NEW **Bug 259787**

CVE-2025-66286 [WPE][GTK] Certain connections to remote sites cannot be intercepted using WebPage::send-request signal

Albrecht Dreß
Reported 2023-08-03 11:40:54 PDT

Created attachment 467194 [details]
sample application and HTML test input to reproduce the issue

OS version: Debian Bookworm/x86_64
Webkit GTK package: libwebkit2gtk-4.1 v. 2.40.3-2~deb12u2

Overview:
=========
Even if the request to access a remote site is intercepted in the WebPage::send-request signal handler, a socket connection is opened and –if applicable– the TLS handshake is performed. If the access is triggered e.g. by malicious HTML content in an e-mail, this will already give the attacker valuable information, so this might (should?) be considered a security bug.

Steps to Reproduce:
==========
See the attached sample code package "sample.tar.gz" (note: tested on Debian Bookworm, should work similarly on other Linux systems):

(1) Unpack the sample
Unpack the package, cd into the folder “sample”, and say “make”

(2) Log network traffic
In an other terminal, start “tcpdump” or a similar tool to listen on ports 80/tcp and 443/tcp, e.g.:

```
sudo tcpdump -vvv -K -X \( tcp port 80 or tcp port 443 \)
```

(3) Run test application
In “sample” run the application to display the included HTML file:

```
./samp-main Test.html
```

Status NEW
Resolution

Priority P2
Severity Major

Version Other

Hardware PC
OS Linux

Product Security

Component Security

Assignee
WebKitGTK+ bugs

Reported
2023-08-03 11:40 PDT

Modified
2026-04-25 18:18 PDT History

CC List
35 users Show

URL

Keywords InRadar

Duplicates (3)
~~287218~~ ~~288907~~ ~~309513~~ View as bug list

Depends on

Blocks

See Also
~~288907~~

Alias
CVE-2025-66286

The application prints (time stamps omitted)

--8<------------------------
webkit_web_extension_initialize: done!
web_page_created_cb: page 10 created for (null)
send_request_cb: uri
'http://ftp.de.debian.org/debian/doc/00-INDEX'
caught, redirect to 'about:blank', stop event emission
--8<------------------------

The HTML contains two “link” containers
(preconnect, stylesheet) triggering this event
without any further user interaction. The tcpdump
log shows a connect() to the remote site.

(4) Click link
Click on the link in the window. The application
prints

--8<------------------------
send_request_cb: uri 'https://www.posteo.de/'
caught, redirect to 'about:blank', stop event emission
--8<------------------------

The tcpdump log shows that the connection opened
in step (3) is closed, a new connect() to
www.posteo.de is opened, and the full (!) TLS
handshake is performed.

The sample package contains the tcpdump log in the
file tcpdump.log:
* start the test application at 19:06:59
* click the link at 19:07:39

Expected Results:
==========
No connection to the remote site must be opened,
and in particular no TLS handshake must occur if the
WebPage::send-request signal handler redirects the
request to a different location.

Speculation: the connection is established before
the WebPage::send-request is emitted, resulting in
this behavior.

Attachments

sample application and HTML test input to reproduce the issue (14.85 KB, application/gzip)
2023-08-03 11:40 PDT, Albrecht Dreß

no flags Details

Add attachment *proposed patch, testcase, etc.* View All

Michael Catanzaro
Comment 1 2025-07-09 07:33:25 PDT

There is a corresponding Evolution issue report: https://gitlab.gnome.org/GNOME/evolution/-/issues/27

But I think this bug report contains everything we need to know. send-request is indeed supposed to be emitted, allowing the application to stop the TCP connection before it happens. Evidently something is wrong.

Michael Catanzaro
Comment 2 2025-07-09 07:57:49 PDT

Your test cases uses rel="preconnect" and rel="stylesheet". There are a bunch of other cases that we should test as well:

https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Attributes/rel

dns-prefetch, icon, modulepreload, pingback, prefetch, preload, prerender

Hopefully these will all be fixable in one place and not require separate fixes.

Albrecht Dreß
Comment 3 2025-07-09 12:13:56 PDT

Hi, great that someone takes care of this rather old bug!

> Your test cases uses rel="preconnect" and rel="stylesheet". There are a bunch of other cases that we should test as well:

I know – my example basically should only demonstrate that an attacker could exploit the bug both without and with any user interaction. There are of course plenty of other options for him to “use” it...

beanbo
Comment 4 2025-07-09 14:11:09 PDT

I already reported this issue as a security issue of Webkit and got no response...

Michael Catanzaro
Comment 5 2025-07-09 14:57:02 PDT

*** ~~Bug 287218~~ has been marked as a duplicate of

this bug. ***

Radar WebKit Bug Importer
Comment 6 2025-07-10 05:35:36 PDT

<rdar://problem/155518218>

renrenking86
Comment 7 2025-07-17 02:31:35 PDT

Comment on attachment 467194 [details]
sample application and HTML test input to reproduce the issue

renrengornica86@gmail.com

renrenking86
Comment 8 2025-07-17 02:31:58 PDT

9ok

Michael Catanzaro
Comment 9 2025-07-19 07:27:03 PDT

rel="dns-prefetch" might be tricky, because that is not an HTTP request, so we *can't* emit send-request. In https://gitlab.gnome.org/GNOME/evolution/-/issues/3(I indicated that we do not need to bring back the enable-dns-prefetching setting, but I think this is wrong. We will need to undeprecate it and implement it. (Currently, there is no way to control it.)

Moreover, in https://gitlab.gnome.org/GNOME/balsa/-/issues/99 and https://gitlab.gnome.org/GNOME/geary/-/issues/1680, Mike discovered that rel="preconnect" only creates a TLS connection, not an HTTP request. So again, relying on send-request won't be sufficient. We'll need yet another setting to control this.

Everything else looks like an HTTP request, and we need to make sure they are all blockable via WebKitWebPage::send-request.

So that's a lot of stuff that needs to be fixed. I think it's fair to say this bug should only be closed if all of the above is resolved.

Albrecht Dreß
Comment 10 2025-08-03 10:15:14 PDT

[sorry for the late reply/comment, I've been on vacation, away from my computer...]

(In reply to Michael Catanzaro from comment #9)
> rel="dns-prefetch" might be tricky, because that is not an HTTP request, so we *can't* emit send-request. In https://gitlab.gnome.org/GNOME/evolution/-/issues/3 I indicated that we do not need to bring back the enable-dns-prefetching setting, but I think this is wrong. We will need to undeprecate it and implement it. (Currently, there is no way to control it.)

IMHO, the DNS query is *not* a privacy (or even a security) issue, as the application will ask the system's configured DNS server (ideally one from https://www.joindns4.eu/ or similar) for the IP address of the potentially malicious site, but *not* connect the site itself. Please correct me if you are aware of any criminal or APT actor who actually could nevertheless abuse such lookups. A wrong DNS configuration (i.e. contacting a DNS server which logs requests and shares the data with a secret service or criminals) is a general problem and beyond the scope of Webkit. IOW, I don't see a valid reason for blocking these requests.

> Moreover, in https://gitlab.gnome.org/GNOME/balsa/-/issues/99 and https://gitlab.gnome.org/GNOME/geary/-/issues/168(Mike discovered that rel="preconnect" only creates a TLS connection, not an HTTP request. So again, relying on send-request won't be sufficient. We'll need yet another setting to control this.

Well, this is exactly what happens in my initial example...

IMHO, the key for a solution is to catch the connect() system call for SOCK_STREAM sockets.

Just my € 0.01, though...

Michael Catanzaro
Comment 11 2025-08-03 11:02:19 PDT

> Well, this is exactly what happens in my initial example...
>
> IMHO, the key for a solution is to catch the connect() system call for
> SOCK_STREAM sockets.

But we have no way for applications to do this.

I think we just need a new setting to completely disable preconnect.

Real
Comment 15 2025-11-16 23:41:18 PST

Can you please increase the priority and severity of this bug? It is unacceptable that this has been languishing for so long, considering that this is known to affect privacy (and also reported elsewhere and discussed on other SM threads), .

Michael Catanzaro
Comment 16 2025-11-17 07:43:07 PST

(Sure, but I highly doubt anybody actually looks at the Priority or Severity fields on Bugzilla to decide what to work on....)

Real
Comment 17 2025-11-17 07:52:54 PST

So like this is gonna stay unfixed for the foreseeable future? The bug was reported 2+ years ago.

I want a clear answer since I'm responsible for IT security policies in my org and many people probably use programs depending on it. If this isn't gonna get fixed, I would like to mark all programs using this as exploited in the company's IT policy.

I don't have the know-how to produce a patch for fixing this so I can't contribute code, and I can't just sit waiting for nobody.

Real
Comment 18 2025-11-17 08:01:46 PST

(If this has been determined to not be a security issue then please correct me)

Michael Catanzaro
Comment 19 2025-11-17 08:07:58 PST

This is an open source project. Issues don't fix themselves. Somebody has to volunteer to work on it. If nobody cares enough to work on it, then it will never be fixed.

Michael Catanzaro
Comment 20 2025-11-17 08:08:37 PST

This is definitely a security bug.

Albrecht Dreß
Comment 21 2025-11-17 11:20:04 PST

Yes, it is a security bug, and it should be fixed.

However, IMHO its impact is somewhat limited. As I pointed out in my initial report, an attacker can gain some information about the targeted system, basically (due to the connect() system call) that it exists. As worst case a TLS handshake is performed before the WebPage::send-request callback kicks in, blocking all further transactions.

Whilst this might of course be used for information harvesting (no idea if any attackers actually use this particular technique, though), this bug does not leak any other information, nor is it possible to inject malicious code. I.e. IMHO the confidentiality impact is rather low, integrity and availability of the target system are not affected.

I have no idea how to correctly calculate a CVSS, but maybe it would be helpful to create a CVE entry for this bug with a proper score as to avoid confusion about its criticality.

Real
Comment 22 2025-11-17 12:07:45 PST

Hmm... something like this?

CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:L/VI:N/VA:N/S

Score 5.3

Used https://www.metaeffekt.com/security/cvss/calculator/

Albrecht Dreß
Comment 23 2025-11-18 10:20:20 PST

I think, after reading the specs, your assessment is correct. However, the classification (in particular for VC in this case) is somewhat coarse. Afaict, the leaked information boils down to the IP address of the box (or typically the gateway, proxy, …), and its TLS capabilities (version, ciphers, …). My gut feeling is that 5.3 is rather high for that. It would be more critical if the connected (malicious) server could request a user certificate from the client, but I think (sic!) this would require a further callback in the application (never tried it, though).

If this is relevant at all for a particular application depends on whether it has the capability to intercept “unwanted” accesses to web sites using the

WebKitWebProcessExtension WebPage::send-request signal or not. This is typically the case for MUA's (I saw it in Balsa <https://gitlab.gnome.org/GNOME/balsa>), no idea about other applications.

Michael Catanzaro
Comment 24 2025-11-18 10:37:22 PST

I'll request a CVE.

Michael Catanzaro
Comment 25 2025-11-18 12:19:32 PST

*** ~~Bug 288907~~ has been marked as a duplicate of this bug. ***

Real
Comment 26 2025-11-18 18:40:05 PST

> Afaict, the leaked information boils down to the IP address of the box (or typically the gateway, proxy, ...), and its TLS capabilities (version, ciphers, ...). My gut feeling is that 5.3 is rather high for that. It would be more critical if the connected (malicious) server could request a user certificate from the client, but I think (sic!) this would require a further callback in the application (never tried it, though).

It's also the fact that the attacker can track the user and see (if, when, where, how many times) the user is using the computer and is online and has opened the email in real time.

Considering that, 5.3 is not that high.

Real
Comment 27 2025-11-18 18:45:07 PST

And since there is already an exploit code available (the person in evolution issue mentioned he has it on his open source website), a low rating would be misleading for such a targetted and personalised harvesting attack.

Albrecht Dreß
Comment 28 2025-11-19 10:06:36 PST

> It's also the fact that the attacker can track the user and see (if, when, where, how many times) the user is using the computer and is online and has opened the email in real time.

I don't think it's possible to really identify (or even track) any specific user (unless it would be possible to actually request a TLS client certificate), as the only information the attacker can extract is from the accept() system call, i.e. the struct sockaddr returned by it, plus of course time stamps. Any further transaction, like sending a GET request containing some kind of unique target identifier, *will* be intercepted by the WebPage::send-request signal. So this boils down to a (IP address; time stamp) statistics – given that ISP-assigned addresses change frequently, and the users of larger organisations will usually use some kind of proxy, I wonder how useful this is.

> And since there is already an exploit code available (the person in evolution issue mentioned he has it on his open source website), a low rating would be misleading for such a targetted and personalised harvesting attack.

The attachment to my original post (the HTML example) demonstrates how to (ab)use the bug; I wouldn't call that "exploit code"...

Michael Catanzaro
Comment 29 2025-11-19 10:36:30 PST

Well the email sender has your email address. Surely it's easy to create a unique tracking domain per email address, and use it nowhere else. *Any* IP connection indicates the mail has been read, violating the trust model.

It's also incorrect to assume that ISP-assigned addresses change frequently. Maybe that's true for your particular ISP, but it's certainly not true in general.

(In reply to Real from comment #22)
> CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:L/VI:N/VA:N/

I agree with this.

Michael Catanzaro
Comment 30 2025-11-26 12:29:04 PST

We received CVE-2025-66286.

Gergo K
Comment 31 2025-12-15 02:53:08 PST

I've been getting emails lately with images on suspicious hostnames. In these cases, all the subdomains resolve to the same IP.
I haven't seen one with preconnect yet, but I'm sure I will. These techniques are already being used to track users.

Tjareson
Comment 35 2026-03-10 12:39:26 PDT

I tested CVE-2025-66286 on my system and cannot reproduce the leak. The connection seems to be successfully blocked by Evolution.

My setup
OS: Linux Mint (Ubuntu 22.04 Jammy base)
Mail Client: Evolution 3.44.4
WebKitGTK: libwebkit2gtk-4.0-37 (Version 2.50.4-0ubuntu0.22.04.1)

I used the test mail from Email Privacy Tester (which includes <link rel="preconnect"> and <link rel="dns-prefetch">). I monitored the network using tcpdump on port 53 (DNS) and port 80/443.
Zero packets captured for the tracker domains upon opening the mail. Evolution seems to successfully block the preconnect/prefetch. The DNS lookups and HTTP requests only happen after I manually click "Load remote content" in Evolution.

Has this been silently mitigated in the 2.50.4 Ubuntu build, or does Evolution 3.44 configure the WebKit sandbox in a way that prevents this bug?

Milan Crha
Comment 36 2026-03-11 01:44:52 PDT

Evolution 3.44 is ancient, there is going to be release version 3.60.0 by the end of this week, which will be the new stable series. If you are wondering how ancient it is, the Evolution 3.44.4 is from Aug 5, 2022.

Tjareson
Comment 37 2026-03-11 02:43:35 PDT

Ah right. Ok, staying ancient looks like the best option then for the time being. (at least as far as the linux distribution cares for backports of critical fixes)
My understanding is 3.60.0 will not circumvent the current webkit issue.

Milan Crha
Comment 38 2026-03-11 04:04:31 PDT

It's not in the Evolution hands, this is filled against WebKit, because the problem is in the WebKit and

the WebKit does not provide any API nor setting to block all connection attempts, including the preconnects and all the others (it used to have a setting to disable `dns-prefetch`, but it's deprecated since 2.48 according to the doc: https://webkitgtk.org/reference/webkitgtk/2.50.5/meth).

I guess, from the process name, that WebKitNetworkProcess is responsible for the network connections. Ideally, for the WebKitGTK variant of the WebKit, would be to have an API to pass any network-related requests through the interested application(s) and to not use that process at all when the app advertises use of that new API. In short, not to use its own SoupSession (if it still uses it), but have a wrapper on top of it, where the apps could implement this wrapper and override anything the WebKit would like to get from it. That would be a game changer and a pretty cool thing ;)

Michael Catanzaro
Comment 39 2026-03-18 06:51:09 PDT

*** ~~Bug 309513~~ has been marked as a duplicate of this bug. ***

Note

You need to log in before you can comment on or make changes to this bug.

Top of Page | Format For Printing | XML | Clone This Bug