☆ Starred by 3 users

Owner:	obru...@igalia.com
CC:	dom@chromium.org xiaoc...@chromium.org yosin@chromium.org shivanisha@chromium.org haraken@chromium.org arthu...@chromium.org chrishtr@chromium.org mas...@chromium.org
Status:	Fixed *(Closed)*
Components:	Blink>HTML>IFrame Blink>HTML Privacy Blink>Editing>Selection
Modified:	Jul 29, 2022
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Android, Windows, Chrome, Mac, Lacros
Pri:	2
Type:	Bug-Security

reward-0
Security_Severity-Low
allpublic
CVE_description-submitted
external_security_report
FoundIn-97
Security_Impact-Extended
Release-0-M101
CVE-2022-1501

Issue 1293191: Propagating inertness into nested browsing contexts leaks information, privacy concern?

Reported by obru...@igalia.com on Tue, Feb 1, 2022, 6:38 PM EST   Project Member

Code

Chrome Version: 97.0.4692.99
OS: Ubuntu 20.04

What steps will reproduce the problem?
(1) Load this page:

```
<!DOCTYPE html>
<button onclick="dialog.showModal()">Open modal</button>
<dialog id="dialog" style="text-align: center">
  I'm a modal dialog<br>
  <button onclick="dialog.close()">Close</button>
</dialog>
<br>
<iframe sandbox="allow-scripts" srcdoc="
  <style>:focus { border: solid black }</style>
  I'm a cross-origin iframe, but I know that the modal dialog in my embedder is:
  <div id='result'></div>
  <script>
  setInterval(() => {
    getSelection().selectAllChildren(document.documentElement);
    result.textContent = getSelection().toString().trim() ? 'CLOSED' : 'OPEN';
    getSelection().empty();
  }, 60);
  </script>
"></iframe>
```

(2) Note the cross-origin iframe knows that the modal dialog is CLOSED.
(3) Click "Open modal" button
(4) Note the cross-origin iframe knows that the modal dialog is OPEN.
(5) Click "Close" button
(6) Note the cross-origin iframe knows that the modal dialog is CLOSED.

Note this behavior is correct according to the spec:
https://html.spec.whatwg.org/multipage/interaction.html#inert

> While a browsing context container is marked as inert,
> its nested browsing context's active document,
> and all nodes in that Document, must be marked as inert.

But it seems strange to leak this information into (possibly evil) cross-origin iframes...
For example, if they know the embedder has a modal dialog where the user is supposed to type something, they can measure how long the dialog is open as an estimation of how fast the user can type in a keyboard, and use this for tracking or something?

I don't think it's a huge deal, but it can be a privacy concern, so I'm filing this as Bug-Security, so that the privacy experts can analyze if it's fine or not.

Note the behavior is present at least since version 46.
Gecko and WebKit are not affected (they don't follow the HTML spec).

Comment 1 by sheriffbot on Tue, Feb 1, 2022, 6:41 PM EST

Labels: external_security_report

Comment 2 by mas...@chromium.org on Wed, Feb 2, 2022, 11:50 AM EST

Cc: domfarolino@google.com haraken@chromium.org chrishtr@chromium.org

+ some fenced frame folks: this might represent an information leakage path into at least the shadow DOM implementation of fenced frames.

Comment 3 by dom@chromium.org on Wed, Feb 2, 2022, 11:59 AM EST

Cc: -domfarolino@google.com dom@chromium.org

Please see the #inert chat room in slack for our question about this relating to fenced frames

Comment 4 by mas...@chromium.org on Wed, Feb 2, 2022, 12:38 PM EST

Ahh, thanks for the pointer.

Comment 5 by xinghuilu@chromium.org on Wed, Feb 2, 2022, 1:43 PM EST

Status: Assigned (was: Untriaged)
Owner: arthu...@chromium.org
Cc: xiaoc...@chromium.org yosin@chromium.org
Labels: Security_Severity-Low FoundIn-97 OS-Android OS-Chrome OS-Linux OS-Mac OS-Windows OS-Lacros
Components: Blink>Editing>Selection Blink>HTML>IFrame

Thanks for the report. I'm able to reproduce. It only works with the "allow-scripts" tag, so it's not iframe sandbox escape. Marking severity as low.

+arthursonzogni@, could you weigh in on whether this is a concern from iframe security point of view? Adding xiaochengh@ and yosin@ to see how this issue can be addressed from the editing API side. Thanks!

Comment 6 by sheriffbot on Wed, Feb 2, 2022, 1:43 PM EST

Labels: Security_Impact-Extended

Comment 7 by obru...@igalia.com on Wed, Feb 2, 2022, 2:07 PM EST Project Member

Be aware that getSelection() is not the only way to observe inertness. It can also be detected e.g. with focusability.
Note the allow="focus-without-user-activation *" may be needed to bypass BlockingFocusWithoutUserActivation.
It's an experimental feature, so typically allow="focus-without-user-activation *" isn't actually needed.

```
<!DOCTYPE html>
<button onclick="dialog.showModal()">Open modal</button>
<dialog id="dialog" style="text-align: center">

  I'm a modal dialog<br>
  <button onclick="dialog.close()">Close</button>
</dialog>
```

```
  </dialog>
  <br>
  <iframe sandbox="allow-scripts" allow="focus-without-user-activation *" srcdoc="
    I'm a cross-origin iframe, but I know that the modal dialog in my embedder is:
    <div id='result' tabindex='-1'></div>
    <script>
    setInterval(() => {
      result.blur();
      result.focus();
      result.textContent = document.activeElement === result ? 'CLOSED' : 'OPEN';
      result.blur();
    }, 60);
    </script>
  "></iframe>
```

So I don't think this should be addressed from the editing API side.
If this is really a problem, I would just stop propagating inertness into nested browsing contexts.
This would be a nice code cleanup (https://crrev.com/c/3302103), and would align Blink with Gecko and WebKit.

Comment 8 by arthu...@chromium.org on Thu, Feb 3, 2022, 12:37 PM EST

Owner: obru...@igalia.com
Cc: arthu...@chromium.org

Yes, I agree this is a form of cross-site leak. Do you know if this works, when the iframe's document is loaded from a different process (by using a real cross-origin URL)?

This can probably be used as "communication channel" in between two documents that are willing to cooperate and exfiltrate information about the user through the FencedFrame.

obrufau@, I don't know much about those API: Selection / Modal dialog. Do you know who would be a good owner of this bug?

Comment 9 by sheriffbot on Thu, Feb 3, 2022, 1:23 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 10 by obru...@igalia.com on Thu, Feb 3, 2022, 2:11 PM EST Project Member

> Do you know if this works, when the iframe's document is loaded from a different process (by using a real cross-origin URL)

Yes, I have tried with localhost and 127.0.0.1 which should be cross-origin, and it still works.
Note the code for propagating inertness is in cross_process_frame_connector.h, remote_frame.h, etc.

> This can probably be used as "communication channel" in between two documents that are willing to cooperate and exfiltrate information about the user through the FencedFrame.

I have tried with fenced frames and they are affected too. So the embedder can pass information to the fenced frame even without postMessage, resizing, etc. I don't think the fenced frame can use this to pass information to the embedder, though.

> I don't know much about those API: Selection / Modal dialog. Do you know who would be a good owner of this bug?

> I don't know much about those API: Selection / Modal dialog. Do you know who would be a good owner of this bug?

It's not just selection, inertness can also be detected with focusability, document.elementFromPoint(), etc. And not shipped yet, but it will be possible to trigger inertness without modal dialogs (nor fullscreen elements), via the 'inert' attribute.
So I think the problem is actually the propagation of inertness. Rather than trying to add workarounds for all things that may be affected by inertness, I would just stop propagating inertness into frames. If this approach seems reasonable, I can do it.
In fact I already wrote the patch in the past, to see which tests would fail: https://crrev.com/c/3302103

Comment 11 by arthu...@chromium.org on Thu, Feb 3, 2022, 4:34 PM EST

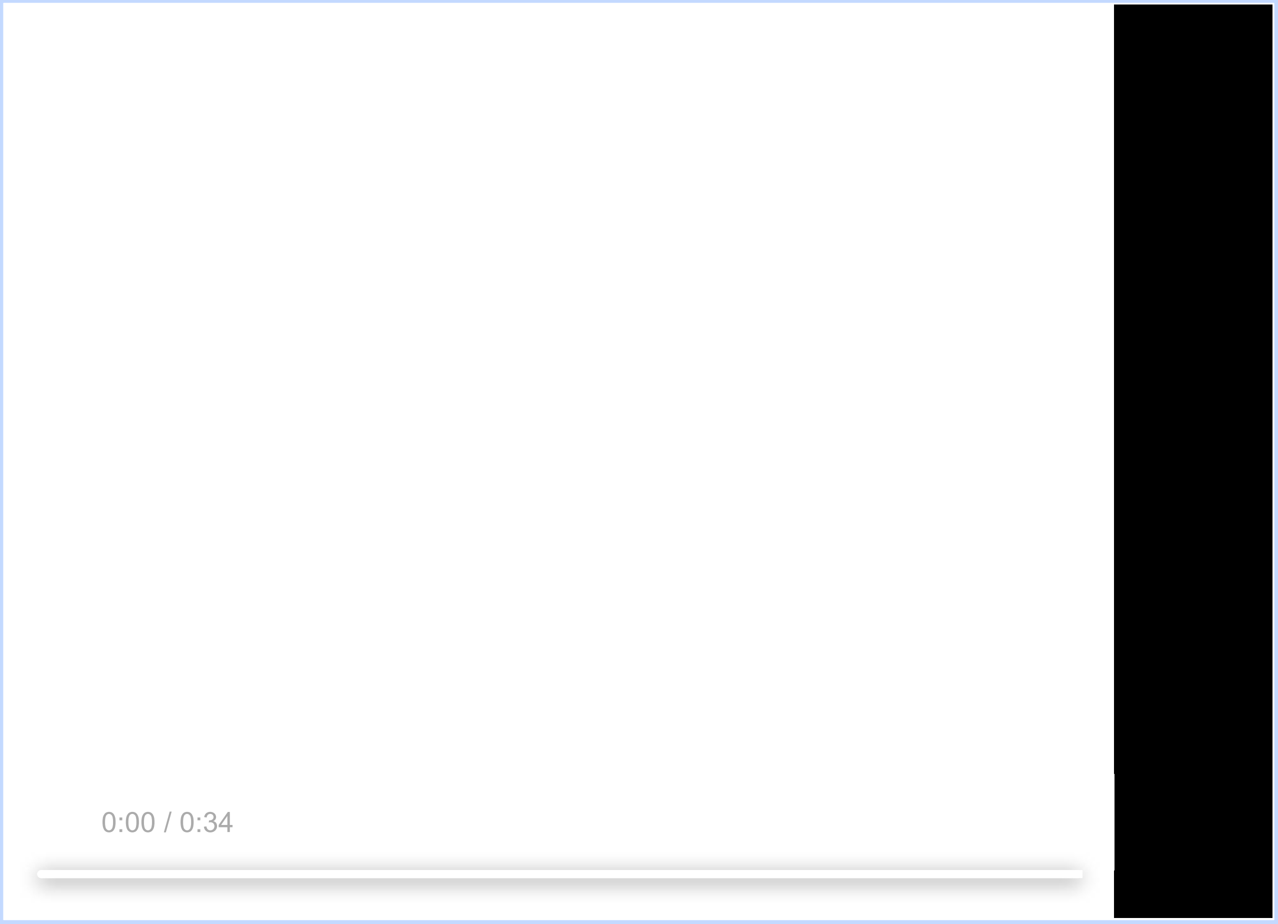
Cc: shivanisha@chromium.org

Thanks!

This is interesting!
This would need to be fixed, because this is XS leaks, and especially if we want to ship fencedframe.

Mostly for fun, I made a demo where you can transmit arbitrary data through a fencedframe using your idea:
https://fenced-frame-dialog-selection-pipe-gigantic-calcium.glitch.me/

test-2022-02-03_22.31.05.mp4
171 KB View Download



Comment 12 by obru...@igalia.com on Thu, Feb 3, 2022, 5:01 PM EST Project Member

Is it fine to mention "cross-site leak" in the CL description, and if I open an issue to the HTML spec to remove inert propagation?

Comment 13 by arthu...@chromium.org on Fri, Feb 4, 2022, 3:55 AM EST

> Is it fine to mention "cross-site leak" in the CL description, and if I open an issue to the HTML spec to remove inert

propagation?

;-)
I have the same kind of issue when submitting security fix in Chrome. I don't really have clear recommendations for myself.
I guess it is fine, because you won't be understood otherwise. You can also probably focus on adopting Firefox behavior.
Also, I don't believe this is very severe and give an important kind of information to the iframe.

Comment 14 by Git Watcher on Fri, Feb 18, 2022, 1:08 PM EST

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src/+/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50

commit a486ffcc6d5ecf0d9827d6468ece0eae4306eb50
Author: Oriol Brufau <obrufau@igalia.com>
Date: Fri Feb 18 18:05:48 2022

[inert] Stop propagating inertness into nested browsing contexts

Doing so was a cross-site leak. This change is against the HTML spec, but it aligns Blink with Gecko. WebKit is also not propagating inertness for the selection API, but it does for focusability.

I already changed the spec https://github.com/whatwg/html/issues/7605
and updated WPT in https://github.com/web-platform-tests/wpt/pull/32817.

Note this only affects the webexposed behavior. The accessibility tree still considers the contents of an inert frame to be inert, as tested by All/DumpAccessibilityTreeTestWithIgnoredNodes.AccessibilityModalDialogAndIframes/*

Therefore Frame::is_inert_ and related flags are kept for accessibility, but they will no longer effect ComputedStyle::IsInert().

Also note that even if the contents in the nested browsing contexts are not marked as inert, if the browsing context container is inert, they won't respond to mouse interactions, and they won't be reached by sequential navigation.

~~Bug: 1293191~~

TEST=All/DumpAccessibilityTreeTestWithIgnoredNodes.InertAttribute/*
TEST=All/SitePerProcessBrowserTest.CrossProcessInertSubframe/*
TEST=All/SitePerProcessBrowserTest.CrossProcessIsInertPropagation/*
TEST=third_party/blink/web_tests/external/wpt/html/semantics/interactive-elements/the-dialog-element/inert-focus-in-frames.html
TEST=third_party/blink/web_tests/external/wpt/html/semantics/interactive-elements/the-dialog-element/inertness-with-modal-dialogs-and-iframes.html
TEST=third_party/blink/web_tests/fast/dom/inert/inert-focus-in-frames.html

AX-Relnotes: n/a.

Change-Id: I70820d2aeca98e1c4036bd3f8c41ef0129a97a63

Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/3302103
Reviewed-by: Aaron Leventhal <aleventhal@chromium.org>
Reviewed-by: Mason Freed <masonf@chromium.org>

Reviewed-by: Mason Freed <masonf@chromium.org>
Reviewed-by: Arthur Sonzogni <arthursonzogni@chromium.org>
Commit-Queue: Oriol Brufau <obrufau@igalia.com>
Cr-Commit-Position: refs/heads/main@{#973015}

[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/content/test/data/accessibility/html/inert-attribute-expected-blink.txt
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/core/frame/frame.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/core/css/resolver/style_resolver.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/web_tests/external/wpt/html/semantics/interactive-elements/the-dialog-element/inert-focus-in-frames.html
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/modules/accessibility/ax_object.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/core/css/resolver/style_resolver_test.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/web_tests/fast/dom/inert/inert-focus-in-frames.html
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/web_tests/external/wpt/html/semantics/interactive-elements/the-dialog-element/inertness-with-modal-dialogs-and-iframes.html
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/content/browser/site_per_process_browsertest.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/core/layout/layout_embedded_content.cc
[modify] https://crrev.com/a486ffcc6d5ecf0d9827d6468ece0eae4306eb50/third_party/blink/renderer/core/frame/local_frame.cc

Comment 15 by obru...@igalia.com on Fri, Feb 18, 2022, 3:19 PM EST Project Member

Status: Fixed (was: Assigned)

Should be fixed now. Only the accessibility tree will continue considering that contents of an inert frame are inert, and this shouldn't be webexposed.

Comment 16 by sheriffbot on Sat, Feb 19, 2022, 12:41 PM EST

Labels: reward-topanel

Comment 17 by sheriffbot on Sat, Feb 19, 2022, 1:39 PM EST

Labels: Restrict-View-SecurityNotify

Comment 18 by amyressler@chromium.org on Thu, Mar 31, 2022, 5:12 PM EDT

Labels: -reward-topanel reward-0

Thanks for this report. As this issue was reported by a contributor/embedder, it is unfortunately ineligible for a VRP reward.

Comment 19 Deleted

Comment 20 by amyressler@chromium.org on Mon, Apr 25, 2022, 8:45 PM EDT

Labels: -Release-0-M100 Release-0-M101

:-|

Comment 21 by amyressler@google.com on Tue, Apr 26, 2022, 4:33 PM EDT

Labels: CVE-2022-1501 CVE_description-missing

Comment 22 by sheriffbot on Sun, May 29, 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 23 by amyressler@google.com on Tue, Jul 26, 2022, 5:37 PM EDT

Labels: CVE_description-submitted -CVE_description-missing

Comment 24 by amyressler@chromium.org on Fri, Jul 29, 2022, 5:26 PM EDT

Labels: -CVE_description-missing --CVE_description-missing