

New issue

[Jump to bottom](#)

zstd adds read permissions to files while being compressed or uncompressed #1630

🔒 Closed chungy opened this issue on Jun 4, 2019 · 13 comments · Fixed by #1644

Assignees



Labels

good first issue

chungy commented on Jun 4, 2019

Contributor

While the final file mode is reflective of the input file, when compressing or uncompressing, the file can temporarily gain greater permissions than the input and potentially leading to security issues (especially if large files are being handled).

Example: `file` has mode `600` -> `zstd file` -> `file.zst` has mode `644` while compression is happening. (Same happens with `unzstd file.zst` and `file`)



1

Cyan4973 commented on Jun 4, 2019 • edited

Contributor

It's necessary to grant write access while modifying the file.
But it should be possible to restrict file access by other groups during compression and decompression.

A solution could be to enforce `600` rights during compression or decompression, and only then apply same rights as source file once the operation is completed.

 Cyan4973 added the `good first issue` label on Jun 4, 2019

chungy commented on Jun 4, 2019

Contributor

Author

A solution could be to enforce `600` rights during compression or decompression, and only then apply same rights as source file once the operation is completed.

This is indeed a good solution. `200` might also work, but I can't imagine a scenario where it's important to keep read permissions off for the owner :)

meetmatt commented on Jun 8, 2019

Setting `x00` rights will break applications that are reading from the source...

chungy commented on Jun 9, 2019

Contributor

Author

Setting `x00` rights will break applications that are reading from the source...

Mind elaborating? We are only talking about the temporary file that is being written to, and setting said file back to the original's mode. This temporary file isn't intended to be read by anything.

chungy mentioned this issue on Jun 9, 2019

[\[programs\] set chmod 600 after opening destination file #1644](#)🔗 Merged

meetmatt commented on Jun 9, 2019

Mind elaborating? We are only talking about the temporary file that is being written to, and setting said file back to the original's mode. This temporary file isn't intended to be read by anything.

I've overlooked that detail, you are right, shouldn't be a problem

 Cyan4973 closed this as completed in [#1644](#) on Jun 13, 2019

carnil commented on Feb 3, 2021

Hi,

While the final file mode is reflective of the input file, when compressing or uncompressing, the file can temporarily gain greater permissions than the input and potentially leading to security issues (especially if large files are being handled).

Example: `file` has mode `600` -> `zstd file` -> `file.zst` has mode `644` while compression is happening. (Same happens with `unzstd file.zst` and `file`)

Would this issue potentially warrant a CVE to be assigned? According to https://cve.mitre.org/cve/request_id.html#cna_participants I guess this would fall into being assigned by the Facebook CNA?

sdelafond mentioned this issue on Feb 11, 2021

[Race condition allows attacker to access world-readable destination file #2491](#)

🔒 Closed

mctaggatart commented on Mar 1, 2021

Hi! also wondering if this would warrant a CVE being assigned. We're considering such at Red Hat. It seems that the first time this issue was reported was:

#1630

and the patch for it has been merged already in 1.4.8 version by this PR:

#1644

Then later, another issue related to the possible access to the temporary files as reported:

#2491

and the fix has been already merged to dev:

a774c57

and will be added to the new zstd release, could be 1.4.9? I dunno.

So then we have this new, but related issue. Do you support Red Hat assigning a CVE, or should we treat it as a bug?


carnil commented on Mar 1, 2021 • edited

@stacifractals I do not think (or let's say suspect) that Red Hat CNA is allowed to assign CVEs here. I was for that in contact with MITRE and it looks that Facebook CNA is responsible to assign CVEs here. We tried to reach out, but unsuccessful yet (because it will need to have a Facebook account for reporting that).

It will for sure need two CVEs. One for the original issue #1630 and one for #2491 as it was incomplete fix for the former.

In case you still consider Red Hat to assign two CVEs, we at least would appreciate to have two, given we released already two advisories (<https://lists.debian.org/debian-security-announce/2021/msg00031.html> and <https://lists.debian.org/debian-security-announce/2021/msg00040.html>), but then please check with MITRE CNA. As said I got clear indication here that the CVE responsibility is here on Facebook CNA.

Hope this helps.

🔍  felixhandte self-assigned this on Mar 1, 2021

felixhandte commented on Mar 1, 2021

Contributor

Hi @stacifractals and @carnil,

I'm discussing with folks internally how to get the ball rolling on issuing CVEs for these. I'll provide updates as that proceeds.

felixhandte commented on Mar 1, 2021

Contributor

We are preparing a release of Zstandard with #2495 which will go out soon as v1.4.9.

We've allocated CVE-2021-24031 for #1630 and CVE-2021-24032 for #2491.

mctaggatart commented on Mar 1, 2021

Thanks so much! I'll update our internal team here.

carnil commented on Mar 2, 2021

We are preparing a release of Zstandard with #2495 which will go out soon as v1.4.9.

We've allocated CVE-2021-24031 for #1630 and CVE-2021-24032 for #2491.

Thank you @felixhandte

felixhandte commented on Mar 5, 2021

Contributor

Final update on this topic: these CVEs have been published.

👍 1

🔗  felixhandte mentioned this issue on Mar 5, 2021

Improve Setting Permissions of Created Files #2525

🔗 Merged

🔗 This was referenced on Jul 25, 2021

Security Alert ChinmayManchanda/9447-Team1#1

🔒 Closed

Security Alert ChinmayManchanda/9447-Team1#3

🔒 Open

Assignees

 felixhandte

Labels

good first issue

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

 [\[programs\]](#) set chmod 600 after opening destination file

6 participants

