element.

I'm not sure why we didn't do this way for fixing [1].

[1] ~~http://crbug.com/121163~~: Pasting from Excel spreadsheet does not keep all formatting.
[2] https://trac.webkit.org/changeset/223440 introduces sanitizeMarkup()

https://trac.webkit.org/browser/webkit/trunk/Source/WebCore/editing/markup.cpp

```
177  std::unique_ptr<Page> createPageForSanitizingWebContent()
178  {
179      auto pageConfiguration = pageConfigurationWithEmptyClients(PAL::SessionID::defaultSessionID());
180
181      auto page = makeUnique<Page>(WTFMove(pageConfiguration));
182      page->setIsForSanitizingWebContent();
```

```
    page->settings().setMediaEnabled(false);
    page->settings().setScriptEnabled(false);
    page->settings().setPluginsEnabled(false);
    page->settings().setAcceleratedCompositingEnabled(false);

    Frame& frame = page->mainFrame();
    frame.setView(FrameView::create(frame, IntSize { 800, 600 }));
    frame.init();

    FrameLoader& loader = frame.loader();
    static char markup[] = "<!DOCTYPE html><html><body></body></html>";
    ASSERT(loader.activeDocumentLoader());
    auto& writer = loader.activeDocumentLoader()->writer();
    writer.setMIMEType("text/html");
    writer.begin();
    writer.insertDataSynchronously(String(markup));
    writer.end();
    RELEASE_ASSERT(page->mainFrame().document()->body());

    return page;
}
String sanitizeMarkup(const String& rawHTML, MSOListQuirks msoListQuirks, Optional<WTF::Function<void(DocumentFragment&)>> fragmentSanitizer)
{
    auto page = createPageForSanitizingWebContent();
    Document* stagingDocument = page->mainFrame().document();
    ASSERT(stagingDocument);

    auto fragment = createFragmentFromMarkup(*stagingDocument, rawHTML, emptyString(), DisallowScriptingAndPluginContent);

    if (fragmentSanitizer)
        (*fragmentSanitizer)(fragment);

    return sanitizedMarkupForFragmentInDocument(WTFMove(fragment), *stagingDocument, msoListQuirks, rawHTML);
}
```

Comment 13 by xiaoc...@chromium.org on Tue, Nov 19, 2019, 12:35 PM EST Project Member

Cc: futhark@chromium.org

Comment 14 by dcheng@chromium.org on Fri, Nov 22, 2019, 1:59 AM EST Project Member

yosin, to give some background, the solution implemented in https://chromium-review.googlesource.com/c/chromium/src/+/1922919 (using a dummy document to sanitize the paste) was originally considered, but a number of Blink engineers felt it was too hacky.

But given that we want this feature (for improved paste) and the security issues, it seems like this is the best solution.

Comment 15 by bugdroid on Fri, Nov 22, 2019, 4:19 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a

commit d96236b5d2bad68a0cc8f62501ba15c38c8cf96a
Author: Xiaocheng Hu <xiaochengh@chromium.org>
Date: Fri Nov 22 21:18:34 2019

Sanitize style elements in clipboard markup

This patch sanitizes clipboard markup before pasting it into document
by removing all pasted style elements and serializing them onto
elements as inline style. In this way, we stop stylesheets in clipboard
markup from being applied to the original elements in the document.

This patch follows the same approach as in WebKit [1]:
- First create a dummy document to insert the markup
- Then computes style and layout in the dummy document
- Re-serialize the dummy document as the markup to be inserted. This
  reuses the code path that we serialize a selection range into
  clipboard, where we need to serialize element computed style into
  inline styles so that the element styles are preserved.
- Make sure all style elements are removed before inserting markup
  into document

This patch also adds a complete test to ensure that content pasted from
Excel is still properly styled, which is the main reason we used to
preserve style elements in clipboard markup [2].

[1] https://trac.webkit.org/changeset/223440
[2] ~~http://crbug.com/121163~~

~~Bug: 1017874~~
Change-Id: I3bb5a4ae7530a3fdef5ba251975e004857c06f1e
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1922919
Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
Reviewed-by: Yoshifumi Inoue <yosin@chromium.org>
Reviewed-by: Kent Tamura <tkent@chromium.org>
Cr-Commit-Position: refs/heads/master@{#718281}

[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/commands/clipboard_commands.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/commands/replace_selection_command.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/commands/replace_selection_command_test.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/editing_style_utilities.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/editing_style_utilities.h
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/create_markup_options.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/create_markup_options.h
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/serialization.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/serialization.h
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.cc
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.h
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/renderer/core/editing/serializers/styled_markup_serializer.cc
[add] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/web_tests/editing/pasteboard/paste-from-excel.html
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/web_tests/editing/pasteboard/paste-head-contents-expected.txt
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/web_tests/editing/pasteboard/paste-head-contents.html
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/web_tests/editing/pasteboard/paste-xss-injection.html
[modify] https://crrev.com/d96236b5d2bad68a0cc8f62501ba15c38c8cf96a/third_party/blink/web_tests/editing/pasteboard/preserve-underline-color-expected.txt

Comment 16 by xiaoc...@chromium.org on Fri, Nov 22, 2019, 5:44 PM EST Project Member

Fixed in M80.

adetaylor: Could you help with re-assessing the security severity, and clearing the Security_Needs_Attention-Severity label? Thanks!

Comment 17 by mic...@bentkowski.info on Fri, Nov 22, 2019, 6:34 PM EST

Hey, the fix can be bypassed using similar trick to the one that was shown in ~~issue 1011059~~.

Reproduction steps:
1. Go to https://jsbin.com/mozidoxegi/edit?html,output
2. Copy to clipboard the following code:

```
A<math>B<a style=display:block>C<title>D<a id="</title><svg><style>*{background:red}</style>">c
```

3. Put cursor in the yellow box - but make sure it is not at the very end.
4. Paste from clipbard; everything turns red.

It seems the problem is that after CompositeEditCommand::MoveParagraphs(), HTMLStyleElement is removed, but the same doesn't happen for SVGStyleElement.

Comment 18 by xiaoc...@chromium.org on Fri, Nov 22, 2019, 6:54 PM EST Project Member

Thanks for the quick catch!

And... In fact CompositeEditCommand::MoveParagraphs() doesn't strip HTMLStyleElement, either. The good old attack in ~~crbug.com/1011059#c2~~ still works

Comment 19 by xiaoc...@chromium.org on Fri, Nov 22, 2019, 7:02 PM EST Project Member

Sorry please ignore comment 18.

Didn't update my local checkout when testing...

Comment 20 by adetaylor@chromium.org on Fri, Nov 22, 2019, 7:39 PM EST Project Member

Labels: -Security_Needs_Attention-Severity

I think Medium seems about right from the discussions. I'll keep it at that.

Comment 21 by mic...@bentkowski.info on Sat, Nov 23, 2019, 3:34 AM EST

Also one more issue: when the pasted style contains an @import rule, then the tab crashes with "Aw, Snap!". In the console I'm getting:

```
Received signal 11 SEGV_MAPERR 000000000000
```

I think that @import should be ignored in pasted content. This is also what Safari does.

The code I'm pasting is:

```
<style>@import'https://example.com';</style>
```

Comment 22 by bugdroid on Mon, Nov 25, 2019, 3:09 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b

commit f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b
Author: Owen Min <zmin@chromium.org>
Date: Mon Nov 25 20:08:58 2019

Revert "Sanitize style elements in clipboard markup"

This reverts commit d96236b5d2bad68a0cc8f62501ba15c38c8cf96a.

Reason for revert: This may cause "WebKit Linux Leak" failure
First failure: https://ci.chromium.org/p/chromium/builders/ci/WebKit%20Linux%20Leak/7276

Original change's description:
> Sanitize style elements in clipboard markup
>
> This patch sanitizes clipboard markup before pasting it into document
> by removing all pasted style elements and serializing them onto
> elements as inline style. In this way, we stop stylesheets in clipboard
> markup from being applied to the original elements in the document.
>
> This patch follows the same approach as in WebKit [1]:
> - First create a dummy document to insert the markup
> - Then computes style and layout in the dummy document
> - Re-serialize the dummy document as the markup to be inserted. This
> reuses the code path that we serialize a selection range into
> clipboard, where we need to serialize element computed style into
> inline styles so that the element styles are preserved.
> - Make sure all style elements are removed before inserting markup
> into document
>
> This patch also adds a complete test to ensure that content pasted from
> Excel is still properly styled, which is the main reason we used to
> preserve style elements in clipboard markup [2].
>
> [1] https://trac.webkit.org/changeset/223440
> [2] ~~http://crbug.com/121162~~
>
> ~~Bug: 1017871~~
> Change-Id: I3bb5a4ae7530a3fdef5ba251975e004857c06f1e
> Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1922919
> Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
> Reviewed-by: Yoshifumi Inoue <yosin@chromium.org>
> Reviewed-by: Kent Tamura <tkent@chromium.org>
> Cr-Commit-Position: refs/heads/master@{#718281}

TBR=yosin@chromium.org,tkent@chromium.org,xiaochengh@chromium.org

Not skipping CQ checks because original CL landed > 1 day ago.

~~Bug: 1017871~~, ~~1027286~~
Change-Id: I1d500647d6227c9be3ae14d9604ba702e9c29834
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1933452

Reviewed-by: Owen Min <zmin@chromium.org>
Reviewed-by: Xiaocheng Hu <xiaochengh@chromium.org>
Commit-Queue: Owen Min <zmin@chromium.org>
Cr-Commit-Position: refs/heads/master@{#718778}

[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/commands/clipboard_commands.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/commands/replace_selection_command.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/commands/replace_selection_command_test.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/editing_style_utilities.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/editing_style_utilities.h
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/create_markup_options.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/create_markup_options.h
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/serialization.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/serialization.h
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.cc
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.h
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/renderer/core/editing/serializers/styled_markup_serializer.cc
[delete] https://crrev.com/219d095da1dae034bb4de66bfb5bf252a70bd9af/third_party/blink/web_tests/editing/pasteboard/paste-from-excel.html
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/web_tests/editing/pasteboard/paste-head-contents-expected.txt
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/web_tests/editing/pasteboard/paste-head-contents.html
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/web_tests/editing/pasteboard/paste-xss-injection.html
[modify] https://crrev.com/f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b/third_party/blink/web_tests/editing/pasteboard/preserve-underline-color-expected.txt

Comment 23 by xiaoc...@chromium.org on Mon, Nov 25, 2019, 5:28 PM EST Project Member

Michal: Thanks for the other catch!

By default, @import is already disabled in pasting. Even without the patch (comment 15), we can already see @import rules stripped from pasted style elements.

The crash is due to the dummy page trying to create a WebURLLoaderFactory, which current fails due to the empty clients provided. Providing a mock object that doesn't load anything should be enough.

Comment 24 by xiaoc...@chromium.org on Mon, Nov 25, 2019, 5:34 PM EST Project Member

Or setting a flag to the dummy document to prevent imports. It should also work.

Comment 25 by bugdroid on Mon, Nov 25, 2019, 6:55 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/4886f590400a0fd3e4756333e69784c5dd313580

commit 4886f590400a0fd3e4756333e69784c5dd313580
Author: Xiaocheng Hu <xiaochengh@chromium.org>
Date: Mon Nov 25 23:54:46 2019

Reland "Sanitize style elements in clipboard markup"

This reverts commit f6953a5e9d62cde66ea6edd2f4f46d1dcee7940b.

Reason for revert: Manually destroyed the dummy page to ensure no leak

Original change's description:
> Revert "Sanitize style elements in clipboard markup"
>
> This reverts commit d96236b5d2bad68a0cc8f62501ba15c38c8cf96a.
>
> Reason for revert: This may cause "WebKit Linux Leak" failure
> First failure: https://ci.chromium.org/p/chromium/builders/ci/WebKit%20Linux%20Leak/7276
>
> Original change's description:
> > Sanitize style elements in clipboard markup
> >
> > This patch sanitizes clipboard markup before pasting it into document
> > by removing all pasted style elements and serializing them onto
> > elements as inline style. In this way, we stop stylesheets in clipboard
> > markup from being applied to the original elements in the document.
> >
> > This patch follows the same approach as in WebKit [1]:
> > - First create a dummy document to insert the markup
> > - Then computes style and layout in the dummy document
> > - Re-serialize the dummy document as the markup to be inserted. This
> > reuses the code path that we serialize a selection range into
> > clipboard, where we need to serialize element computed style into
> > inline styles so that the element styles are preserved.
> > - Make sure all style elements are removed before inserting markup
> > into document
> >
> > This patch also adds a complete test to ensure that content pasted from
> > Excel is still properly styled, which is the main reason we used to
> > preserve style elements in clipboard markup [2].
> >
> > [1] https://trac.webkit.org/changeset/223440
> > [2] ~~http://crbug.com/121163~~
> >
> > ~~Bug: 1017871~~
> > Change-Id: I3bb5a4ae7530a3fdef5ba251975e004857c06f1e
> > Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1922919
> > Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
> > Reviewed-by: Yoshifumi Inoue <yosin@chromium.org>
> > Reviewed-by: Kent Tamura <tkent@chromium.org>
> > Cr-Commit-Position: refs/heads/master@{#718281}
>
> TBR=yosin@chromium.org,tkent@chromium.org,xiaochengh@chromium.org
>
> # Not skipping CQ checks because original CL landed > 1 day ago.
>
> ~~Bug: 1017871~~, ~~1027186~~
> Change-Id: I1d500647d6227c9be3ae14d9604ba702e9c29834
> Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1933452
> Reviewed-by: Owen Min <zmin@chromium.org>
> Reviewed-by: Xiaocheng Hu <xiaochengh@chromium.org>
> Commit-Queue: Owen Min <zmin@chromium.org>
> Cr-Commit-Position: refs/heads/master@{#718778}

TBR=yosin@chromium.org,tkent@chromium.org,zmin@chromium.org,xiaochengh@chromium.org

Cq-Include-Trybots=luci.chromium.try:layout_test_leak_detection

~~Bug: 1017871~~, ~~1027386~~
Change-Id: Ia56ee941979cad71e2bac06998c7ac417b4731bd
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1934650
Reviewed-by: Xiaocheng Hu <xiaochengh@chromium.org>
Reviewed-by: Kent Tamura <tkent@chromium.org>
Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
Cr-Commit-Position: refs/heads/master@{#718896}

[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/commands/clipboard_commands.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/commands/replace_selection_command.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/commands/replace_selection_command_test.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/editing_style_utilities.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/editing_style_utilities.h
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/create_markup_options.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/create_markup_options.h
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/serialization.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/serialization.h
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.cc
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/styled_markup_accumulator.h
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/renderer/core/editing/serializers/styled_markup_serializer.cc
[add] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/web_tests/editing/pasteboard/paste-from-excel.html
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/web_tests/editing/pasteboard/paste-head-contents-expected.txt
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/web_tests/editing/pasteboard/paste-head-contents.html
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/web_tests/editing/pasteboard/paste-xss-injection.html
[modify] https://crrev.com/4886f590400a0fd3e4756333e69784c5dd313580/third_party/blink/web_tests/editing/pasteboard/preserve-underline-color-expected.txt

Comment 26 by bugdroid on Mon, Nov 25, 2019, 7:07 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/7d16958ff0de76a5420397d1d0448a9d8e68e05d

commit 7d16958ff0de76a5420397d1d0448a9d8e68e05d
Author: Xiaocheng Hu <xiaochengh@chromium.org>
Date: Tue Nov 26 00:06:38 2019

Strip SVGStyleElement in ReplaceSelectionCommand

crrev.com/c/1922919 added a stylesheet sanitizer for clipboard, but left
a loophole for SVGStyleElement. This patch also strips it.

~~Bug: 1017871~~
Change-Id: Icc6c513f79597c191f732cd63a98cc59afe1fc69
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1931412
Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
Reviewed-by: Yoshifumi Inoue <yosin@chromium.org>
Cr-Commit-Position: refs/heads/master@{#718902}

[modify] https://crrev.com/7d16958ff0de76a5420397d1d0448a9d8e68e05d/third_party/blink/renderer/core/editing/commands/replace_selection_command.cc
[add] https://crrev.com/7d16958ff0de76a5420397d1d0448a9d8e68e05d/third_party/blink/web_tests/editing/pasteboard/mathml-sanitizer-bypass.html
[delete] https://crrev.com/69f722944e1875ef429bfea2cdeccef6824a29e7/third_party/blink/web_tests/editing/pasteboard/paste-xss-injection.html

Comment 27 by mic...@bentkowski.info on Tue, Nov 26, 2019, 5:03 AM EST

It seems that the bypass via <svg><style> no longer works (but I'll keep trying to find more ways ;)).

But I can still trigger the crash and it seems to me that @import at-rules are processed.

I'm using https://jsbin.com/mozidoxegi/edit?html,output again and when I try to copy

```
foo<style>@import'data:,*{background:red}'</style>bar
```

Then "foobar" has red background when being pasted.

Also, when copying and pasting

```
foo<style>@import'https://anyting'</style>bar
```

Then I'm getting the "Aw, Snap" error as I mentioned earlier.

I'm testing it on MacOS 10.14.6, on https://cr-rev.appspot.com/719066 downloaded from https://download-chromium.appspot.com/.

Comment 28 by xiaoc...@chromium.org on Tue, Nov 26, 2019, 6:45 PM EST Project Member

Hi Michal,

So far I've landed two patches that (i) adds general style sanitization for pasting, and (ii) removes SVGStyleElement from ReplaceSelectionCommand.

I'm still working on the third patch to ban import rules during sanitization: crrev.com/c/1935429. You may test again after it is landed.

(Hopefully there won't be a fourth one)

Comment 29 by bugdroid on Wed, Nov 27, 2019, 5:17 PM EST Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/a94963cedd74ba312af09970cf8e91a5b89dce9d

commit a94963cedd74ba312af09970cf8e91a5b89dce9d
Author: Xiaocheng Hu <xiaochengh@chromium.org>
Date: Wed Nov 27 22:14:31 2019

Disable CSS @import rules in clipboard markup sanitization

While clipboard markup is allowed to carry style sheets to style the
elements to be pasted (e.g., when copying from Excel), @import rules
should be disabled for security reasons.

This patch disables @import rules when sanitizing the markup in a dummy
document to make sure we don't initiate any stylesheet loading during
the process.

~~Bug: 1017871~~
Change-Id: I484341dc34e2ceea1891a18ac158ed2cc4920c9b

Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/1935429
Commit-Queue: Xiaocheng Hu <xiaochengh@chromium.org>
Reviewed-by: Yoshifumi Inoue <yosin@chromium.org>
Reviewed-by: Rune Lillesveen <futhark@chromium.org>
Reviewed-by: Kent Tamura <tkent@chromium.org>
Cr-Commit-Position: refs/heads/master@{#719779}

[modify] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/renderer/core/css/parser/css_parser_context.cc
[modify] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/renderer/core/css/parser/css_parser_context.h
[modify] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/renderer/core/css/parser/css_parser_impl.cc
[modify] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/renderer/core/dom/document.h
[modify] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/renderer/core/editing/serializers/serialization.cc
[add] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/web_tests/editing/pasteboard/block-stylesheet-import-rules.html
[add] https://crrev.com/a94963cedd74ba312af09970cf8e91a5b89dce9d/third_party/blink/web_tests/editing/resources/all-red.css

Comment 30 by xiaoc...@chromium.org on Thu, Nov 28, 2019, 12:44 AM EST Project Member

Status: Fixed (was: Assigned)

Comment 31 by sheriffbot@chromium.org on Thu, Nov 28, 2019, 10:46 AM EST Project Member

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

Comment 32 by mic...@bentkowski.info on Sun, Dec 1, 2019, 2:19 PM EST

Just for the record: the fix now seems fine to me.

Comment 33 by natashapabrai@google.com on Mon, Dec 2, 2019, 1:09 PM EST Project Member

Labels: reward-topanel

Comment 34 by sheriffbot@chromium.org on Tue, Dec 3, 2019, 11:11 AM EST Project Member

Labels: Merge-Request-79

Requesting merge to beta M79 because latest trunk commit (719779) appears to be after beta branch point (706915).

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

Comment 35 by sheriffbot@chromium.org on Tue, Dec 3, 2019, 11:12 AM EST Project Member

Labels: -Merge-Request-79 Hotlist-Merge-Review Merge-Review-79

This bug requires manual review: We are only 6 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: benmason@(Android), kariahda@(iOS), cindyb@(ChromeOS), govind@(Desktop)

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

Comment 36 by adetaylor@google.com on Tue, Dec 3, 2019, 12:43 PM EST Project Member

Security TPM note: I'd like to merge this into M79, but I regard this as a substantial change which introduces some risk, so if we end up saving this for M80 that's OK with me.

Comment 37 by gov...@chromium.org on Tue, Dec 3, 2019, 12:47 PM EST Project Member

Cc: benmason@chromium.org pbomm...@chromium.org
Labels: -Merge-Review-79 Merge-Rejected-79

Rejecting merge to M79 per comment #36 as we're trying to be super careful with merges due to upcoming holidays.

Comment 38 by natashapabrai@google.com on Thu, Dec 5, 2019, 9:28 AM EST Project Member

Labels: -reward-topanel reward-unpaid reward-10000

*** 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 39 by natashapabrai@google.com on Thu, Dec 5, 2019, 9:34 AM EST Project Member

Thanks for all the help on this report! The Panel decided to reward $10,000 for this report! Nice work!

Comment 40 by natashapabrai@google.com on Thu, Dec 5, 2019, 9:41 AM EST Project Member

Labels: -reward-unpaid reward-inprocess

Comment 41 by mic...@bentkowski.info on Tue, Dec 10, 2019, 1:22 PM EST

Thanks, that's an amazing reward!

Comment 42 by adetaylor@google.com on Sat, Feb 1, 2020, 8:13 PM EST Project Member

Labels: Release-0-M80

Comment 43 by adetaylor@chromium.org on Mon, Feb 3, 2020, 6:47 PM EST Project Member

Labels: CVE-2020-6391 CVE_description-missing

Comment 44 by adetaylor@chromium.org on Mon, Feb 10, 2020, 4:37 PM EST Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 45 by adetaylor@google.com on Wed, Mar 4, 2020, 1:43 PM EST Project Member

Cc: achuith@chromium.org

Comment 46 by sheriffbot on Thu, Mar 5, 2020, 1:58 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 47 by xiaoc...@chromium.org on Wed, Apr 29, 2020, 7:34 PM EDT Project Member

Cc: vogelheim@chromium.org