contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

hefy2027 cf-manager: new CVE

CVE-2026-71499 · record published 18 November 2026

Analysis Report

If you run cf-manager, move to 1.1.2 this week and check that API_SECRET is set before you do anything else. If you do not run cf-manager, you can stop reading: nothing here concerns you.

PREPARED BY	NamiQ	DATE	19 November 2026
FACTS AS OF	19 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 10.0**	STATUS	Not on CISA's exploited list; exploitation not recorded

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/hefy2027-cf-manager-cve-2026-71499

The bottom line

> *If you run cf-manager, move to 1.1.2 this week and check that API_SECRET is set before you do anything else. If you do not run cf-manager, you can stop reading: nothing here concerns you.*

At a glance

Fact	Detail
Product	hefy2027 cf-manager (CVE-2026-71499)
Severity	Critical: 10 (CVSS 3.1)
Status	A new CVE record, published by GitHub on 18 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 19 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	cf-manager 1.1.2. Both the CVE record and the project's release notes name it as the fixed version.
Who looks first	Anyone who runs the cf-manager dashboard for Cloudflare accounts, especially where its API can be reached by people you do not fully trust, or where API_SECRET was never set.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you or your IT provider run the cf-manager dashboard for Cloudflare accounts?	Nothing to do for this CVE.	Check the version (next question).
Is the version older than 1.1.2?	Nothing further for this CVE. Still check API_SECRET (next question).	Plan the update to 1.1.2 and read the next question first.
Is API_SECRET set in the environment cf-manager runs in?	Treat it as open to anyone who can reach it. Restrict access now, then update.	Only people holding the secret can use the deploy API. Update anyway.
Can anyone untrusted reach the cf-manager API from the internet or a shared network?	Lower exposure, but still update to 1.1.2.	Restrict access first, then update. Treat it as urgent.

What it is

The record says cf-manager fetches a user-supplied URL with no checks and deploys whatever comes back as a Worker script, using stored Cloudflare credentials. An attacker can reach internal resources or run their own JavaScript in your Cloudflare Workers account.

Why it matters

Our reading is that the danger is the stored Cloudflare credentials. The flaw turns the dashboard into a way to publish attacker-chosen code into your own Workers account, with the account's own authority. The record also says unauthenticated access is possible when API_SECRET is not configured. The release notes say earlier versions skipped authentication silently in that case.

What the vendor's advisory says

We read two pages on github.com. In short:

- The release notes say 1.1.2 adds a safe fetch function that replaces three bare fetch calls in Worker deployment and batch deployment.
- The notes say the new function allows only https, and refuses loopback, private, link-local and unique-local addresses.
- The notes say redirects are checked at each hop, only JavaScript or text content types are accepted, and responses are capped at 5 MiB.
- The notes say that if API_SECRET is not set, the app now generates a random temporary secret and prints a security warning to the console.
- The notes say the audit log for Worker deployment now records the source URL, and credit the researchers who reported the flaw.

Things people will get wrong

- "It needs a login, so we are safe." The record says unauthenticated access works when API_SECRET is not configured. Check that it is set.
- "Updating is enough; nothing could have happened before." The record does not say whether anyone used this flaw. Check your Workers for scripts you did not deploy.
- "This only affects Cloudflare itself." The flaw is in the cf-manager dashboard. The code runs in the victim's own Cloudflare Workers account.

How we would triage it

This is our judgement.

The commands only read: they print a version or list what is installed. They come from general knowledge of this kind of product, not from the sources, so check them against the vendor's page

before you run them. Words in capitals are for you to fill in.

Step 1: Find where cf-manager runs and its version

- Ask whoever manages your Cloudflare accounts whether cf-manager is used, and where it is hosted.
- Note the running version. Anything older than 1.1.2 is the one to fix.

```
docker ps --format '{{.Names}} {{.Image}}'
```

Step 2: Reduce exposure before updating

- Make sure API_SECRET is set to a strong value, because the record says it is the difference between an open and a closed deploy API.
- Keep the dashboard off the public internet until it is updated.

Step 3: Update to 1.1.2

- Move to 1.1.2, the version both the record and the release notes name as fixed.
- Consider rotating the Cloudflare credentials cf-manager holds if it was reachable while unprotected.

Step 4: Check your Workers account

- Look for Worker scripts you did not deploy, and review the deployment audit log.
- The audit log only records source URLs from 1.1.2 on.

To close it, read the version from the product itself, not from an old inventory, and save the output with the date. If you cannot update today, write down who owns it, the new date, and what limits exposure until then.

Send this to your IT provider

Copy it, change the names and dates, and send it. It is a draft for you to check, not a template tested with any provider.

> *We would like to know whether we run hefy2027 cf-manager, and where. It is affected by CVE-2026-71499 in versions before 1.1.2. Please tell us which version is running, whether API_SECRET is set, and whether the API can be reached from outside our network. The fixed version is 1.1.2. Please confirm in writing the version and that the fix is in place by a date you propose. Please also check our Cloudflare Workers for scripts we did not deploy.*

The weakness behind it

The record classes this as CWE-94 and CWE-918.

Likely ATT&CK mapping

MITRE ATT&CK is a public catalogue of the steps attackers take, from getting in to causing damage. No source has mapped this flaw to it yet, so the steps below are our prediction from what the record says the flaw allows. They are not MITRE's, the vendor's or CISA's, and an attacker may do other things.

Steps an attacker would likely take with this flaw

The step, in plain words	Stage of an attack	Why we think so
Exploit Public-Facing Application (T1190): Breaking in through a service that faces the internet, such as a web server, VPN or login page.	Initial Access: getting in	The deploy API is a web endpoint, and the record says it may need no login when API_SECRET is unset. That would likely make it reachable by anyone who can connect.
Ingress Tool Transfer (T1105): Downloading more attacker tools onto the machine.	Command and Control: talking to the attacker	The flaw fetches JavaScript from a URL the attacker chooses and deploys it, so attacker-supplied code would typically be brought onto the victim's Cloudflare account.

How to use this: ask your IT provider which control or log would show each step, for example whether sign-in attempts from the internet are recorded. If nobody can answer, that is the gap to close first.

MITRE ATT&CK® is a registered trademark of The MITRE Corporation. Technique names are MITRE's.

The sources, in their words

The CVE record:

> *cf-manager is an all-in-one dashboard for managing multiple Cloudflare accounts and services. Prior to 1.1.2, worker/src/routes/workers.ts uses bare fetch(url) calls to retrieve user-supplied URLs and deploys the returned content as a Worker script using stored Cloudflare credentials. The multipart and JSON branches of POST /api/:accountId/workers and POST /api/batch-deploy accept URLs without protocol, destination, redirect, content-type, or response-size validation. An attacker with deployment API access, or unauthenticated access when API_SECRET is not configured, can make cf-manager access internal resources or fetch attacker-controlled JavaScript and execute it in the victim Cloudflare …*

Versions of hefy2027 cf-manager the record lists as affected:

- < 1.1.2

The record gives no version ranges. Use the vendor's advisory for the affected versions.

Check your own project

If you build software, you can compare what a project declares with this CVE from a terminal. The tool reads requirements files, package.json, Maven, Gradle, Cargo, CMake and Dockerfile files, runs nothing, and only the CVE id leaves your machine. Naming the CVEs needs no key.

```
lattice-scan project . --cve CVE-2026-71499
```

Each row says in-range, outside-range, undetermined or lead, with the file and line. That is evidence about what a manifest declares, not about what runs, and a person decides. The verdicts, the CSV columns and a pipeline recipe are in the guide.

Check the servers you run

The same id can be checked against a host. Run this on the machine itself, not on the laptop that deploys to it: it reads installed packages (dpkg, rpm or apk), snaps, the hardware (maker, model, firmware), the running container images with their operating system and packages, and the kernel version, compares them on the spot, and only the CVE ids leave the machine; the inventory stays on it, in ~/.lattice-scan. Naming the CVEs needs no key. Do it on each server after a deployment, and again when a new entry lands.

```
lattice-scan --cve CVE-2026-71499
```

A match is a candidate for a person to confirm, not a finding. A distribution can fix a CVE under an unchanged version number, so a match is a candidate; on Debian-family hosts the report also shows

whether the package's own changelog names the CVE. By default it opens the images your containers are running and reads their operating system and package list; a deep scan (--deep, a few minutes) opens every local image and the Python, Node, Java and Ruby packages inside them; for what an image declares, use the project check on its Dockerfile.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

What we do not know yet

- Whether anyone has used this flaw. The record gives no sign of it.
- Whether any version before 1.1.2 is exempt. The record lists everything below 1.1.2 as affected.
- Whether the temporary secret in 1.1.2 is enough if you leave API_SECRET unset. The notes only say a warning is printed.
- Exploitation is not recorded. That is not the same as none.
- This is a new record: scores, versions and wording can change in the first days, and they are as at 19 November 2026. We read the vendor pages listed under Sources; nothing here has been tested.
- The sections headed "The bottom line", "Does this apply to you?", "What it is", "Why it matters", "Exposure by situation", "Things people will get wrong", "Likely ATT&CK mapping", "How we would triage it" and "Send this to your IT provider" are our interpretation of the sources, not a statement by CISA or the vendor.

Appendix 1. Severity decoded

CVE-2026-71499: score 10, `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` As the CVE record gives it.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	Low	No special conditions beyond the flaw itself.
Privileges required	None	No account is needed.
User interaction	None	No one has to do anything.
Scope	Changed	Damage reaches beyond the vulnerable component.
Confidentiality	High	Total loss of confidentiality.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
18 November 2026	CVE record published by GitHub.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe
cf-manager older than 1.1.2, API_SECRET not set, reachable from the internet	High	Now: restrict access today, then update
cf-manager older than 1.1.2, reachable from a network with users you do not fully trust	High	This week
cf-manager older than 1.1.2, API_SECRET set, reachable only by trusted administrators	Medium	Next patch cycle, soon
cf-manager at 1.1.2 or later	Low	Confirm API_SECRET is set
You do not run cf-manager	None	None for this CVE

Appendix 4. Definition of done

- Every hefy2027 cf-manager you run is listed, with its version.
- Each is outside the affected list, or the vendor's fix or mitigation is in place.
- The version output or a screenshot is saved with the date.
- Anything not done has an owner and a date.

Sources

- CVE record for CVE-2026-71499: https://www.cve.org/CVERecord?id=CVE-2026-71499
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/hefy2027/cf-manager/security/advisories/GHSA-wm: https://github.com/hefy2027/cf-manager/security/advisories/GHSA-wmx3-qxcc-mh64
- Advisory or release notes read: github.com/hefy2027/cf-manager/releases/tag/1.1.2: https://github.com/hefy2027/cf-manager/releases/tag/1.1.2
- Reference in the CVE record: github.com: https://github.com/hefy2027/cf-manager/commit/934aecbe614d8a08a8c96f393c0126519c35e3be