contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

flytohub flyto-core: new CVE

CVE-2026-77312 · record published 19 November 2026

Analysis Report

If you do not run Flyto2 Core (the flyto-core package), you can stop here. If you do, plan an update to version 2.26.8 without delay, and first list where it runs and which version each host has.

PREPARED BY	NamiQ	DATE	20 November 2026
FACTS AS OF	20 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/flytohub-flyto-core-cve-2026-77312

The bottom line

> *If you do not run Flyto2 Core (the flyto-core package), you can stop here. If you do, plan an update to version 2.26.8 without delay, and first list where it runs and which version each host has.*

At a glance

Fact	Detail
Product	flytohub flyto-core (CVE-2026-77312)
Severity	Critical: 10 (CVSS 3.1)
Status	A new CVE record, published by GitHub on 19 November 2026. Not on CISA's list of exploited vulnerabilities when we read it on 20 November 2026, which means nothing public says yet that anyone is using it.
Fixed in	flyto-core 2.26.8 (PyPI); the vendor's release notes call it a security release fixing this arbitrary file write and four other advisories.
Who looks first	Anyone running flyto-core below 2.26.8, above all where untrusted users can submit modules or recipes or reach its HTTP endpoint. It is a developer automation and AI-agent tool, so most small firms will not run it.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run Flyto2 Core (the flyto-core Python package) anywhere?	Nothing to do for this CVE.	Check the version next.
Is that version below 2.26.8?	You are on a fixed release; nothing to do.	Plan the update to 2.26.8.
Can untrusted users run modules or recipes, or reach its HTTP endpoint?	Lower risk, but still update.	Treat the update as urgent.

What it is

The record says someone able to run the product's modules or recipes can write files outside the intended sandbox folder. Several data and file modules do not confine the output path, and the path check does not reject absolute paths, so an attacker can choose where a file is written.

Why it matters

An arbitrary file write outside the sandbox is a strong primitive. Our reading is that it can become more than a stray file: overwriting a configuration or a file the process later loads can lead to code running. The record says this is an incomplete fix of an earlier remediation, so a partial patch left four named modules, and others in the data, file and report-writer families, still exposed. The block on '..' did not reject absolute paths, which is why a server thought to be fixed may not be.

What the vendor's advisory says

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

- The vendor's release notes say 2.26.8 also fixes a second-order SSRF in http.paginate (GHSA-2mr3-rxrq-238c, High) via an unvalidated Link rel=next header, now revalidated.
- They also report two DNS-rebinding SSRF flaws (GHSA-pfg2-w999-497v, High; GHSA-6pm8-6f34-9v3g, Medium); outbound modules now pin the validated IP.
- They also fix a path traversal in the MCP run_recipe feature (GHSA-mxcc-cr6x-2mvr, Medium); recipe loading is now confined to the bundled recipes directory.
- The notes say all five advisories are confined to versions 2.26.7 and earlier, and all eight file-writing modules now confine to the sandbox directory.
- A separate advisory describes unauthenticated command execution via the HTTP MCP endpoint in versions 2.26.2 to before 2.26.4 (High, fixed in 2.26.4); it is a different flaw from this one.

Things people will get wrong

- It only writes inside the sandbox, so it is contained. The record says four named modules write outside FLYTO_SANDBOX_DIR, and the block on '..' does not reject absolute paths.
- We patched the MCP flaw at 2.26.4, so we are covered. That fixed a different flaw (CVE-2026-55786). This file-write is fixed in 2.26.8 per the vendor's release notes.
- Only the HTTP endpoint is at risk. The write is in the data and file modules; any caller able to run those modules or recipes can trigger it, not only HTTP callers.

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 the versions you run

- List the installed flyto-core version on every host that runs it; the record says every release below 2.26.8 is affected.

```
pip show flyto-core
pip list | grep -i flyto
```

Step 2: Reduce who can reach it

- Until updated, limit who can submit or run modules and recipes, and restrict network access to its HTTP endpoint, since a caller able to run modules is all the flaw needs.

Step 3: Update to the fixed release

- Move flyto-core to 2.26.8; the vendor's release notes name it as the release that confines all file-writing modules to the sandbox.

Step 4: Confirm the version afterwards

- After updating, re-check that each host shows 2.26.8 or later, so no instance is left behind.

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.

> *Hello, we need help with a security issue. A path-traversal flaw (CVE-2026-77312) in Flyto2 Core, the flyto-core Python package, lets a caller who can run its modules or recipes write files outside the intended sandbox folder. It affects every version below 2.26.8. Please confirm in writing whether we run flyto-core, on which hosts, and which version each runs. If any is below 2.26.8, please update to 2.26.8, which the vendor's release notes give as the fix, and meanwhile limit who can reach its module execution and HTTP endpoint. Please send written confirmation of the versions and the update by the end of this week. Thank you.*

The weakness behind it

The record classes this as CWE-22.

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 HTTP endpoint dispatches requests to registered modules, so an attacker could likely reach the vulnerable write modules through the public-facing API.
Hijack Execution Flow (T1574): Making a program load the attacker's code instead of the real one.	Persistence: staying in	An arbitrary write outside the sandbox could overwrite a file the product or host later loads, which would typically let planted content run.
Escape to Host (T1611): Breaking out of a container to reach the machine that hosts it.	Privilege Escalation: gaining higher rights	Writing outside the sandbox directory breaks its confinement boundary, which would likely let an attacker affect the host beyond the sandbox.

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:

> *Flyto2 Core is an execution kernel for automation and AI-agent workflows. Prior to 2.26.8, the earlier CVE-2026-67429 remediation applied validate_path_with_env_config() to only a hand-picked set of file-writing modules, leaving data.csv.write in src/core/modules/atomic/data/csv_write.py, data.json_to_csv in src/core/modules/atomic/data/json_to_csv.py, file.copy in src/core/modules/atomic/file/copy.py, and file.move in src/core/modules/atomic/file/move.py outside the FLYTO_SANDBOX_DIR confinement boundary. The advisory also identifies unguarded modules across the broader data.*, file.*, and report-writer families. data.csv.write relies on a .. substring denylist that does not reject …*

Versions of flytohub flyto-core the record lists as affected:

- < 2.26.8

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-2026-77312
```

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-2026-77312
```

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 this flaw has been exploited anywhere is not recorded.
- Which further modules in the data, file and report-writer families remain affected beyond the four named.
- Whether any workaround short of upgrading exists; the release notes give none for this flaw.
- Whether a default install exposes module or recipe execution to unauthenticated callers in this version range.
- 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 20 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-2026-77312: score 10, `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/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	None	No data exposed.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
19 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
A flyto-core server below 2.26.8 whose HTTP endpoint or module execution is reachable by untrusted users	High	this week
flyto-core below 2.26.8 run locally for trusted automation only	Medium	next patch cycle
flyto-core already on 2.26.8 or later	None	none for this CVE

Appendix 4. Definition of done

- Every flytohub flyto-core 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-2026-77312: https://www.cve.org/CVERecord?id=CVE-2026-77312
- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- Advisory or release notes read: github.com/flytohub/flyto-core/security/advisories/GHSA-p3: https://github.com/flytohub/flyto-core/security/advisories/GHSA-p34x-fmph-9fjx
- Advisory or release notes read: github.com/flytohub/flyto-core/releases/tag/v2.26.8: https://github.com/flytohub/flyto-core/releases/tag/v2.26.8
- Advisory or release notes read: github.com/advisories/GHSA-h9f9-h6gm-wc85: https://github.com/advisories/GHSA-h9f9-h6gm-wc85
- Reference in the CVE record: github.com: https://github.com/flytohub/flyto-core/commit/bbb6f1cfec22fee1ea0e6ecf1129d7103c85e09f