


☆ Starred by 3 users

Owner:


rousian@chromium.org

CC:

a...@google.com


 mkwst@chromium.org

dominickn@chromium.org


 danyao@chromium.org

tedc...@chromium.org

lukasza@chromium.org

 jerryzz@google.com

wfh@chromium.org

 creis@chromium.org

ajgo@chromium.org

Status:

Fixed (Closed)

Components:

Blink>Payments

Internals>Sandbox>SiteIsolation

Blink>SecurityFeature

Blink>SecurityFeature>FetchMetadata

Modified:

May 20, 2020

Backlog-Rank:

----

Editors:

----

EstimatedDays:

----

NextAction:

----

OS:

Linux, Android, Windows, Chrome, Mac

Pri:

1

Type:

Bug-Security

ReleaseBlock-NA

Security\_Impact-Stable

Security\_Severity-Medium

allpublic

CVE\_description-submitted

Target-77

Target-78

Target-79

FoundIn-76


M-79


Release-0-M83

CVE-2020-6483

BlockedOn:

Issue 980641

 ~~Issue 971320~~

 View details

### Issue 966507: Possible Sec-Fetch-Site bypass via PaymentRequest

Reported by jun.k...@microsoft.com on Thu, May 23, 2019, 1:55 PM EDT Project Member

 Code

#### VULNERABILITY DETAILS

A payment method identifier in Payment Handler API is a URL, and browser will send a request to that URL which is a renderer specified URL but request will be browser initiated. See following document for more details.

<https://docs.google.com/document/d/1wM9b3szNH4-w0tpefjLYSGNtyLr31Q4ARNTB52bj0/edit>

Request to this URL has 'sec-fetch-site: none' which is weird because this request is made with JavaScript API from renderer.

#### VERSION

Chrome Version: 76.0.3800.0 + dev

Operating System: Windows 10

#### REPRODUCTION CASE

1. Open attached file and request will be sent to [google.com](https://www.google.com) (This will not appear in DevTools)

**notpop.html**

405 bytes [View](#) [Download](#)

Comment 1 Deleted

Comment 2 by jun.k...@microsoft.com on Thu, May 23, 2019, 2:10 PM EDT Project Member

I have 2 more bugs that bypasses sec-fetch-site, but those are same as following SameSite cookie bypasses.

<https://bugs.chromium.org/p/chromium/issues/detail?id=831731>

<https://bugs.chromium.org/p/chromium/issues/detail?id=831725>

Should I file those bugs for sec-fetch-site too?

Comment 3 by lukasza@chromium.org on Thu, May 23, 2019, 6:06 PM EDT

Cc: mkwst@chromium.org a...@google.com lukasza@chromium.org

Labels: FoundIn-76

Blocking: 843478 786673

Components: Blink>SecurityFeature Internals>Sandbox>SiteIsolation Blink>Payments

Thanks for the report and for finding this issue!

Comment 4 by lukasza@chromium.org on Thu, May 23, 2019, 6:07 PM EDT

RE: #c2: I think there is no need to open separate, Sec-Fetch-Site-flavoured bugs for issue 831731 and ~~issue 834725~~ - I am guessing that the same fix (ensuring request\_initiator is propagated in a trustworthy way) should help with both 1) SameSite cookies and 2) Sec-Fetch-Site.

Comment 5 by lukasza@chromium.org on Thu, May 23, 2019, 6:08 PM EDT

Labels: Security\_Impact-Head

Sec-Fetch-Site was enabled by default in M76.

Comment 6 by [lukasza@chromium.org](#) on Thu, May 23, 2019, 6:19 PM EDT

Owner: [rouslan@chromium.org](#)

[rouslan@](#), could you PTAL?

I wonder if `PaymentManifestDownloader::InitiateDownload` should populate `resource_request->request_initiator`. The initiator should come from a trustworthy source (e.g. come from a browser process, rather than the renderer process; or get validated via `CanAccessDataForOrigin`).

I looked at other fields of `network::ResourceRequest` and think most should be fine. I had some doubts about also mimicking how the renderer would populate `[top_frame_origin]` (to use the correct double-keyed cache), but I am not yet sure if leaving this field unset can enable any specific attacks.

Comment 7 by [lukasza@chromium.org](#) on Thu, May 23, 2019, 6:21 PM EDT

Summary: Possible Sec-Fetch-Site bypass via `PaymentRequest` (was: Possible sec-fetch-site bypass)

Comment 8 by [rouslan@chromium.org](#) on Thu, May 23, 2019, 6:25 PM EDT

[lukasza@](#): Do you have more information about Sec-Fetch-Site or examples in code of how other places do this?

Does <https://w3c.github.io/payment-method-manifest/#fetch-pmm> need to be modified to mention this or is the implementation in Chrome not matching the spec?

Comment 9 by [jun.k...@microsoft.com](#) on Thu, May 23, 2019, 6:33 PM EDT Project Member

1 more bug that bypasses sec-fetch-site:

<https://bugs.chromium.org/p/chromium/issues/detail?id=854424>

I don't have Android phone but following should also bypass sec-fetch-site:

<https://bugs.chromium.org/p/chromium/issues/detail?id=831761>

Comment 10 by [lukasza@chromium.org](#) on Thu, May 23, 2019, 6:34 PM EDT

RE: [#c8](#): [rouslan@](#):

RE: Do you have more information about Sec-Fetch-Site or examples in code of how other places do this?

There is <https://mikewest.github.io/sec-metadata/#sec-fetch-site-header> and <https://www.chromestatus.com/feature/5155867204780032>.

RE: Does <https://w3c.github.io/payment-method-manifest/#fetch-pmm> need to be modified to mention this or is the implementation in Chrome not matching the spec?

I am not sure. I think having a proper initiator [1] is kind of implied - I am not sure if this needs to be explicitly mentioned in <https://w3c.github.io/payment-method-manifest/#fetch-pmm>.

[1] <https://fetch.spec.whatwg.org/#concept-request-initiator>

Comment 11 by [rouslan@chromium.org](#) on Fri, May 24, 2019, 10:30 AM EDT

[@lukasza](#): Three questions regarding Sec-Fetch-Site.

Question #1. Should the requestor be the top-level context (e.g., <https://mom-and-pop-shop.com>) or the actual iframe that made the request (which can be cross-origin, e.g., <https://google.com/pay>).

Question #2. Should the requester of the payment method manifest (e.g., [https://pay.google.com/gp/p/payment\\_method\\_manifest.json](https://pay.google.com/gp/p/payment_method_manifest.json)) be the payment method identifier (e.g., <https://pay.google.com/about>) or the merchant (<https://mom-and-pop-shop.com>).

Question #3. Should the requester of the web app manifest (e.g., [https://pay.google.com/gp/p/web\\_manifest.json](https://pay.google.com/gp/p/web_manifest.json)) pointed to from the payment method manifest (e.g., [https://pay.google.com/gp/p/payment\\_method\\_manifest.json](https://pay.google.com/gp/p/payment_method_manifest.json)) be the origin of the payment method manifest or the merchant again (<https://mom-and-pop-shop.com>).

-----  
For example, let's take the case of <https://mom-and-pop-shop.com> embedding an iframe from <https://google.com/pay>, which causes Chrome to make the following requests:

Step #1: HEAD <https://google.com/pay> -> 303 redirect to HEAD <https://pay.google.com/about>  
<https://pay.google.com/about> has a link header to [https://pay.google.com/gp/p/payment\\_method\\_manifest.json](https://pay.google.com/gp/p/payment_method_manifest.json)

Step #2: GET [https://pay.google.com/gp/p/payment\\_method\\_manifest.json](https://pay.google.com/gp/p/payment_method_manifest.json)  
[https://pay.google.com/gp/p/payment\\_method\\_manifest.json](https://pay.google.com/gp/p/payment_method_manifest.json) is a JSON file with "default\_applications": [[https://pay.google.com/gp/p/web\\_manifest.json](https://pay.google.com/gp/p/web_manifest.json)].

Step #3: GET [https://pay.google.com/gp/p/web\\_manifest.json](https://pay.google.com/gp/p/web_manifest.json)

-----  
Comment 12 by [sheriffbot@chromium.org](#) on Fri, May 24, 2019, 11:07 AM EDT

Status: Assigned (was: Unconfirmed)

Comment 13 by [lukasza@chromium.org](#) on Fri, May 24, 2019, 12:23 PM EDT

RE: [#c11](#): [rouslan@](#):

A1: The requestor should be the actual iframe that made the request.

A2/A3: I am not sure if I understand the details, but I think it should be the origin that triggered the request by constructing `PaymentRequest` object - I am guessing this means the merchant?

Let's consider 1) <https://attacker.com> that wants to trigger 2) <https://victim.com/give-all-money-to-the-attacker> (with cookies) and 3) one of defenses <https://victim.com> has is rejecting all requests with 'Sec-Fetch-Site: cross-site'. We want to prevent <https://attacker.com> from being able to trigger request to <https://victim.com/give-all-money-to-the-attacker> with 'Sec-Fetch-Site: none' or 'Sec-Fetch-Site: same-origin'. I think my answers above should achieve this, right?

Thank you for a great example in [#c11](#) and asking: "when X.com causes Chrome to download Y.com and, after Chrome parses the data from Y.com, Chrome finds in there another URL that it needs to fetch, should the request\_initiator by X.com or Y.com." I don't know how to answer this question. I think either answer ensures that we won't send 'Sec-Fetch-Site: none' which is what the current bug is about. OTOH, the answer can change what Sec-Fetch-Site value is sent with the request + whether SameSite cookies are attached - therefore it is probably worth it to open a separate spec bug for this question (not sure exactly where such bug should be opened).

Comment 14 Deleted

Comment 15 by [rouslan@chromium.org](#) on Fri, May 24, 2019, 12:43 PM EDT

Filed a spec issue at <https://github.com/w3c/webappsec-fetch-metadata/issues/30>

Comment 16 by [infe...@chromium.org](#) on Tue, May 28, 2019, 1:33 AM EDT

Labels: Security\_Severity-Medium

Comment 17 by [sheriffbot@chromium.org](#) on Tue, May 28, 2019, 9:40 AM EDT

Labels: M-76 Target-76

Setting milestone and target because of Security\_Impact=Head and medium severity.

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

[Comment 18](#) by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Tue, May 28, 2019, 9:50 AM EDT

**Labels:** ReleaseBlock-Stable

This is a serious security regression. If you are not able to fix this quickly, please revert the change that introduced it.

If this doesn't affect a release branch, or has not been properly classified for severity, please update the Security\_Impact or Security\_Severity labels, and remove the ReleaseBlock label. To disable this altogether, apply ReleaseBlock-NA.

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

[Comment 19](#) by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Tue, May 28, 2019, 10:05 AM EDT

**Labels:** -ReleaseBlock-Stable ReleaseBlock-NA OS-Android OS-Chrome OS-Linux OS-Mac OS-Windows

Manifest fetching has been in Chrome for at least half a year and Sec-Fetch-Site is a new security feature. Therefore, this is not a regression. I still plan to fix it asap, of course.

[Comment 20](#) by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Tue, May 28, 2019, 10:06 AM EDT

**Status:** Started (was: Assigned)

**Labels:** Pri-1

[Comment 21](#) by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Tue, May 28, 2019, 11:25 AM EDT

**Cc:** danyao@chromium.org

Danyao, FYI

[Comment 22](#) by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Wed, Jun 5, 2019, 9:00 AM EDT

**Labels:** -Security\_Impact-Head Security\_Impact-Beta

[Comment 23](#) by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Thu, Jun 6, 2019, 3:34 PM EDT

**Blockedon:** 971339

[Comment 24](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Tue, Jun 11, 2019, 2:09 PM EDT

**Labels:** -Target-76 -M-76 Target-77

Changing the target milestone based on the sync-up with rouslan@ last week (the hope is to finish the work on the fix in one of the next 2 sprints, both of which should happen within M77).

[Comment 25](#) by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Wed, Jun 12, 2019, 9:51 AM EDT

**Labels:** M-75 Target-75

Setting milestone and target because of Security\_Impact=Beta and medium severity.

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

[Comment 26](#) by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Thu, Jun 13, 2019, 9:00 AM EDT

**Labels:** -Security\_Impact-Beta Security\_Impact-Stable

[Comment 27](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Thu, Jun 27, 2019, 12:44 PM EDT

**Blocking:** 979231

[Comment 28](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Tue, Jul 2, 2019, 7:38 PM EDT

**Labels:** -Target-75 -Target-77 Target-78

M77 is unlikely at this point as discussed over chat with rouslan@.

[Comment 29](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Wed, Jul 17, 2019, 2:57 PM EDT

**Cc:** jerryzz@google.com

[Comment 30](#) by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Wed, Jul 31, 2019, 9:01 AM EDT

**Labels:** -M-75 M-76 Target-76

[Comment 31](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Mon, Aug 12, 2019, 12:19 PM EDT

rouslan@, can you please give a quick status update? Is this still on track for being fixed in M78 (which according to [go/chrome-schedule](#) branches on Sep 5th - in slightly less than 4 weeks)?

[Comment 32](#) by [danyao@chromium.org](mailto:danyao@chromium.org) on Fri, Aug 16, 2019, 5:12 PM EDT

**Status:** Assigned (was: Started)

**Owner:** danyao@chromium.org

**Labels:** -Target-76 -M-76 Needs-Feedback Pri-2

lukasza@ - I'm the new owner of this bug and trying to understand the priority. Based on "Fetch Metadata Request Headers"[1], my understanding is that the Sec-Fetch-\* headers are designed to be a hint to the destination server to avoid unnecessarily revealing sensitive user information. I feel this use case may not apply to PaymentManifestDownloader, because of two reason:

1. The request is sent without user credentials [2], so server cannot identify who the user is.
2. The destination file is expected to be a static JSON manifest [3] that is intended for consumption by a user agent only. In fact, the browser doesn't pass any information from this file to the renderer.

So I don't think we need to fix this. Can you double check my logic? Thanks!

[1] <https://w3c.github.io/webappsec-fetch-metadata/>

[2] [https://cs.chromium.org/chromium/src/components/payments/core/payment\\_manifest\\_downloader.cc?l=323&q=CredentialsMode::kOmit](https://cs.chromium.org/chromium/src/components/payments/core/payment_manifest_downloader.cc?l=323&q=CredentialsMode::kOmit)

[3] <https://w3c.github.io/payment-method-manifest/#format>

[Comment 33](#) by [lukasza@chromium.org](mailto:lukasza@chromium.org) on Fri, Aug 16, 2019, 5:22 PM EDT

**Cc:** rouslan@chromium.org

**Labels:** Pri-1

Sec-Fetch-Site is a security feature that http servers can use to avoid handling requests initiated by cross-site websites. The security feature can be used both to

- 1) prevent cross-site data disclosure / revealing sensitive user information (e.g. as a more secure [no timing side-channels] version of Cross-Origin-Read-Blocking or Cross-Origin-Resource-Policy) and
- 2) as a CSRF defense (e.g. to reject "give money to the attacker" requests with "Sec-Fetch-Site: cross-site", even if the request has appropriate session cookies).

If an attacker-controlled website can force "Sec-Fetch-Site: none" then it defeats the 2nd kind of protection Sec-Fetch-Site tries to offer. (whether protection #1 is defeated depends on whether the initiator/attacker gets the response body - I assume this is not the case here.)

Based on the above, this is very much a security bug that should be fixed.

According to <https://chromium.googlesource.com/chromium/src/+/-/master/docs/security/severity-guidelines.md#toc-medium-severity>, bugs like this one are normally assigned priority Pri-1 and assigned to the current stable milestone (or earliest milestone affected). If the fix seems too complicated to merge to the current stable milestone, they may be assigned to the next stable milestone. Please prioritize fixing this bug - hopefully not much work is left: AFAIK rouslan@ had a WIP CL for propagating request initiator that was mostly blocked on exposing RenderFrameHost::GetLastCommittedOrigin to Java (see <https://chromium-review.googlesource.com/c/chromium/src/+/-/1636635>).

Comment 34 by danyao@chromium.org on Fri, Aug 16, 2019, 5:45 PM EDT

Thanks lukasza@. You highlight a use case I didn't think of: an attacker can create a PaymentRequest like this:

```
const request = new PaymentRequest([{"supportedMethods": "https://evil.com"}], details);
```

This causes Chrome to perform a GET on <https://evil.com>. If it returns the following in the response:

```
link: <https://victim.com/give-money-to?dest=evil.com>; rel="payment-method-manifest"
```

Then Chrome will call GET <https://victim.com/give-money-to?dest=evil.com>.

Because Chrome does not send any part of the response (header or body) with the renderer controlled by [evil.com](https://evil.com), protection #1 does not add anything - is this correct?

Protection #2 may still be needed, if the mere act of calling GET <https://victim.com/give-money-to?dest=evil.com> is the attack. Is this reasoning sound?

Comment 35 by lukasza@chromium.org on Fri, Aug 16, 2019, 5:59 PM EDT

RE: #c34: Right - this bug shows that PaymentRequests can be abused to bypass protection #2 (protection against CSRF). Based on your description, protection #1 (protection against cross-site data disclosure) seems to work just fine for PaymentRequests.

I think it is reasonable to treat as a potential attack the act of calling GET <https://victim.com/give-money-to?dest=evil.com>:

- If the server wants to use CSRF protection of Sec-Fetch-Site, then it will reject requests with 'Sec-Fetch-Site: cross-site' and handle the request otherwise (note that the request may be cookie-authenticated but it may also be authorized by ambient-authority like being issued on intranet/corpnat/private-network).
- I assume that in some cases the server cannot reject requests with 'Sec-Fetch-Site: none' (e.g. because of a need to handle requests initiated by user interacting with the browser UI - bookmarks, open-in-new-tab, etc.)

Comment 36 by rouslan@google.com on Mon, Aug 19, 2019, 8:20 AM EDT

FYI, <http://evil.com> cannot have a link to <https://victim.com> because Chrome requires a same-origin link.

Comment 37 by lukasza@chromium.org on Mon, Aug 19, 2019, 11:43 AM EDT

RE: #c36: rouslan@:

AFAIU there are 3 origins involved:

1. Merchant origin (or attacker origin) - the origin triggering the payment request
2. Payment method manifest URL/origin (handled via PaymentManifestDownloader::DownloadPaymentMethodManifest). #1 controls/provides this URL/origin and AFAICT this can be any URL/origin (at least the examples I found [1] make it seem that any URL/origin can be used. I didn't fully grok <https://w3c.github.io/payment-method-manifest/#fetch-pmm>, so maybe I am still missing something here).
3. Web app manifest URL/origin (handled via PaymentManifestDownloader::DownloadWebAppManifest). #2 controls/provides this URL/origin and it is enforced to be the same origin as #2 in PaymentManifestDownloader::OnURLLoaderCompleteInternal (not sure if I got it right - I thought that #2 and #3 need to be the same "site", but not necessarily the same origin)

If #1 can be cross-site from #2, then I believe PaymentRequest can be abused by #1 to attempt a CSRF attack on #2.

Did I get this right?

[1] <https://github.com/w3c/webappsec-fetch-metadata/issues/30>:

A merchant website calls:

```
new PaymentRequest([{"supportedOrigins": 'http://google.com/pay'}],
  {total: {label: 'Total', amount: {value: '1.00', currency: 'USD'}}});
```

Comment 38 by rouslan@chromium.org on Mon, Aug 19, 2019, 11:53 AM EDT

> Did I get this right?

Yeah, sounds about right.

> abused by #1 to attempt a CSRF attack on #2.

Meaning that the merchant can cause Chrome to download data from <https://google.com/pay>. Correct. My understanding previously was: this is the same as embedding an <iframe src="https://google.com/pay">. But now I realize that an <iframe> would send the Fetch-Site headers, which can be used to safeguard the target site, whereas PaymentRequest currently does not send that header. This is the gist of the attack, right?

Comment 39 by danyao@chromium.org on Mon, Aug 19, 2019, 12:13 PM EDT

Re: #c37: lukasza@:

> If #1 can be cross-site from #2, then I believe PaymentRequest can be abused by #1 to attempt a CSRF attack on #2.

Would #1 still be able to pull off the CSRF attack if Chrome only performs a HEAD (instead of GET) request on #2?

I'm guessing yes, because it depends on #1's implementation - it may not differentiate between HEAD and GET requests and still create a side effect. Is this right?

Comment 40 by lukasza@chromium.org on Mon, Aug 19, 2019, 12:19 PM EDT

RE: #c38: rouslan@:

Exactly right - the difference between <iframe>, <img>, XHR VS PaymentRequest is that

1. <iframe> navigation and <img>-or-XHR subresource fetch preserve the initiator origin and therefore will send correct Sec-Fetch-Site request header
2. PaymentRequest doesn't preserve the initiator origin (at least, until some form of <https://chromium-review.googlesource.com/c/chromium/src/+/-/1636635> lands)

RE: #c39: danyao@:

Right - we should send the correct Sec-Fetch-Site header for all http methods (including HEAD, OPTIONS, GET, etc.).

Comment 41 by mmoroz@google.com on Tue, Aug 20, 2019, 12:55 AM EDT

Labels: M-76

Comment 42 by a...@google.com on Tue, Aug 20, 2019, 5:17 AM EDT

FWIW we discussed this a bit in <https://github.com/w3c/webappsec-fetch-metadata/issues/30> and I attempted to outline what I think is the desired behavior in <https://github.com/w3c/webappsec-fetch-metadata/issues/30#issuecomment-495928696>

Comment 43 by rouslan@chromium.org on Thu, Aug 29, 2019, 3:45 PM EDT

Cc: tedc...@chromium.org dominickn@chromium.org

Comment 44 by rouslan@chromium.org on Fri, Aug 30, 2019, 1:02 PM EDT

**Blockedon:** 980641

[Comment 45](#) by [sheriffbot@chromium.org](#) on Tue, Sep 3, 2019, 9:00 AM EDT

danyao: 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 46](#) by [rouslan@chromium.org](#) on Tue, Sep 3, 2019, 9:30 AM EDT

**Owner:** [rouslan@chromium.org](#)

**Cc:** [-rouslan@chromium.org](#)

Work for this issue is in progress at <https://crrev.com/c/1774322>.

[Comment 47](#) by [sheriffbot@chromium.org](#) on Wed, Sep 11, 2019, 9:01 AM EDT

**Labels:** -M-76 M-77 Target-77

[Comment 48](#) by [sheriffbot@chromium.org](#) on Wed, Sep 18, 2019, 9:05 AM EDT

rouslan: 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 49](#) by [rouslan@chromium.org](#) on Wed, Sep 18, 2019, 4:58 PM EDT

<https://crrev.com/c/1774322> is still a work in progress, but it's getting close to being done.

[Comment 50](#) by [bugdroid](#) on Mon, Sep 30, 2019, 5:54 PM EDT

The following revision refers to this bug:

<https://chromium.googlesource.com/chromium/src.git/+63cdb148f496d531720a4edec8376e66b97f620e>

commit 63cdb148f496d531720a4edec8376e66b97f620e

Author: Rouslan Solomakhin <[rouslan@chromium.org](#)>

Date: Mon Sep 30 21:53:14 2019

[Android] Expose origin of render frame host in Java.

~~Bug-966507~~, 980641, ~~974230~~

Change-Id: I2ef7c051ba8b2cd9c6cc952743bfb3a55ab623ad

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1774322>

Reviewed-by: Andrew Grieve <[agrieve@chromium.org](#)>

Reviewed-by: Daniel Cheng <[dcheng@chromium.org](#)>

Reviewed-by: Ted Choc <[tedchoc@chromium.org](#)>

Commit-Queue: Rouslan Solomakhin <[rouslan@chromium.org](#)>

Cr-Commit-Position: refs/heads/master@{#701304}

[modify] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/browser/BUILD.gn>

[modify] [https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/browser/frame\\_host/render\\_frame\\_host\\_android.cc](https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/browser/frame_host/render_frame_host_android.cc)

[modify] [https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/browser/frame\\_host/render\\_frame\\_host\\_android.h](https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/browser/frame_host/render_frame_host_android.h)

[modify] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/android/BUILD.gn>

[modify]

<https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/android/java/src/org/chromium/content/browser/framehost/RenderFrameHostImpl.java>

[modify] [https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/android/java/src/org/chromium/content\\_public/browser/RenderFrameHost.java](https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/android/java/src/org/chromium/content_public/browser/RenderFrameHost.java)

[modify]

[https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/test/android/javatests/src/org/chromium/content\\_public/browser/test/mock/MockRenderFrameHost.java](https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/content/public/test/android/javatests/src/org/chromium/content_public/browser/test/mock/MockRenderFrameHost.java)

meHost.java

[modify] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/BUILD.gn>

[add] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/android/java/src/org/chromium/url/Origin.java>

[add] [https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/android/origin\\_android.cc](https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/android/origin_android.cc)

[modify] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/mojom/origin.mojom>

[modify] <https://crrev.com/63cdb148f496d531720a4edec8376e66b97f620e/url/origin.h>

[Comment 51](#) by [mkwst@chromium.org](#) on Tue, Oct 1, 2019, 8:52 AM EDT

**Components:** Blink>SecurityFeature>FetchMetadata

[Comment 52](#) by [dougman@google.com](#) on Tue, Oct 15, 2019, 1:25 AM EDT

**Labels:** -Needs-Feedback

[Comment 53](#) by [adetaylor@google.com](#) on Fri, Oct 18, 2019, 5:49 PM EDT

rouslan@ do you consider this fixed? If so please could you mark it thus? However I'm assuming not if the commit in [#c50](#) is Android-specific.

[Comment 54](#) by [lukasza@chromium.org](#) on Fri, Oct 18, 2019, 5:51 PM EDT

AFAIU [r701304](#) is not an actual fix, but a prerequisite for fixing this bug - to fix this bug, we actually need to start using the origin of the RFH as the initiator origin.

[Comment 55](#) by [rouslan@google.com](#) on Sat, Oct 19, 2019, 2:50 PM EDT

[lukasza@](#) is correct. This is not fixed.

[Comment 56](#) by [sheriffbot@chromium.org](#) on Wed, Oct 23, 2019, 9:12 AM EDT

**Labels:** -M-77 M-78

[Comment 57](#) by [mkwst@chromium.org](#) on Tue, Dec 3, 2019, 9:45 AM EST

Ping. Is there any movement here, Rouslan?

[Comment 58](#) by [rouslan@chromium.org](#) on Tue, Dec 3, 2019, 12:27 PM EST

The work in <https://chromium-review.googlesource.com/1636635> was suspended for a while, but I'm picking it up now after talking to [mkwst@](#) over chat.

Comment 59 by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Wed, Dec 11, 2019, 9:12 AM EST

Labels: -M-78 Target-79 M-79

Comment 60 by [dominickn@chromium.org](mailto:dominickn@chromium.org) on Wed, Jan 22, 2020, 12:37 PM EST

Friendly security marshal ping: any more progress on fixing this?

Comment 61 by [rouslan@google.com](mailto:rouslan@google.com) on Wed, Jan 22, 2020, 1:11 PM EST

Work is in progress :-D

Comment 62 by [bugdroid](mailto:bugdroid) on Fri, Jan 31, 2020, 5:18 PM EST

The following revision refers to this bug:

<https://chromium.googlesource.com/chromium/src.git/+c71cca62949021fe70c170ce68a020fe407562d2>

commit c71cca62949021fe70c170ce68a020fe407562d2

Author: Rouslan Solomakhin <[rouslan@chromium.org](mailto:rouslan@chromium.org)>

Date: Fri Jan 31 22:15:35 2020

[Payment Handler] Set request initiator for manifest fetches.

This patch passes the origin of the iframe that called the PaymentRequest API to the fetcher of manifest files, which uses it for sec-fetch-site headers. This patch is for service worker based payment handlers on both Android and desktop. This patch does not alter how iOS or native Android apps work.

[Bug-666507](#)

Change-Id: I97b6ca08fd9e08d1762f95d7b78edf7d73450aee

Reviewed-on: <https://chromium-review.googlesource.com/c/chromium/src/+1636635>

Commit-Queue: Rouslan Solomakhin <[rouslan@chromium.org](mailto:rouslan@chromium.org)>

Reviewed-by: Lukasz Anforowicz <[lukasza@chromium.org](mailto:lukasza@chromium.org)>

Reviewed-by: Danyao Wang <[danyao@chromium.org](mailto:danyao@chromium.org)>

Cr-Commit-Position: refs/heads/master@{#737484}

[modify]

<https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/android/java/src/org/chromium/chrome/browser/payments/PaymentAppFactoryParams.java>

[modify] <https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/android/java/src/org/chromium/chrome/browser/payments/PaymentRequestImpl.java>

[modify]

<https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/android/java/src/org/chromium/chrome/browser/payments/ServiceWorkerPaymentAppBridge.java>

[modify] <https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/BUILD.gn>

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/android/service\\_worker\\_payment\\_app\\_bridge.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/android/service_worker_payment_app_bridge.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/manifest\\_verifier\\_browser\\_test.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/manifest_verifier_browser_test.cc)

[add] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/sec\\_fetch\\_site\\_browser\\_test.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/sec_fetch_site_browser_test.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/service\\_worker\\_payment\\_app\\_finder\\_browser\\_test.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/browser/payments/service_worker_payment_app_finder_browser_test.cc)

[modify] <https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/chrome/test/BUILD.gn>

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/content/android/payment\\_manifest\\_downloader\\_android.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/content/android/payment_manifest_downloader_android.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/installable\\_payment\\_app\\_crawler.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/installable_payment_app_crawler.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/installable\\_payment\\_app\\_crawler.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/installable_payment_app_crawler.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/manifest\\_verifier.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/manifest_verifier.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/manifest\\_verifier.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/manifest_verifier.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_app\\_factory.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_app_factory.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_request.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_request.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_request.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_request.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_request\\_state.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_request_state.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_request\\_state.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_request_state.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment\\_request\\_state\\_unittest.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/payment_request_state_unittest.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service\\_worker\\_payment\\_app\\_factory.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service_worker_payment_app_factory.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service\\_worker\\_payment\\_app\\_finder.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service_worker_payment_app_finder.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service\\_worker\\_payment\\_app\\_finder.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/content/service_worker_payment_app_finder.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment\\_manifest\\_downloader.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment_manifest_downloader.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment\\_manifest\\_downloader.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment_manifest_downloader.h)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment\\_manifest\\_downloader\\_unittest.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/payment_manifest_downloader_unittest.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/test\\_payment\\_manifest\\_downloader.cc](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/test_payment_manifest_downloader.cc)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/test\\_payment\\_manifest\\_downloader.h](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/payments/core/test_payment_manifest_downloader.h)

[add] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/test/data/payments/payment\\_request\\_creator.html](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/test/data/payments/payment_request_creator.html)

[add] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/test/data/payments/payment\\_request\\_creator.js](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/components/test/data/payments/payment_request_creator.js)

[modify] [https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/ios/chrome/browser/payments/ios\\_payment\\_instrument\\_finder.mm](https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2/ios/chrome/browser/payments/ios_payment_instrument_finder.mm)

Comment 63 by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Sun, Feb 2, 2020, 5:45 AM EST

<https://crrev.com/c71cca62949021fe70c170ce68a020fe407562d2> fixed the issue for service worker based payment handlers. The issue is still present for native Android payment apps, so let's keep this bug open until that is fixed.

Comment 64 by [rouslan@chromium.org](mailto:rouslan@chromium.org) on Mon, Feb 3, 2020, 12:20 PM EST

Status: Fixed (was: Assigned)

Lukasz pointed out that Android payment apps are setting Sec-Fetch-Site using an opaque origin now, so it's always cross-origin. That means this bug itself is fixed. Setting the correct origin for Android payment apps is tracked in <https://crrev.com/1048204>.

Comment 65 by [sheriffbot@chromium.org](mailto:sheriffbot@chromium.org) on Tue, Feb 4, 2020, 12:16 PM EST

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

Comment 66 by [sheriffbot](mailto:sheriffbot) on Mon, May 11, 2020, 2:55 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 67 by [adetaylor@google.com](mailto:adetaylor@google.com) on Fri, May 15, 2020, 3:55 PM EDT

Labels: Release-0-M83

Comment 68 by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Mon, May 18, 2020, 11:59 AM EDT

Labels: CVE-2020-6483 CVE\_description-missing

Comment 69 by [adetaylor@chromium.org](mailto:adetaylor@chromium.org) on Wed, May 20, 2020, 11:44 PM EDT

Labels: -CVE\_description-missing CVE\_description-submitted

