

8

CVE-2022-27778: curl removes wrong file on error

Share:



TIMELINE



nyymi submitted a report to [curl](#).

Apr 28th (7 months ago)

Summary:

Curl command has a logic flaw that results in removal of a wrong file when combining `--no-clobber` and `--remove-on-error` if the target file name exists and an error occurs.

Steps To Reproduce:

1. `echo "important file" > foo`
2. `echo -ne "HTTP/1.1 200 OK\r\nContent-Length: 666\r\n\r\nHello\n" | nc -l -p 9999`
3. `curl -m 3 --no-clobber --remove-on-error --output foo http://testserver.tld:9999/`
4. `ls -l foo*`
5. `cat foo.1`

`-m 3` is used here to simulate a denial of service of the connection performed by the attacker.

The bug appears to happen because the remote-on-error `unlink` is called without considering the no-clobber generated file name:

- no-clobber name generation;
https://github.com/curl/curl/blob/3fd1d8df3a2497078d580f43c17311e6f58186a1/src/tool_cb_wrt.c#L88
- remove-on-error unlink:
https://github.com/curl/curl/blob/f7f26077bc563375becdb2adbcd49eb9f28590f9/src/tool_operate.c#L598

Impact

Removal of a file that was supposed not to be overwritten (data loss). Incomplete file left of disk when it should have been removed. This can lead to potential loss of integrity or availability.



[bagder](#) curl staff posted a comment.

Apr 28th (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!



[nyymi](#) posted a comment.

Apr 28th (7 months ago)

Targeted attack would require the attacker to be able to induce an error somehow.

This could be a simple denial of service. A totally blind attack would be tricky as it would need the transmission to start, but then fail before completing. Timing the attack would be tricky, as well as actually causing an error as TCP is quite resilient.

Another scenario would be MitM of HTTPS connection, where the attacker could identify the actual connection being initiated (at least if it's a timed script or similar, or the connection can be fingerprinted) and then let the connection initiate, but after some amount of transfer cause connection termination by tampering with the TLS communication (any random change should be enough to cause an abort due to HMAC being incorrect).



[nyymi](#) posted a comment.

Updated Apr 28th (7 months ago)

Possibly related issue: combination of `--remote-name`, `--remote-header-name` and `--remove-on-error` won't lead to the file being removed on error. Partial file is left on disk.



[bagder](#) curl staff posted a comment.

Apr 28th (7 months ago)

It won't lead to the file being removed

That sounds just like a good old bug.



[bagder](#) curl staff posted a comment.

Apr 28th (7 months ago)

First attempt at patch. I need to write up a test or two as well...

Code 759 Bytes

[Wrap lines](#) [Copy](#) [Download](#)

```
1 diff --git a/src/tool_operate.c b/src/tool_operate.c
2 index cb587e0a6..618735924 100644
3 --- a/src/tool_operate.c
4 +++ b/src/tool_operate.c
5 @@ -594,12 +594,12 @@ static CURLcode post_per_transfer(struct GlobalConfig *global,
```

```

9      }
10     if(result && config->rm_partial) {
11 -     notef(global, "Removing output file: %s", per->outfile);
12 -     unlink(per->outfile);
13 +     notef(global, "Removing output file: %s", outs->filename);
14 +     unlink(outs->filename);
15     }
16 }
17
18 /* File time can only be set _after_ the file has been closed */
19 if(!result && config->remote_time && outs->s_isreg && outs->filename) {
20
21

```



nyyimi posted a comment.

Updated Apr 28th (7 months ago)

MitM of HTTPS connection

Just to clarify that this scenario would not require decrypting any of the TLS traffic. The attacker would merely inspect the TLS handshake to identify when the victim connects to the target host. Once connected, the attacker would then disrupt the TLS connection after set amount of packets have been transmitted (certain amount of file has been downloaded). When successful the curl command would fail => wrong file would be removed.

This of course is a blind attack in a sense that the attacker has a limited information about the request performed and has to guess which connection to disrupt. Transfers occurring at specific time (cron runs etc) to specific hosts would be easiest to target.



nyyimi posted a comment.

Apr 28th (7 months ago)

Tested the patch and it fixes both issues for me.



nyyimi posted a comment.

Apr 28th (7 months ago)

Linefeed is missing though, should be "Removing output file: %s\n" I think.



bagder curl staff changed the status to Triaged.

Apr 29th (7 months ago)

Confirmed security problem.


bagder curl staff posted a comment.

Apr 29th (7 months ago)

 [CWE-706](#) maybe?


CWE-706: Use of Incorrectly-Resolved Name or Reference

The software uses a name or reference to access a resource, but the name/reference resolves to a resource that is outside of the intended control sphere.

 [nyymi](#) posted a comment. Apr 29th (7 months ago)
Perhaps an attempt to use both `--continue-at` and `--no-clobber` should result in a warning or even an error.

 [nyymi](#) posted a comment. Apr 29th (7 months ago)
[CWE-706](#)
Sounds good to me.

 [bagder](#) curl staff updated CVE reference to [CVE-2022-27778](#). Apr 29th (7 months ago)

 Apr 29th (7 months ago)
[bagder](#) curl staff
changed the report title from Removal of wrong file with `--no-clobber --remove-on-error` to [CVE-2022-27778: curl removes wrong file on error](#).

 [bagder](#) curl staff posted a comment. Apr 29th (7 months ago)
Attached advisory draft.

1 attachment:
F1711571: [CVE-2022-27778.md](#)

 [dgustafsson](#) curl staff posted a comment. Apr 29th (7 months ago)
+1 on the advisory draft.

 [nyymi](#) posted a comment. Apr 29th (7 months ago)
Advisory is looking good to me.

[bagder](#) curl staff posted a comment. Apr 29th (7 months ago)



[bagder](#) curl staff posted a comment.

May 5th (7 months ago)

I have notified distros [@openwall](#) about this issue now. Set for announcement with the pending release on May 11.



[bagder](#) curl staff closed the report and changed the status to Resolved.

May 11th (7 months ago)

Published. This issue is now eligible for a bounty claim from [IBB](#).



[bagder](#) curl staff requested to disclose this report.

May 11th (7 months ago)



[nyymi](#) agreed to disclose this report.

May 11th (7 months ago)



This report has been disclosed.

May 11th (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.