5

CVE-2022-22576: OAUTH2 bearer bypass in connection re-use

Share:

TIMELINE

monnerat submitted a report to curl. Mar 30th (8 months ago)

Summary:

A cached connection authenticated with the OAUTH2 mechanisms can be reused by a subsequent request even if the bearer is not correct.
This affects SASL-enabled protcols: SMPTP(S), IMAP(S), POP3(S) and LDAP(S) (openldap only).

An application that can be accessed by more than one user (such as a webmail server) would be affected by this flaw.

Steps To Reproduce:

```
curl 'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u user: --oauth2-bearer validbearer --next 'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u user: --oauth2-bearer anything
```

Supporting Material/References:

- Patch 0001-url-check-sasl-additional-parameters-for-connection-.patch fixes this flaw.

As an alternative to apply the patch, use another (unused) password within each request: the second request in the command below will properly fail.

```
curl 'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u user:dummy1 --oauth2-bearer validbearer --next 'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u user:dummy2 --oauth2-bearer anything
```

Impact

Access (read/write) unauthorized data

1 attachment:

As per previous mail conversation, this is a confirmed security issue.

bagder curl staff posted a comment. Mar 30th (8 months ago)

Thanks for this @monnerat,

I will re-review the patch and write up a first advisory within soon.

bagder curl staff posted a comment. Updated Mar 30th (8 months ago)

I think `CWE-305: Authentication Bypass by Primary Weakness` is suitable and is the one we used for previous reusing connections wrongly mistakes.

Examples: CVE-2014-0015, CVE-2014-0138 and CVE-2016-0755

bagder curl staff posted a comment. Apr 1st (8 months ago)

@monnerat since there was never previously any attempt to consider the oauth2 bearer in the connection reuse logic, I figure we can presume that this flaw has existed for as long as the support for SASL XOAUTH2 has been around, right?

It seems the first commit for this was https://github.com/curl/curl/commit/19a05c908f7d8be82de6f69f533317d8a0db49dd, shipped in curl 7.33.0 (this exact commit is one in a series so while this was the first with SASL OAUTH2 support it wasn't the exact single commit that brought the flaw)

Or was there any other logic back then that mitigated this problem that made it not possible to trigger so that it was introduced later?

monnerat posted a comment. Apr 1st (8 months ago)

> since there was never previously any attempt to consider the oauth2 bearer in the connection reuse logic, I figure we can presume that this flaw has existed for as long as the support for SASL XOAUTH2 has been around, right?

Yes.

> It seems the first commit for this was https://github.com/curl/curl/commit/19a05c908f7d8be82de6f69f533317d8a0db49dd, shipped in curl 7.33.0 (this exact commit is one in a series so while this was the first with SASL OAUTH2 support it wasn't the exact single commit that brought the flaw)

themselves and effective oauth2 feature was added in subsequent commits.
smtp:
https://github.com/curl/curl/commit/90ab65c632ec0405893466637c7971e327f1067a
imap:
https://github.com/curl/curl/commit/34122800b898596f3657f89621dd6762f227653f
pop3:
https://github.com/curl/curl/commit/18db7438512de1d6f63c616af5755ea2859597b8
SASL has been factorized out of protocol modules by commit
https://github.com/curl/curl/commit/79543caf908118d056353714ba2cb5c5348e20c2 and recently introduced in openldap by
https://github.com/curl/curl/commit/eeca818b1e8d1e61c2d4d833aed56ce4c510a9d4.

> Or was there any other logic back then that mitigated this problem that made it not possible to trigger so that it was introduced later?

Unless connection re-use was not enabled for these protocols at this time (I did not check), the problem exists since introduction of oauth2 in modules's SASL (7.33.0).

bagder curl staff posted a comment. Apr 1st (8 months ago)

> Unless connection re-use was not enabled for these protocols at this time

I can't remember any such special treatment. They did connection re-use already then.

bagder curl staff posted a comment. Apr 1st (8 months ago)

Here's my first attempt at a security advisory. Inlined and attached.

OAUTH2 bearer not-checked for connection re-use

Project curl Security Advisory, April 27th 2022 -
Permalink

VULNERABILITY

libcurl might reuse OAUTH2-authenticated connections without properly making sure that the connection was authenticated with the same credentials as set for this transfer. This affects SASL-enabled protcols: SMPTP(S), IMAP(S), POP3(S) and LDAP(S) (openldap only).

libcurl maintains a pool of connections after a transfer has completed. The pool of connections is then gone through when a new transfer is requested and

A connection that is successfully created and authenticated with a user name + OAUTH2 bearer could subsequently be reused even for user + [other OAUTH2 bearer], even though that might not even be a valid bearer. This could lead to an authenticion bypass, either by mistake or by a malicious actor.

We are not aware of any exploit of this flaw.

INFO

This flaw was introduced in the commit series that started with 19a05c908f7d8be82.

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

CWE-305: Authentication Bypass by Primary Weakness

Severity: Medium

AFFECTED VERSIONS

- Affected versions: curl 7.33.0 to and including 7.82.0
- Not affected versions: curl < 7.33.0 and curl >= 7.83.0

Note that libcurl is used by many applications, but not always advertised as such.

THE SOLUTION

A fix for CVE-2022-AARGH

(This URL is for pre-announcement only)

RECOMMENDATIONS

We suggest you take one of the following actions immediately, in order of preference:

A - Upgrade curl and libcurl to version 7.83.0

B - Apply the patch to your version and rebuild

C - Avoid using OAUTH2 bearer authentication with SMTP, IMAP, POP3 and LDAP

TIME LINE

libcurl 7.83.0 was released on April 27 2022, coordinated with the publication of this advisory.

CREDITS

Reported and patched by Patrick Monnerat.

Thanks a lot!

1 attachment:
F1675636: CVE-2022-AARGH.md

monnerat posted a comment. Apr 1st (8 months ago)

> As an alternative to apply the patch, use another (unused) password within each request: the second request in the command below will properly fail.

A better mitigation would be to also set the password with the bearer's value: it won't affect the authentication itself and is already checked in connection reuse logic:

```
curl 'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u
user:validbearer --oauth2-bearer validbearer --next
'imap://server:port/path/;MAILINDEX=1' --login-options 'AUTH=OAUTHBEARER' -u
user:anotherbearer --oauth2-bearer anotherbearer
```

bagder curl staff posted a comment. Apr 1st (8 months ago)

Ah yes, good point. I'll amend the recommendation.

monnerat posted a comment. Apr 1st (8 months ago)

> Ah yes, good point. I'll amend the recommendation.

Please note that this is not compatible with alternative mechanisms: if plain and oauth2 are enabled, this would conflict with the plain password !

bagder curl staff posted a comment. Apr 1st (8 months ago)

For the bounty-reward, the process is slightly updated and you will be the first one to try this out with a curl security bug @monnerat. As we're now part of the "IBB", the Internet Bug Bounty.

monnerat posted a comment. Apr 1st (8 months ago)

Thanks for the info.

As the linked page has no submission form, I presume I'll have to mail the CVE to ibb@hackerone.com ? Or content might change as you act on this ?

Let's wait and see !

bagder curl staff posted a comment. Apr 1st (8 months ago)

There should be a "Submit report" button at the top of that page. See screenshot.

1 attachment:

F1675707: ibb-submit-button.jpg

monnerat posted a comment. Apr 1st (8 months ago)

Ah yes, thanks!

bagder curl staff updated CVE reference to **CVE-2022-22576**. Apr 4th (8 months ago)

Apr 4th (8 months ago)

bagder curl staff

changed the report title from **Bypass the OAUTH2 bearer validation** to **CVE-2022-22576: OAUTH2 bearer bypass in connection re-use**.

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.

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

bagder curl staff disclosed this report. Apr 29th (7 months ago)

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

Bounty managed by IBB.