Starred by 8 users

Owner:	kevers@chromium.org
CC:	flackr@chromium.org adetaylor@chromium.org andruud@chromium.org pbomm...@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>Animation Privacy>Fingerprinting
Modified:	Feb 25, 2022

Hotlist-Merge-Review
reward-5000
Security_Impact-Stable
Security_Severity-Medium
allpublic
reward-inprocess
CVE_description-submitted
M-91
Merge-Rejected-91
Target-91
external_security_report
merge-merged-4430
merge-merged-90
LTS-Merged-90
LTS-Security-90
Release-0-M92
CVE-2021-30582
LTS-Size-Small
LTS-Complexity-Trivial

Issue 1205981: Visited links leak via CSS transitions and the transitionrun event (Windows 10, Linux)

Reported by gliu1...@gmail.com on Wed, May 5, 2021, 2:23 PM EDT

Code

VULNERABILITY DETAILS
By applying a CSS transition to the color of visited links, JavaScript can detect if a transition runs, and thus infer if a link is visited. For the setup, a div contains a bunch of links to test. Then, 'a' and 'div.changer a' are the *same* color, and 'a:visited' is a *different* color. Also a CSS transition to 'a' elements are added.

With the setup, visited links will fire transition events ('transitionrun','transitionstart',etc.) whenever the changer class is toggled, but unvisited links will not. This requires no user interaction and such history sniffing is not visible to a user.

VERSION
Chrome Version: 90.0.4430.93 (64-bit) stable
Operating Systems:
I tested it on both Operating Systems with the same version of Chrome
* Ubuntu 20.04.2 LTS
* Windows 10 Home 20H2: OS build: 19042.867 + Windows Feature Experience Pack 120.2212.551.0

REPRODUCTION CASE
See attached .html file.

CREDIT INFORMATION
Reporter credit: George Liu <https://gliu20.github.io>

silent-exploit.html
1.5 KB View Download

Comment 1 by sheriffbot on Wed, May 5, 2021, 2:26 PM EDT Project Member
Labels: external_security_report

Comment 2 by rsleevi@chromium.org on Wed, May 5, 2021, 3:33 PM EDT Project Member
Summary: Visited links leak via CSS transitions and the transitionrun event (Windows 10, Linux) (was: Security: Visited links leak via CSS transitions and the transitionrun event (Windows 10, Linux))
Status: Assigned (was: Unconfirmed)
Owner: flackr@chromium.org
Labels: Security_Severity-Medium Security_Impact-Stable OS-Android OS-Chrome OS-Fuchsia OS-Linux OS-Mac OS-Windows OS-Lacros Pri-2
Components: Privacy>Fingerprinting Blink>Animation

Thanks for reporting this!

Normally, we don't treat privacy bugs as security bugs, as covered at https://chromium.googlesource.com/chromium/src/+/master/docs/security/faq.md#are-privacy-issues-considered-security-bugs . However, I note our severity guidance does mention reliably inferring browser history as being an exception, as covered at https://chromium.googlesource.com/chromium/src/+/master/docs/security/severity-guidelines.md . I may have mis-triaged this, though, so I'll work to confirm this is still the case :)

flackr@: I noticed you added the transition events in https://source.chromium.org/chromium/chromium/src/+/5b648ebafd7ce9b6780bd0c14b57de2524e7b15c , would you be

a good first person to take a look at this? I've confirmed this repros on macOS, so I'm going ahead and assuming all platforms are affected. Note that this relies on rAF, so I had to tab switch to get rAF to trigger :)

Comment 3 by flackr@chromium.org on Wed, May 5, 2021, 3:53 PM EDT Project Member

Owner: kevers@chromium.org
Cc: flackr@chromium.org

I added transitionrun, transitionstart, and transitioncancel however transitionend has been available for a long time and likely provides the same information (whether a transition ran).

As for how we should be protecting against this, the animations code for interpolating colors has a pair of values for the visited and non-visited color. We should always trigger a transition regardless of whether the used color value changes as long as either the visited or non-visited style changes. In this way the transition running provides no information as it would run for a visited or non-visited link.

Kevin, can you ensure that we always initiate a transition animations regardless of whether we are using the visited or non-visited values?

Comment 4 by gliu1...@gmail.com on Wed, May 5, 2021, 4:02 PM EDT

Thank you so much!
For requiring a tab switch due to rAF, I've tested it and it seems like changing the tick function to
`const tick = () => new Promise((resolve) => setTimeout(resolve,100));` also works, but I don't have a Mac to verify if this is the case. The only purpose of the tick function is to give a pause so that the browser can do any style calculations if necessary to determine if a transition should occur.

I've attached this variant below.

Also, this is my first time reporting a bug, but maybe it is Pri-1 judging from ~~https://crbug.com/381808~~ and ~~https://crbug.com/835500~~ ? I'm not sure either though so some confirmation would be nice.

If there's anything else you need please let me know I'll be happy to help!

silent-exploit-no-rAF.html
1.5 KB View Download

Comment 5 by davidben@chromium.org on Wed, May 5, 2021, 4:06 PM EDT Project Member

Nice find! In the short term, I agree we'll need an animation-specific fix. Though I think this is yet another data point in the sea of reasons why we ought to do https://github.com/w3c/csswg-drafts/issues/3012 or something along those lines.

Comment 6 by sheriffbot on Thu, May 6, 2021, 1:03 PM EDT Project Member

Labels: M-91 Target-91

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

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

Comment 7 by sheriffbot on Thu, May 6, 2021, 1:39 PM EDT Project Member

Labels: -Pri-2 Pri-1

Setting Pri-1 to match security severity Medium. If this is incorrect, please reset the priority. Sheriffbot won't make this change again.

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

Comment 8 by kevers@chromium.org on Tue, May 11, 2021, 10:20 AM EDT Project Member

Cc: andruud@chromium.org

Comment 9 by andruud@chromium.org on Tue, May 11, 2021, 10:45 AM EDT Project Member

Following the existing patterns of "visited stuff has an effect paint time, but no effect through APIs", the correct behavior seems to be:

- Interpolate visited and unvisited colors separately. This was impractical earlier, but now that visited colors are their own properties, it should be feasible.
- Interpolations on the visited colors produce paint-time effects, but do not trigger events.

In other words, if _only_ visited colors changed, there should be no events at all.

Comment 10 by kevers@chromium.org on Tue, May 11, 2021, 11:19 AM EDT Project Member

Investigating the issue... Triggering a transition on a style change regardless of whether the link was visited is not trivial to implement since the visited color is not updated in the computed style for links that have not been visited (not relevant for painting). An alternate solution would be to suppress reporting of transition events on properties for an element inside of a link if the property is potentially affected by the visited state. This fix simply requires resetting the event delegate for transitions that should not be visible in JS. If we adopt this strategy, then we should also ensure that getAnimations does not report these transitions.

Comment 11 by flackr@chromium.org on Tue, May 11, 2021, 11:25 AM EDT Project Member

We should ensure that we have consistency in wpt since such changes to animations / dispatched events will be developer visible.

Note that for animations, we should still fire events even if visited color is the only property being animated. This is because even animations with no properties still are spec'd to run and produce events. The firing of these events at the appropriate times also gives away nothing of the output state.

For transitions, I can see an argument that since transitions are only created if the property changes, then we could treat a visited-only property change as not having any transition from a developer-visible standpoint. This seems like something that should be explicitly called out in the spec, though we should also check what Firefox and Safari do to see if they have chosen the same or a different position.

Comment 12 by gliu1...@gmail.com on Tue, May 11, 2021, 1:12 PM EDT

I'm not too familiar with C++, but from my limited knowledge, I'm pretty sure the Firefox source code appears to be using a function `GetVisitedDependentColor(&nsStyleText::mColor);` [1] which returns the normal link color regardless of visited state.

The code that handles transition events uses that function so it never knows the true color of a link. I remember finding it in their source code, but I can't seem to find it right now. I'll update this if I can remember which file I was looking at.

[1] https://searchfox.org/mozilla-central/search?q=GetVisitedDependentColor&path=&case=false&regexp=false

Comment 13 by andruud@chromium.org on Tue, May 11, 2021, 3:35 PM EDT Project Member

> Triggering a transition on a style change regardless of whether the link was visited is not trivial to implement since the visited color is not updated in the computed style for links that have not been visited (not relevant for painting).

(I should have mentioned this earlier during our chat): We don't store it on ComputedStyle for unvisited links *currently*, but I think it's only a performance optimization, and not necessarily an important one either (not sure). If we really want the "transition when either change" behavior, we can look into always storing it.

> Note that for animations, we should still fire events even if visited color is the only property being animated.

Hmm, is it possible to create a visited-dependent *animation* though?

> explicitly called out in the spec

Yes, that would be nice. AFAIK almost none of the current :visited behavior is really specified in detail. There's a high-level carte blanche in

https://drafts.csswg.org/selectors-4/#link. And there are some nice MDN pages on the subject that serve as a de-facto spec for the style team. :-)

Overall it seems like the behavior we should aim for is to appear as if all links are permanently unvisited, as seen from JS. This is why I think the two colors should ideally be interpolated separately, where the visited interpolation is "silent". It may be too much work / not worth it, though, if there are simpler ways to plug the leak.

> which returns the normal link color regardless of visited state.

Are you saying Firefox transitions (visually) as if the link is unvisited, even if it isn't? (I would just check myself, but getting late in my timezone ...)

Comment 14 by gliu1...@gmail.com on Tue, May 11, 2021, 4:30 PM EDT

> Are you saying Firefox transitions (visually) as if the link is unvisited, even if it isn't? (I would just check myself, but getting late in my timezone ...)
Yep, that is correct. Except that once it's done transitioning, it seems to do another paint to correct for the visited/un-visited discrepancies. I've attached a video of this in case it helps, and I've compared it to Chrome in the attached screencapture. I also attached the file I used to create the video in case you want to try it on your machine.

The red color is for visited. The blue color is for on hover. The black color is for not visited links.

Comment 15 by gliu1...@gmail.com on Tue, May 11, 2021, 4:31 PM EDT

screencapture.webm
9.3 MB View Download

firefox-transition.html
312 bytes View Download

Comment 16 by andruud@chromium.org on Wed, May 12, 2021, 3:28 AM EDT Project Member

gliu10000@: That's interesting. It's one way of plugging the leak I guess. :-)

Looks like Webkit (tested with Epihpany TP) has the same visual behavior as us: they transition from the visited color.

Comment 17 by Git Watcher on Fri, May 14, 2021, 10:25 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/ce84f05208f522fb8e0bf5a7e374abb5cd7435f1

commit ce84f05208f522fb8e0bf5a7e374abb5cd7435f1
Author: Kevin Ellis <kevers@chromium.org>
Date: Fri May 14 14:24:39 2021

Fix leaking of visited links via CSS transitions.

~~Bug: 1205081~~
Change-Id: I0c39e32dee8c71bbf4ab3792f3da29bd0b765575
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2889431
Reviewed-by: Anders Hartvoll Ruud <andruud@chromium.org>
Commit-Queue: Kevin Ellis <kevers@chromium.org>
Cr-Commit-Position: refs/heads/master@{#882965}

[modify] https://crrev.com/ce84f05208f522fb8e0bf5a7e374abb5cd7435f1/third_party/blink/renderer/core/animation/css/css_animations.cc
[modify] https://crrev.com/ce84f05208f522fb8e0bf5a7e374abb5cd7435f1/third_party/blink/renderer/core/css/element_rule_collector.cc
[modify] https://crrev.com/ce84f05208f522fb8e0bf5a7e374abb5cd7435f1/third_party/blink/renderer/core/css/element_rule_collector_test.cc
[add] https://crrev.com/ce84f05208f522fb8e0bf5a7e374abb5cd7435f1/third_party/blink/web_tests/animations/transition-visited.html

Comment 18 by kevers@chromium.org on Fri, May 14, 2021, 10:36 AM EDT Project Member

Status: Fixed (was: Assigned)

Issue resolved. If either the visited or unvisited style changes for a transition property we create a transition. Visually, we apply the corresponding style, but it either case we report the unvisited style to JavaScript (via document.getAnimations() --> animation.effect.getKeyframes()). Using the sample code from the original comment, a transitionrun event will be fired for each anchor element regardless of whether visited. That's Rob for the suggested approach to address the issue.

Comment 19 by sheriffbot on Fri, May 14, 2021, 12:42 PM EDT Project Member

Labels: reward-topanel

Comment 20 by sheriffbot on Fri, May 14, 2021, 2:02 PM EDT Project Member

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

Comment 21 by gliu1...@gmail.com on Fri, May 14, 2021, 2:14 PM EDT

Thank you so much and for the quick fix!

Comment 22 by sheriffbot on Fri, May 14, 2021, 2:27 PM EDT Project Member

Labels: Merge-Request-91

Requesting merge to beta M91 because latest trunk commit (882965) appears to be after beta branch point (965).

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

Comment 23 by sheriffbot on Fri, May 14, 2021, 2:29 PM EDT Project Member

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

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

1. Does your merge fit within the Merge Decision Guidelines?

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

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

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

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

Comment 24 by kevers@chromium.org on Fri, May 14, 2021, 3:18 PM EDT Project Member

Re comment #23:
1. Yes, this is a privacy fix.
2. https://chromium-review.googlesource.com/c/chromium/src/+/2889431
3. The change has landed on ToT
4. Unclear, M91 may be sufficient
5. Fixes a privacy exploit
6. Not a new feature
7 NA

Comment 25 by pbommana@google.com on Sat, May 15, 2021, 1:13 AM EDT Project Member

Cc: adetaylor@chromium.org pbomm...@chromium.org

+Adetaylor(Security TPM) for Merge-decision

Note : The change has just landed on 92.0.4507.0 which went out couple of hours back.

Comment 26 by adetaylor@chromium.org on Mon, May 17, 2021, 10:10 AM EDT Project Member

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

I'm inclined not to merge this to M91 at this late stage. As a medium severity bug, I think we can let this flow through beta to find out if any websites are disrupted by this change.

Comment 27 by amyressler@google.com on Thu, May 20, 2021, 1:08 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-5000

*** 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 28 by amyressler@chromium.org on Thu, May 20, 2021, 5:35 PM EDT Project Member

Congratulations, George! The VRP Panel has decided to award you $5000 for this report. Someone from our finance team will be in touch soon to arrange payment. Nice work!

Comment 29 by gliu1...@gmail.com on Thu, May 20, 2021, 6:36 PM EDT

Thank you so much for the reward! I really appreciate it! :)

Comment 30 by amyressler@google.com on Fri, May 21, 2021, 5:29 PM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 31 by amyressler@chromium.org on Mon, Jul 19, 2021, 3:16 PM EDT Project Member

Labels: Release-0-M92

Comment 32 by amyressler@google.com on Mon, Jul 19, 2021, 7:17 PM EDT Project Member

Labels: CVE-2021-30582 CVE_description-missing

Comment 33 by rzanoni@google.com on Mon, Aug 2, 2021, 8:40 AM EDT Project Member

Labels: LTS-Security-90 LTS-Merge-Request-90 LTS-Size-Small LTS-Complexity-Trivial

Comment 34 by amyressler@google.com on Tue, Aug 3, 2021, 3:42 PM EDT Project Member

Labels: -CVE_description-missing CVE_description-submitted

Comment 35 by gianluca@google.com on Thu, Aug 5, 2021, 6:23 AM EDT Project Member

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

Comment 36 by Git Watcher on Thu, Aug 5, 2021, 9:26 AM EDT Project Member

Labels: merge-merged-4430 merge-merged-90

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/17b41519ac9ed6eddb2771522dfdb604e2c36d0b

commit 17b41519ac9ed6eddb2771522dfdb604e2c36d0b
Author: Kevin Ellis <kevers@chromium.org>
Date: Thu Aug 05 13:25:28 2021

[M90-LTS] Fix leaking of visited links via CSS transitions.

(cherry picked from commit ce84f05208f522fb8e0bf5a7e374abb5cd7435f1)

~~Bug: 1205981~~
Change-Id: I0c39e32dee8c71bbf4ab3792f3da29bd0b765575
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2889431
Commit-Queue: Kevin Ellis <kevers@chromium.org>
Cr-Original-Commit-Position: refs/heads/master@{#882965}
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3066245

Reviewed-by: Achuith Bhandarkar <achuith@chromium.org>
Reviewed-by: Jana Grill <janagrill@google.com>
Owners-Override: Jana Grill <janagrill@google.com>
Commit-Queue: Roger Felipe Zanoni da Silva <rzanoni@google.com>
Cr-Commit-Position: refs/branch-heads/4430@{#1553}
Cr-Branched-From: e5ce7dc4f7518237b3d9bb93cccca35d25216cbe-refs/heads/master@{#857950}

[modify] https://crrev.com/17b41519ac9ed6eddb2771522dfdb604e2c36d0b/third_party/blink/renderer/core/animation/css/css_animations.cc
[modify] https://crrev.com/17b41519ac9ed6eddb2771522dfdb604e2c36d0b/third_party/blink/renderer/core/css/element_rule_collector.cc
[modify] https://crrev.com/17b41519ac9ed6eddb2771522dfdb604e2c36d0b/third_party/blink/renderer/core/css/element_rule_collector_test.cc
[add] https://crrev.com/17b41519ac9ed6eddb2771522dfdb604e2c36d0b/third_party/blink/web_tests/animations/transition-visited.html

Comment 37 by rzanoni@google.com on Thu, Aug 5, 2021, 9:42 AM EDT Project Member

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

Comment 38 by sheriffbot on Fri, Aug 20, 2021, 1:30 PM EDT Project Member

Labels: -Restrict-View-SecurityNotify allpublic

This bug has been closed for more than 14 weeks. Removing security view restrictions.

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

Comment 39 by ndevtk@protonmail.com on Wed, Oct 27, 2021, 1:24 PM EDT

Seems duplicate of https://bugs.chromium.org/p/chromium/issues/detail?id=713521 :/

Comment 40 by ericlaw@microsoft.com on Wed, Oct 27, 2021, 8:27 PM EDT Project Member

RE #39: That's the correct umbrella bug for this ~class~ of issue, yes, as noted in #c5. This bug is a concrete ~instance~ of that class of bug.

Comment 41 by ndevtk@protonmail.com on Fri, Feb 25, 2022, 7:50 PM EST

"This bug is a concrete ~instance~ of that class of bug." Im not sure what you mean this bug can still be exploited easily without user interaction the browser history should never be exposed to a different website.