OS Command Injection in ljharb/npm-lockfile

0

Valid Reported on Feb 28th 2022

Description

npm-lockfile before 2.0.4 does not santize unsafe external input and invoke sensitive command execution API with the input, causing command injection vulnerability.

Proof of Concept

```
// npm i npm-lockfile@2.0.3

const getLockfile = require('npm-lockfile/getLockfile');
getLockfile("./package-lock.json", "08/01/2022",{"only":"prod|touch /tmp/r
```

Impact

This vulnerability is capable of executing arbitrary command on the hosting operating system.

CVE
CVE-2022-0841
(Published)

Vulnerability Type
CWE-78: OS Command Injection

Severity
Low (3.8)

Visibility
Public

Status
Fixed

Found by

Feng Xiao

@xiaofen9

unranked

Fixed by

Jordan Harband

@ljharb

maintainer

This report was seen 808 times.

We are processing your report and will contact the **ljharb/npm-lockfile** team within 24 hours.
9 months ago

We have contacted a member of the **ljharb/npm-lockfile** team and are waiting to hear back
9 months ago

Jordan Harband modified the report 9 months ago

Jordan Harband 9 months ago — Maintainer

This is indeed an issue, which v3+ works around by no longer shelling out to npm.

It feels like 8.8 is too high a score - anything where you'd have to attack yourself is barely a vulnerability at all. When this becomes a CVE, please ensure it is scored accordingly low.

Either way, I've released a fix in v2.0.5 as well, so now v2 is safe from this self-attack. v1 lacks "only" support, and v3 uses Arborist instead of shelling out, so they're both immune. Also note that v2.0.3 is the version that added "only" support (it should have been a minor, oops) so only v2.0.3 and v2.0.4 are vulnerable.

Jordan Harband validated this vulnerability 9 months ago

Feng Xiao has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Jordan Harband marked this as fixed in v2.0.5 with commit bfdb84 9 months ago

Jordan Harband has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ✖

Feng Xiao 9 months ago Researcher

Could you help assign a CVE for this vulnerability?
Credit to: Feng Xiao, Zhongfu Su. Thanks.
@admin

Jamie Slome 9 months ago Admin

Hey Feng, Jordan 👋

We can assign a CVE here, as long as Jordan is happy to do so.

Jordan - regarding the CVSS, I can see that you adjusted the CVSS, but was there a reason you didn't adjust this to 8.8 instead of a lower score before approving? Just trying to see if there was a bug here, or something confusing on the platform?

Jordan Harband 9 months ago Maintainer

Mainly because the score is only calculated from all the buttons, and technically all the button answers are correct (after my adjustment).

I'm not sure what score would be appropriate for a self-attack, but tbh i'd consider it slightly above zero.

I'm fine with there being a CVE - i just don't want it to imply this was more dangerous than it actually is.

Jamie Slome 9 months ago Admin

Thanks for the response Jordan!

I will arrange a CVE, and adjust the severity of the report to a significantly lower score, i.e. ~ 1.0 (Low) severity mark.

Are you happy for me to proceed?

Jordan Harband 9 months ago Maintainer

Yep, thank you!

Jamie Slome 9 months ago Admin

I have adjusted the CVSS to as low as I can get it without changing the meaning of the vector items drastically.

For the record, the previous score was 8.8 (High) and has been reduced to 3.8 (Low). For anyone viewing this report from the CVE record, please see the maintainer's (Jordan) comments above on security impact.

`CVE-2022-0841` assigned and published! 🎊

Jordan Harband 9 months ago Maintainer

The CVE says "prior to v2.0.5", but the only vulnerable versions are v2.0.3 and v2.0.4. Can you edit the description to be more precise?

Jamie Slome 9 months ago Admin

Adjustments made here 👍

> Sign in to join this conversation