4

CVE-2022-27775: Bad local IPv6 connection reuse

Share:

TIMELINE

nyymi submitted a report to curl. Apr 20th (7 months ago)

Summary:

Curl doesn't consider IPv6 address zone index when doing connection reuse. if connection exists to specific IPv6 address (and other conditions for connection reuse are fulfilled) it will be reused for connections regardless of the zone index.

Steps To Reproduce:

1.Set up a fake server: `echo -ne 'HTTP/1.1 200 OK\r\nContent-Length: 6\r\n\r\nHello\n' | nc -6 -v -l -p 9999`
2. curl "http://[ipv6addr]:9999/x" "http://[ipv6addr%25lo]:9999/y"

Both connections arrive to the test server:

Code 218 Bytes Wrap lines Copy Download

```
Listening on :: 9999
Connection received on somehost someport
GET /x HTTP/1.1
Host: [ipv6addr]:9999
User-Agent: curl/7.83.0-DEV
Accept: */*

GET /y HTTP/1.1
Host: [ipv6addr]:9999
User-Agent: curl/7.83.0-DEV
Accept: */*
```

Clearly the 2nd connection should fail as the address is not available at interface lo. (Lone connection to `http://[ipv6addr%25lo]:9999/` fails with `curl: (7) Couldn't connect to server`)

Practical impact of this vulnerability is very low, due to the rarity of situation where interfaces would have identical addresses. The attacker would also need to be able to manipulate the addresses the victim app connects to (making it first connect to interface controlled by the attacker).Finally, it doesn't seem likely that TLS would be used for such connections, making the scenario rather insecure to begin with.It seems likely that if the attacker has ability to set up interfaces with identical addresses they would have easier way to compromise the system anyway.

bagder curl staff posted a comment. Apr 21st (7 months ago)

Thank you for your report!

We will take some time and investigate your reports and get back to you with details and possible follow-up questions as soon as we can!

bagder curl staff posted a comment. Apr 21st (7 months ago)

Zone ids are only for non-global addresses too, aren't they? That also limits the impact of this greatly:

curl http://[2a04:4e42:200::561%25lo]:80/ -H "Host: curl.se" -v

... works fine, even though it isn't on localhost and connects to the same host as

curl http://[2a04:4e42:200::561]:80/ -H "Host: curl.se" -v

bagder curl staff posted a comment. Apr 21st (7 months ago)

A minimum patch for this might be like this one below. I think this function could be enhanced a little more but I think that can be done in a separate take. Also, this fix makes my example above not reuse the connections even though they should when the address is global-scoped but I'm not sure it is worth bothering about that.

Code 795 Bytes Wrap lines Copy Download

```
diff --git a/lib/conncache.c b/lib/conncache.c
index ec669b971..8948b53fa 100644
--- a/lib/conncache.c
+++ b/lib/conncache.c
@@ -153,12 +153,16 @@ static void hashkey(struct connectdata *conn, char *buf,

   if(hostp)
-  /* put the number first so that the hostname gets cut off if too long */
-  msnprintf(buf, len, "%ld%s", port, hostname);
+  /* put the numbers first so that the hostname gets cut off if too long */
+#ifdef ENABLE_IPV6
+  msnprintf(buf, len, "%u/%ld/%s", conn->scope_id, port, hostname);
+#else
+  msnprintf(buf, len, "%ld/%s", port, hostname);
+#endif
   Curl_strntolower(buf, buf, len);
 }

 /* Returns number of connections currently held in the connection cache.
    Locks/unlocks the cache itself!


```

nyymi posted a comment. Updated Apr 21st (7 months ago)

> Zone ids are only for non-global addresses too, aren't they?

As far as I understand, yes. The most common use for them is when using link local addresses directly. https://en.wikipedia.org/wiki/Link-local_address#IPv6

It's possible to specify the zone index for global address, too, but I dunno how likely it would be that someone would be connecting to IPv6 addresses like that.

> curl http://[2a04:4e42:200::561%25lo]:80/ -H "Host: curl.se" -v
> ... works fine, even though it isn't on localhost and connects to the same host as

Interesting, it appears to work on Linux, indeed.

It fails for me on MacOS though:

`* Immediate connect fail for 2a04:4e42:200::561: No route to host`

I use lo0 there of course, or some other interface such as `en<n>` where `<n>` is other than the existing interface.

So it seems linux does some magic with the zone index. Maybe linux ignores them for global addresses?

That is certainly what it looks like.

nyymi posted a comment. Apr 21st (7 months ago)

The preliminary patch fixes the issue for me.

Code 1012 Bytes Wrap lines Copy Download

```
$ DYLD_LIBRARY_PATH="./lib/.libs/:$DYLD_LIBRARY_PATH" ./src/.libs/curl -v "http://[2a
*   Trying 2a04:4e42:200::561:80...
* Connected to 2a04:4e42:200::561 (2a04:4e42:200::561) port 80 (#0)
> GET / HTTP/1.1
> Host: curl.se
> User-Agent: curl/7.83.0-DEV
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 301 Moved Permanently
< Server: Varnish
< Retry-After: 0
< Location: https://curl.se/
< Content-Length: 0
< Accept-Ranges: bytes
< Date: Thu, 21 Apr 2022 08:41:06 GMT
< Via: 1.1 varnish
< X-Served-By: cache-hel1410026-HEL
< X-Cache: HIT
< X-Cache-Hits: 0
< X-Timer: S1650530466.290105,VS0,VE0
< Age: 0
< Connection: keep-alive
<
* Connection #0 to host 2a04:4e42:200::561 left intact
* Hostname 2a04:4e42:200::561 was found in DNS cache
*   Trying 2a04:4e42:200::561:80...
* Immediate connect fail for 2a04:4e42:200::561: No route to host
* Closing connection 1
curl: (7) Couldn't connect to server
```

transfer uses credentials or similar it would pass them along to the wrong receiver.

bagder curl staff changed the status to **Triaged**. Apr 21st (7 months ago)

bagder curl staff posted a comment. Apr 21st (7 months ago)

It seems this flaw was brought with this commit https://github.com/curl/curl/commit/2d0e9b40d3237b1, present in 7.65.0 and later. I tested 7.64.1 and it doesn't work with zone ids in the URL.

nyymi posted a comment. Updated Apr 21st (7 months ago)

CWE for this one isn't straight forward. Infoleak one might be appropriate, so CWE-200. Or maybe one of the more specific ones, such as "CWE-668: Exposure of Resource to Wrong Sphere". Mmm or await, CWE-200 is a child of CWE-668 actually.

I guess CWE-200 might work best then.

bagder curl staff posted a comment. Apr 21st (7 months ago)

Agreed, let's go with CWE-200

bagder curl staff updated CVE reference to **CVE-2022-27775**. Apr 21st (7 months ago)

Apr 21st (7 months ago)
bagder curl staff
changed the report title from **Bad IPv6 connection reuse due to ignoring zone index in connection matching** to **CVE-2022-27775: Bad local IPv6 connection reuse**.

bagder curl staff posted a comment. Apr 21st (7 months ago)

My first shot at an advisory for this:

Bad local IPv6 connection reuse

Project curl Security Advisory, April 27 2022 -
Permalink

VULNERABILITY

libcurl keeps previously used connections in a connection pool for subsequent transfers to reuse, if one of them matches the setup.

another.

We are not aware of any exploit of this flaw.

INFO

Zone ids are only used for non-global scoped IPv6 addresses and they are only used when specifying the address numerically.

This flaw has existed in curl since commit 2d0e9b40d3237b1, shipped in libcurl 7.65.0, released on May 22 2019. Previous versions will instead not accept URLs with zone ids.

The Common Vulnerabilities and Exposures (CVE) project has assigned the name CVE-2022-27775 to this issue.

CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

Severity: Low

AFFECTED VERSIONS

- Affected versions: curl 7.65.0 to and including 7.82.0
- Not affected versions: curl < 7.65.0 and curl >= 7.83.0

Also note that libcurl is used by many applications, and not always advertised as such.

THE SOLUTION

A fix for CVE-2022-27775

RECOMMENDATIONS

A - Upgrade curl to version 7.83.0

B - Apply the patch to your local version

C - Do not use non-global numerical IPv6 addresses for insecure URL schemes

TIMELINE

This issue was reported to the curl project on April 21, 2022. We contacted distros@openwall on April 21.

This advisory was posted on April 27, 2022.

Thanks a lot!

1 attachment:
F1701569: CVE-2022-27775.md

bagder curl staff posted a comment. Apr 21st (7 months ago)

I edited the top explanation a bit to now say:

VULNERABILITY

libcurl keeps previously used connections in a connection pool for subsequent transfers to reuse, if one of them matches the setup.

Due to errors in the logic, the config matching function did not take the IPv6 address zone id into account which could lead to libcurl reusing the wrong
connection when one transfer uses a zone id and a subsequent transfer uses another (or no) zone id.

We are not aware of any exploit of this flaw.

nyymi posted a comment. Apr 21st (7 months ago)

Looking good.

bagder curl staff closed the report and changed the status to ○ **Resolved**. Apr 27th (7 months ago)

Published. This issue is now eligible for a bounty claim from IBB.

nyymi requested to disclose this report. Apr 27th (7 months ago)

bagder curl staff agreed to disclose this report. Apr 27th (7 months ago)

This report has been disclosed. Apr 27th (7 months ago)

curl has decided that this report is not eligible for a bounty. May 13th (7 months ago)

Thanks for your work. The actual monetary reward part for this issue is managed by the Internet Bug Bounty so the curl project itself therefor sets the reward amount to **zero USD.** If you haven't already, please submit your reward request to them and refer back to this issue.