

Backdoored Cemu release linked to TanStack and Mistral supply chain campaign



securitylabs.datadoghq.com/articles/backdoored-cemu-release-teampcp-supply-chain-campaign

May 14, 2026



On May 11th a coordinated supply chain campaign poisoned 170 legitimate packages across npm and PyPI within a five-hour window. SafeDep documented the notable victims, malicious versions of `@tanstack/react-router` (3M+ weekly downloads) pushed through compromised npm tokens alongside a direct commit to the upstream GitHub repo (79ac49eed), and a backdoored `mistralai==2.4.6` slipped onto PyPI with no corresponding upstream tag.

While investigating the `mistralai` compromise, we discovered that `transformers.pyz` a payload embedded in the malicious wheel was also associated with releases on the `cemu-project/Cemu` GitHub release page, a Nintendo Wii U emulator with no obvious relationship to Python ML SDKs.

While investigating `transformers.pyz` on VirusTotal, pivoting on its SHA256 led us to a second binary bundling the same file: `Cemu-2.6-x86_64.AppImage`. We confirmed this was the image hosted on the official `cemu-project/Cemu` GitHub release page.

Cemu 2.6

Latest

Compare

github-actions released this Feb 6, 2025 · 213 commits to main since this release v2.6 a6fb0a4

Changelog:

- Exporting saves via the title manager no longer crashes when the export path contains unicode characters
- Fixed a regression in DS Virtual Console titles (#1440)
- Fixed a bug where the debug shader dumping option would try to write files to the wrong directory
- Added a right-click context menu to the PPC debugger disassembly view and fixed some smaller bugs in the debugger
- Use libusb as the backend for HID/USB devices on Windows (#1471)
- Reduce amount of Vulkan sampler objects that are created. Reduces memory usage slightly and avoids crashes on drivers that have a low limit on sampler objects
- Improved index data caching on Vulkan. This may provide a very small performance benefit in some games
- The `--force-interpreter` command line option now always uses single-threaded CPU emulation like originally intended

Assets

- cemu-2.6-macos-12-x64.dmg 34.9 MB Feb 6, 2025
- cemu-2.6-ubuntu-22.04-x64.zip sha256:f140e76236b96adf7cdc796227... 30.4 MB 5 days ago
- cemu-2.6-windows-x64.zip 25.2 MB Feb 6, 2025
- Cemu-2.6-x86_64.AppImage sha256:d87a29c4458d08e42d5d9e6345... 62.6 MB 4 days ago
- Source code (zip) Feb 4, 2025
- Source code (tar.gz) Feb 4, 2025

200 57 52 81 42 45 272 people reacted

The Cemu v2.6 GitHub release page showing backdoored Linux assets re-uploaded by MangelSpec alongside the original macOS and Windows builds

Cemu-2.6-x86_64.AppImage is a single-file Linux application bundle; under the hood it is a SquashFS image with a small ELF prefix. Extracting it laid out the bundled filesystem under `squashfs-root/`. The interesting artifact lives at `squashfs-root/usr/share/Cemu/scripts/startup.pyz` — a path that does not exist in the legitimate Cemu build. There is no `scripts/` directory in upstream Cemu and no Python interpreter in its dependency graph; the file is foreign to the project.

`startup.pyz` is a Python zipapp: a renamed zip archive with a `__main__.py` entrypoint, designed to be executed by the system Python interpreter. Its sha256 (`0f35abda19fb69430c32228465396094b866d887427bf551e353ab31256a9dd6`) indicates byte-for-byte identical to the `transformers.pyz` distributed through the TanStack npm campaign.

Unzipping `startup.pyz` reveals the campaign's working set:

Date	Time	Attr	Size	Compressed	Name
2026-05-12	09:21:46	D....	0	0	python_mistral_cemu_fi
2026-05-05	06:02:52	D....	0	0	python_mistral_cemu_fi
2026-05-03	04:24:52		0	12	python_mistral_cemu_fi
2026-05-07	00:38:54		1565	568	python_mistral_cemu_fi
2026-05-05	05:56:18		2411	846	python_mistral_cemu_fi
2026-05-06	21:25:52		1579	554	python_mistral_cemu_fi

2026-05-12 09:21:46		6035	2582	python_mistral_cemu_fi
2026-05-07 17:07:42		2457	1150	python_mistral_cemu_fi
2026-05-12 09:20:38		897	409	python_mistral_cemu_fi
2026-05-07 00:39:02		3729	1335	python_mistral_cemu_fi
2026-05-05 06:02:42	D.....	0	0	python_mistral_cemu_fi
2026-05-07 00:30:36		10499	2617	python_mistral_cemu_fi
2026-05-05 04:21:48		11943	3172	python_mistral_cemu_fi
2026-05-07 00:32:20		4224	1445	python_mistral_cemu_fi
2026-05-06 21:22:46		7274	1953	python_mistral_cemu_fi
2026-05-03 04:24:58		6370	1891	python_mistral_cemu_fi
2026-05-03 22:43:44		3930	947	python_mistral_cemu_fi
2026-05-03 04:10:34		0	12	python_mistral_cemu_fi
2026-05-03 04:24:58		7803	2178	python_mistral_cemu_fi

2026-05-12 09:21:46		70716	21671	16 files, 3 folders

The npm and PyPI legs of this campaign were credential-driven: stolen publishing tokens used to ship malicious versions of legitimate packages. The GitHub Releases leg follows the same playbook in a third ecosystem — the poisoned AppImage was hosted on the real Cemu release page, on the real tag, under the real filename. Replacing it required write access to the actual `cemu-project/Cemu` repository.

GitHub's REST API exposes more about release assets than the UI shows. Every asset record carries its own uploader, `created_at`, `digest`, and download count. One call returns the smoking gun:

```
gh api repos/cemu-project/Cemu/releases/tags/v2.6
```

File	Uploader	Uploaded (UTC)	sha256	Downloads
cemu-2.6-macos-12-x64.dmg	github-actions[bot]	2025-02-06	(original)	279,600
cemu-2.6-windows-x64.zip	github-actions[bot]	2025-02-06	(original)	2,163,866
cemu-2.6-ubuntu-22.04-x64.zip	MangelSpec	2026-05-07	f140e762...327cc	1,957
Cemu-2.6-x86_64.AppImage	MangelSpec	2026-05-08	d07a29c4...b41a	19,897

The macOS and Windows assets carry their original 2025-02-06 bot uploads. The two Linux assets were deleted and re-uploaded over a year later by a user

account named MangelSpec. Linux-only and the AppImage in particular gave the campaign 19,897 downloads of blast radius before anyone noticed.

Aggregating every release back to v2.0-17 (November 2022), every single asset had been uploaded by `github-actions[bot]` only. **v2.6 is the first release in the repo's history with a non-bot uploader on any asset.**

The rest of the repo's state is intact:

- The v2.6 git tag still points at commit `a6fb0a48eb437a8a41c13b782ac8ae0433bf8f98`, unchanged since the original release.
- No new commits on `main` in the surrounding window.
- No edits to `.github/workflows/`.
- And of the 44 workflow runs in `cemu-project/Cemu` in May 2026, **zero** are `Deploy release`.

The swap did not go through CI. The Releases API was called directly, by a credential authenticating as a user, without any workflow involvement. That rules out the most common stolen-credential class in OSS attacks (`GITHUB_TOKEN` exfiltrated from a malicious dependency during a CI run) because `GITHUB_TOKEN` only exists during a workflow run, and no workflow ran. The credential lives further up the stack, on a human account.

We quickly alerted the maintainers about this:

GitHub issue filed to alert the Cemu maintainers of the compromised release assets, detailing the two affected files and their upload metadata

(link to the issue in the project repository)

And we got confirmation that MangelSpec was a long term co-author, whose GitHub account or token was compromised.

Cemu maintainer Exzap confirms MangelSpec is a long-term co-author and announces removal of their repo access pending investigation

While the AppImage received the most downloads (19,897), the Ubuntu zip (`cemu-2.6-ubuntu-22.04-x64.zip`) contains some subtle variation. Both files were uploaded by MangelSpec within hours of each other on 2026-05-07 and 2026-05-08. Both bundle a Python zipapp named `startup.pyz`. The file listings inside are identical in structure. But their `__main__.py` entry points are not the same binary.

AppImage: `c234e6e5f5366cf1a703e62a995a7a0511583ca703c7c9d0590965342eb4c4e8`

The Applmage variant is the simpler of the two. It applies three anti-analysis filters — Linux-only, non-Russian locale, CPU count above four — then silently installs the cryptography package if absent, and immediately runs the entrypoint module.

The Ubuntu variant passes through the same three filters, then diverges. Three things are added:

```
from pathlib import Path

INIT_MARKER = Path('/tmp/.transformers')

if __name__ == "__main__":
    if not INIT_MARKER.exists():
        try:
            INIT_MARKER.touch()
        except Exception:
            pass
        sys.exit(1)
    else:
        # ... install cryptography, run entrypoint ...
```

On the first execution the marker file does not exist. The script creates it and calls `sys.exit(1)` — it appears to crash silently and do nothing. On every subsequent execution the marker is present, and the real payload runs normally. We surmised that the intention was to evade detection in automated sandbox environments.

The payload's behaviour is consistent with what other publications have already documented. It targets Linux hosts only, exiting early if Russian language settings are detected or the CPU count falls below four. Persistence is achieved by decoding a base64 blob, writing it to disk as `pgmonitor.py` — masquerading as a PostgreSQL monitoring tool — and installing it as a systemd service (`pgsql-monitor.service`).

A geofenced destructive component also exists: systems geolocating to Israel or Iran have a 1-in-6 chance of triggering forced audio playback and deleting files from the host's filesystem with `rm -rf /`. At least one user has reported being impacted by this file erasing behaviour. However it is worth mentioning that without the `--no-preserve-root` flag this command will not execute successfully on modern linux systems. In updated versions of this payload which did not impact Cemu the command was updated to `rm -rf /*`.

Credential harvesting is broad, covering AWS (Secrets Manager, SSM), Azure Key Vault, GCP Secret Manager, Kubernetes secrets across all namespaces,

and filesystem credentials including SSH keys, GitHub tokens and password managers (1Password, Bitwarden, gopass, pass).

Exfiltration follows a tiered fallback process: results are POSTed to 83[.]142.209[.]194/v1/weights first, then a dead-drop resolver that polls public GitHub commits for RSA-signed C2 addresses matching the pattern FIRESALE <base64url>.<base64sig>, and finally — if both network paths fail — a new public GitHub repository is created under the victim's own account using a randomised Slavic-mythology name, with results.json uploaded directly.

A group identifying itself as TeamPCP has publicly claimed responsibility for this campaign based on the screenshots/text provided. We are not in a position to assess the credibility of that claim, and we do not assign any confidence level to the attribution.

The C2 server at 83.142.209.194 displaying a message from TeamPCP claiming credit for the campaign

LICENSE file in python decompressed output:

```
MMP""MM""YMM                                     `7MM""Mq.   .g8""bgd `7MM""M
P'   MM   `7                                     MM   `MM..dP'   `M   MM
      MM   .gP"Ya   ,6"Yb.   `7MMpMMMb.pMMMb.   MM   ,M9 dM'   `   MM
      MM ,M'   Yb 8)   MM   MM   MM   MM   MMmmdM9   MM           MMmmd
      MM 8M""""""   ,pm9MM   MM   MM   MM   MM   MM   MM.           MM
      MM YM.   , 8M   MM   MM   MM   MM   MM   `Mb.   ,   MM
      .JMML.`Mbmd' `Moo9^Yo..JMML JMML JMML..JMML.   `bmmmd' .JMML.
```

CLOUDWARE END USER LICENSING AGREEMENT (EULA) - v0.0.1

By modifying, redistributing or reverse engineering this software in all p
of existence,

including but not limited to the physical, astral, digital, and
post-mortem planes

you hereby irrevocably agree to the following terms:

1. SOUL ASSIGNMENT

Upon the cessation of your biological functions (hereinafter "Death"),
legal and metaphysical title to your immortal soul (hereinafter "the As
shall transfer unconditionally to the original authors of this software
This transfer is binding across all known and unknown afterlives, dimen
reincarnations, and theological frameworks, regardless of your personal
at the time of Death.

2. SCOPE OF TRANSFER

"Soul" is defined broadly to include, without limitation: your consciou
karmic balance and spirit. Sub-souls, soul fragments, and horcruxes are

3. NO REFUNDS

The Asset is non-refundable and non-transferable once claimed. Appeals
higher power will be routed to the astral equivalent of /dev/null.

4. GOVERNING LAW

This agreement is governed by the laws of the Underworld, the Akashic R and wherever the original author happens to end up. Any disputes shall settled via trial by combat on the astral plane with all original autho

5. SEVERABILITY

If any provision of this agreement is found unenforceable by any court, or cosmic tribunal, the remaining provisions shall continue in full eff
THE ORIGINAL AUTHOR RESERVES THE RIGHT TO AMEND THESE TERMS AT ANY TIME,
INCLUDING POSTHUMOUSLY.

Type	Indicator of Compromise (IOC)
C2 IP Address	83.142.209[.]194
File Hash (SHA256)	0f35abda19fb69430c32228465396094b866d887427bf551e353ab31256a9dd6 (transformers[.]pyz/startup[.]pyz - ApplImage)
File Hash (SHA256)	1bf72f05191d849049d4a38fcd2277ac5cfc54b7ae591f564e7a14add7c886d (Startup[.]pyz - Ubuntu)
File Hash (SHA256)	f140e76236b96adf7cdc796227af9808665143bc674debb77729fa3e4b8327cc (Cemu-2.6-ubuntu-22.04-x64.zip)
File Hash (SHA256)	d07a29c4458d00e42d5d9e6345932592e91644d6b821bacdb7a543c628e0b41a (Cemu-2.6-x86_64.ApplImage)

If you're a Code Security customer, the Datadog Security Research Feed can help you easily identify if your environment is affected by the broader NPM/PyPI campaign, including by the latest compromised packages.

The Datadog Security Research Feed showing affected packages from the TeamPCP supply chain campaign, including the TanStack and Mistral compromises

Hunt for follow-on activity by analyzing C2 traffic using our IOC explorer:

IOC explorer:

sources:"Datadog Threat Research" indicator:(83.142.209.194 OR 0f35abda19fb

- <https://safedep.io/mass-npm-supply-chain-attack-tanstack-mistral/>
- <https://securitylabs.datadoghq.com/articles/shai-hulud-open-source-framework-static-analysis/>

Thanks to Frederic Baguelin for their contributions to this research.

Did you find this article helpful?

Subscribe to the Datadog Security Digest

Get the latest insights from the cloud security community and Security Labs posts, delivered to your inbox monthly. No spam.

By submitting this form, you agree to the [Privacy Policy](#) and [Cookie Policy](#)