Bitwarden Statement on Checkmarx Supply Chain Incident

eck (Adam Eckerle) 1 April 23, 2026, 2:58pm

The Bitwarden security team identified and contained a malicious package that was briefly distributed through the npm delivery path for @bitwarden/cli@2026.4.0 between 5:57 PM and 7:30 PM (ET) on April 22, 2026, in connection with a broader Checkmarx supply chain incident.

Was I affected?
If you use the Bitwarden command line interface and deploy using NPM, and downloaded the CLI between 5:57p ET and 7:30p ET on April 22, 2026, you may be affected. See remediation steps below.

If you do not use the Bitwarden command line interface you were not affected.

The investigation has currently found no evidence that end user vault data was accessed or at risk, or that production data or production systems were compromised.

What happened?

The investigation has currently found no evidence that end user vault data was accessed or at risk, or that production data or production systems were compromised. Once the issue was detected, compromised access was revoked, the malicious npm release was deprecated, and remediation steps were initiated immediately.

The issue affected the npm distribution mechanism for the CLI during that limited window, not the integrity of the legitimate Bitwarden CLI codebase or stored vault data.

Users who did not download the package from npm during that window were not affected. Bitwarden has completed a review of internal environments, release paths, and related systems, and no additional impacted products or environments have been identified at this time. A CVE for Bitwarden CLI version 2026.4.0 is being issued in connection with this incident.

What should impacted users do?

Only users who installed Bitwarden CLI 2026.4.0 via NPM during the affected window are impacted. If you are an affected user, Bitwarden recommends the following actions

1. Immediately uninstall Bitwarden CLI 2026.4.0 via npm

- `npm uninstall -g @bitwarden/cli`

2. Clear the npm cache

- `npm cache clean --force`

3. Temporarily disable npm install scripts during cleanup as a precaution

- `npm config set ignore-scripts true`

4. Review the additional indicators and cleanup steps outlined by JFrog **here**
5. Rotate any secrets that may have been exposed on the affected system or stored in environment variables including API tokens and SSH keys (examples listed **here** & **here**)
6. Review GitHub activity, CI workflows, and related credentials for unauthorized access or changes
7. Install **Bitwarden CLI 2026.4.1**

Bitwarden is in the process of completing a full review and will implement mitigation to prevent such attacks in the future.

12 Likes

@bitwarden/cli:2026.4.0 infected with malware?

---

grb 3 April 23, 2026, 3:04pm

Thanks for the update!

What steps are being taken to prevent similar issues in the future? **This comment** on Github outlines some steps for hardening distribution via npm:

> to prevent a user with write access to be able to directly trigger a publish with npm oidc, a few steps have to be done
>
> 1. use a publish environment and in that environment set up a branch rule to limit it to one or multiple specific release branches. List every branch separately and only keep active release branches. do not use a pattern that would allow creating a new matching branch
> 2. that environment must be configured on the npm package
> 3. the release branches must be protected against unreviewed >pushes (require pull request with at least 1 review)
> 4. add a mandatory approval step to the publish environment (this can theoretically be skipped if you trust the combination of review and environment lock above, but given the scope of bitwarden i'd recommend having it)

And this **comment on Xitter** suggests that there is a broader vulnerability of CI/CD pipelines:

> CI/CD pipelines as the attack vector for supply chain compromise is becoming the pattern. Malicious workflows can bypass every code review process. The publish step is the weakest link.

Are there any plans to audit and harden publishing pipelines for the other clients?

5 Likes

---

grb 4 April 23, 2026, 4:18pm

To put this in context, it seems that only 334 Bitwarden users downloaded the malicious version of the CLI:

Version History

☑ show deprecated versions

Version	Downloads (Last 7 Days)	Published
2026.4.0	334	18 hours ago

4 Likes

TheBestPessimist 5 April 23, 2026, 4:53pm

Can you update OP and explain how was the package published?

Nail1684 6 April 23, 2026, 4:54pm

> eck:
>
> A CVE for Bitwarden CLI version 2026.4.0 is being issued in connection with this incident.

To make the “version history” here a bit more explicit: Bitwarden never released any (valid) Bitwarden CLI version 2026.4.0. The latest CLI version was 2026.3.0 until Bitwarden CLI 2026.4.**1** was released about two hours ago. (see **Releases · bitwarden/clients · GitHub** for all official client releases)

2 Likes

eck (Adam Eckerle) 7 April 23, 2026, 4:55pm

More details will be part of the CVE that has been submitted.

1 Like

TiTwo102 8 April 23, 2026, 5:04pm

NPM, CLI, checkmarx… ?

Could you please dumb it down and explain if the basic user is at risk ? aAnd if so, what to do ?

---

eck (Adam Eckerle) 9 April 23, 2026, 5:07pm

> TiTwo102:
>
> NPM, CLI, checkmarx… ?
>
> Could you please dumb it down and explain if the basic user is at risk ? aAnd if so, what to do ?

No Bitwarden vault data was compromised. Regular Bitwarden users are not affected. The only users affected are those who downloaded and attempted to use Bitwarden CLI 2026.4.0.

KeronCyst 10 April 23, 2026, 5:34pm

So it sounds like it was 2 hours from release to detection. That's really good. I guess we'll read more later about how this happened.

---

grb 11 April 23, 2026, 5:46pm

> TiTwo102:

> Could you please dumb it down and explain if the basic user is at risk ?

Unless you were one of the 334 Bitwarden users who updated the **Bitwarden Command Line Interface** software package (a tool for developers) from version 2026.3.0 to version 2026.4.0 sometime last night (April 22) between ~~5:57~~ 5:22 PM and 7:30 PM (ET), then you don't have to worry about any security risk resulting from this incident.

If you are not familiar with the CLI, then this is not applicable to you, and you don't have to worry.

8 Likes

DenBesten 12 April 23, 2026, 6:42pm

> KeronCyst:
>
> 2 hours from release to detection. That's really good.

Even better than that. *Less than* 2 hours (93 minutes) from release to *mitigation* (containing the fire).

1 Like

Using AI to identify vulnerabilities and bugs

---

sj-bitwarden 13 April 23, 2026, 7:13pm

Hi all, please note that we will continue to update the top level post in this thread as additional updates and/or information becomes available.

5 Likes

---

mutilator 14 April 23, 2026, 7:19pm

Will the release paths be better checked and secured in the future?

elmoextremist 15 April 23, 2026, 8:24pm

Did the snap package get generated from npm during the time that the npm package was compromised?

1 Like

---

eck (Adam Eckerle) 16 April 23, 2026, 8:49pm

The snap package was not affected.

2 Likes

---

grb 17 April 23, 2026, 9:39pm

For the technically inclined, some additional detail has been provided in comments by Bitwarden staff member **/u/mandreko** on Reddit:

https://old.reddit.com/r/Bitwarden/comments/1stkc46/bitwarden_cli_has_been_compromised_check_your/ohueado/

1 Like

Neuron5569 18 April 24, 2026, 12:36am

> eck:
>
> between 5:57 PM and 7:30 PM (ET) on April 22, 202

socket.dev seems to report that the malicious package was published at 4/22/2026, 9:22:59PM UTC, i.e. 5:22PM ET, matching what **the Github OP posted**. This may be confusing when compared with the current official statement.

1 Like

BitSimp 19 April 24, 2026, 3:00am

Good question. @eck I'm curious if this is part of the discussion too.

Beyond hardening the existing CI/CD pipeline, are more architectural mitigations being considered, like stronger multi-approver controls or anchoring parts of release integrity and artifact verification in more tamper-resistant environments?

For example, tools like Orbit seem interesting for policy-driven approvals, multi-approver controls, and auditable "four-eye" workflows around sensitive updates:
https://orbit.global/

I've also been looking at whether canister-based systems on the Internet Computer could play a role in anchoring parts of the publishing trust chain or verification process. There's documentation on canister smart contracts on the Internet Computer site for anyone curious to explore that angle further.

Genuinely curious whether approaches like these are being evaluated as part of strengthening the weak points supply chain attacks keep targeting, and if not, what tradeoffs may make them difficult to implement.

1 Like

Neuron5569 20 April 24, 2026, 4:19am

> grb:
>
> ersion 2026.4.0 sometime last night between 5:57 PM and 7:30 PM (ET)

Also, since the release reportedly didn't appear on Github, most likely ONLY people updating the package via npm would be affected.

---

next page →