Directory traversal vulnerability from libzip #54

○ Open **jiahao42** opened this issue on May 9, 2020 · 13 comments

jiahao42 commented on May 9, 2020 • edited ▾

Issue

Given a crafted zip file containing a file of filename `../../../../../../../../tmp/evil.txt` , zip will extract the file to `/tmp/evil.txt` , while actually it should be extracted to `./tmp/evil.txt` . This vulnerability could allow the attacker to write a file to an arbitrary directory.

How to reproduce

You can try to reproduce this vulnerability using this zip file, note that the symbol `nim -d:useLibzipSrc` is needed for compilation. You can find the PoC here

👍 3

dmknght commented on Jun 13, 2021

The variable `dest` should be filtered
`https://github.com/nim-lang/zip/blob/master/zip/zipfiles.nim#L193`

StayPirate commented on Aug 11, 2021

`CVE-2020-23171` has been assigned to this security bug. Is any patch going to be released anytime soon?

Araq commented on Aug 11, 2021 • edited ▾ Member

Well the hyperbole isn't motivating to look into the issue.

> This vulnerability could allow the attacker to write a file to an arbitrary directory.

The operating system enforces access rights...

❤️ 1

rposkocil commented on Aug 18, 2021

Understood but any progress or what is the plan with this issue? Still open and vulnerability tools marks all versions.

👍 11 👎 1

ajurge commented on Aug 25, 2021

Hi is there any update on this because our builds started failing because of CVE-2020-23171?

👎 1

VaeterchenFrost commented on Aug 26, 2021

Some guidance on remedial actions has been collected, for example, in https://snyk.io/research/zip-slip-vulnerability.

rposkocil commented on Aug 30, 2021

Hi guys, it's false positive. See jeremylong/DependencyCheck#3594.

StayPirate commented on Aug 30, 2021

How can that be a FP if the reported **@jiahao42** attached a reproducer? Has anyone tested it?

prosprice commented on Aug 30, 2021 • edited ▾

The linked DependencyCheck project is doing a poor job matching CPEs and has attributed the CVE for this issue to an unrelated Java library. That's the FP that **@rposkocil** is referring to and what brought me here, but the FP report in *that project* should not be interpreted to mean that this issue is false.

StayPirate commented on Aug 31, 2021

Just to clarify, we can ignore jeremylong/DependencyCheck#3594 (FP is their problem) and keep tracking this **actual** security bug with `CVE-2020-23171` . **@prosprice**, am I correct?

lathspell mentioned this issue on Sep 2, 2021

False Positive on lang-tag-1.5.jar CVE-2020-23171 jeremylong/DependencyCheck#3623

Closed

prosprice commented on Sep 10, 2021

@StayPirate I'm not a Nim user; with that caveat, yes I agree with you.

StayPirate commented on Sep 17, 2021

@Araq could you share some extra information about how the upstream is going to address this issue?

supakeen commented on Sep 22, 2021

I must say that without diminishing the exploitability of this that `nim-lang/zip` isn't in Nim by default and like many other libraries it can be misused.

This is the same for other languages that provide wrappers around libzip's `extractall` , see for example the Python documentation warning you against this possibility: https://docs.python.org/3/library/zipfile.html#zipfile.ZipFile.extractall

What would be the proposed 'fix' for this CVE? Removing the `extractall` binding or documenting that it doesn't handle relative paths?

👍 1

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

9 participants