Is LM Studio Infected? How to Check Your Install (March 2026)

March 25, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** LM Studio 0.4.7 on Windows is triggering Windows Defender detections for Trojan:JS/GlassWorm.ZZ!MTB. The flagged file is a standard webpack-bundled Electron JavaScript file. Only 1 of 62 antivirus engines on VirusTotal flags it — Microsoft alone. The LM Studio team says they believe it's a false positive and are investigating. That said, the timing is terrible: the litellm PyPI package was hit by a real supply chain attack the same week, and GlassWorm malware has compromised 400+ repos on GitHub and npm. Verify your install, check the hash, and read on for specific steps.

📚 **More on this topic:** Ollama vs LM Studio · LM Studio Tips & Tricks · Local AI Privacy Guide

If Windows Defender just quarantined your LM Studio install and you're staring at a trojan warning, you're not alone. Reports started hitting Reddit and GitHub this week. Here's what's actually going on.

What happened

On March 23, 2026, users began reporting that Windows Defender was flagging LM Studio 0.4.7 as malware. Defender identified the threat as `Trojan:JS/GlassWorm.ZZ!MTB` in the file:

```
C:\Program Files\LM Studio\resources\app\.webpack\main\index.js
```

That file is a ~14 MB webpack-bundled JavaScript file, the Electron main process that powers LM Studio's desktop app. After detection, Defender deleted or quarantined files, leaving the application broken.

The report landed on LM Studio's GitHub bug tracker within hours. Multiple users confirmed the same detection on fresh installs downloaded directly from lmstudio.ai.

What we know (and what we don't)

The evidence so far points toward a false positive. Here's why.

Only 1 of 62 antivirus engines on VirusTotal flagged the file. That one engine is Microsoft Defender. When a single engine flags something that 61 others pass, the odds favor a heuristic misfire over actual malware.

A user named GoZippy on the GitHub issue did forensic analysis of the index.js file. It contains product-specific strings like "To run LM Studio, you need to quit the daemon first" and references to GGUF quantization names and token counts. The "obfuscation" that triggered the heuristic is standard Vite/esbuild minification. Every Electron app ships code that looks like this.

Team member Yagil posted: "We are investigating this report with priority. We currently believe it is a false positive." They also confirmed LM Studio does not use litellm, which matters because of a separate real attack that week (more on that below).

This has happened before. LM Studio 0.3.5 was flagged as `Trojan:Win32/Cinjo.O!cl` by Windows Defender in late 2024. AVG flagged `IDP.HEUR.26` on LM Studio in January 2025. Both turned out to be false positives. Electron apps are heuristic magnets. Large bundled JavaScript files trigger pattern-matching rules designed to catch actually malicious obfuscated scripts.

What we don't know: whether the GlassWorm detection signature is related to the broader GlassWorm campaign (which has compromised 400+ repositories on GitHub, npm, and VSCode extensions this month), or whether Defender simply picked an unfortunate signature name for an unrelated heuristic match.

How to check your LM Studio installation

If you're worried, here's what to do. Takes about 10 minutes.

1. Check VirusTotal yourself

Upload your LM Studio installer (or the flagged index.js file) to virustotal.com. If only Microsoft flags it and 60+ engines say it's clean, you're looking at a false positive.

2. Verify you downloaded from the right place

The only legitimate source is **lmstudio.ai**. Not lm-studio.ai, not lmstudio.com, not a GitHub release someone linked on Reddit. Check your browser history. If you downloaded from anywhere else, uninstall and redownload from the official site.

3. Check for unexpected network connections

On Windows:

```
# See what LM Studio is connecting to
netstat -b | findstr -i "lm"

# Or use Resource Monitor (resmon.exe) → Network tab
# Look for LM Studio processes and check destination addresses
```

On Mac:

```
# Check active connections while LM Studio is running
lsof -i -P | grep -i "lm"
```

LM Studio should only be connecting to lmstudio.ai (for updates and model downloads), Hugging Face (for model downloads), and localhost (for its local API server). Connections to unknown domains are a red flag.

4. Check file hashes

Compare the hash of your installer against what other users report on the GitHub issue thread.
On Windows:

```
Get-FileHash "C:\path\to\LM-Studio-Setup.exe" -Algorithm SHA256
```

On Mac:

```
shasum -a 256 /path/to/LM\ Studio.dmg
```

If your hash doesn't match what others are reporting for the same version, you may have a tampered copy.

5. Check for litellm contamination

This is separate from LM Studio, but if you use Python for any AI work:

```
pip show litellm
```

If you're running litellm version 1.82.7 or 1.82.8, you have a real problem. Those versions were compromised in a supply chain attack (details below). Pin to 1.82.6 or earlier and rotate every credential on that machine.

The litellm attack is the one to worry about

While the LM Studio detection looks like a false positive, the litellm supply chain attack that dropped the same week is confirmed and serious.

On March 24, a threat group called TeamPCP published backdoored versions of the litellm Python package (v1.82.7 and v1.82.8) to PyPI. The attack exploited a compromised Trivy security scanner in litellm's CI/CD pipeline to steal the PyPI publishing token.

The payload harvested SSH keys, cloud credentials (AWS, GCP, Azure), Kubernetes secrets, cryptocurrency wallets, and .env files. It then deployed privileged pods across every node in accessible Kubernetes clusters and installed a persistent systemd backdoor (`sysmon.service`) that polled for additional payloads. Stolen data was encrypted with a 4096-bit RSA key and exfiltrated to `models.litellm.cloud`, a domain that looks official but isn't part of litellm's infrastructure. The malicious versions were live for about five and a half hours before PyPI pulled them.

Litellm gets 3.4 million downloads per day. Wiz estimates it's present in 36% of cloud environments.

If you installed or updated litellm on March 24: assume compromise, rotate everything, and check for the `litellm_init.pth` file on your system. Version 1.82.8 was especially aggressive. It used a `.pth` file that executes on every Python process startup, not just when you import litellm.

What to do if you're concerned about LM Studio

If you want to keep using LM Studio, the simplest path is to wait for Microsoft to update the Defender signature. Previous false positives were resolved within days. If you've already verified the hash and VirusTotal results and don't want to wait, add an exclusion in Windows Defender for the LM Studio directory (Settings > Virus & threat protection > Exclusions). If Defender already deleted files, redownload from lmstudio.ai and reinstall.

If you want alternatives while this gets sorted out:

Ollama runs the same models from the command line with an OpenAI-compatible API. No Electron, no GUI, nothing for Defender to flag. Or go straight to llama.cpp for direct inference without any wrapper application. Both are open source and you can read every line of code, which you can't do with LM Studio. That's worth thinking about even after this false positive gets resolved.

The pattern

This is the second time in three months that local AI tools have been caught up in security incidents. OpenClaw's skill marketplace hit 1,184 malicious skills on ClawHub. The litellm attack compromised one of the most popular AI Python libraries. GlassWorm is hiding malware in invisible Unicode across 400+ GitHub repos and npm packages. And now Windows Defender is flagging one of the most popular local AI desktop apps.

Running models locally is more private than sending your data to OpenAI. But "local" doesn't mean "safe." The software you use to run those models still comes from the internet. It still has dependencies. It still auto-updates.

Verify your sources. Check your hashes. Watch your network connections. The LM Studio detection is almost certainly a false alarm. The next one might not be.

Related guides

- Ollama vs LM Studio: Speed, Setup, and Verdict
- LM Studio Tips & Tricks
- Local AI for Privacy: What's Actually Private
- Best OpenClaw Alternatives in 2026

Get notified when we publish new guides.

Subscribe — free, no spam
