SUPPLY-CHAIN BRIEFING

LiteLLM Supply-Chain-Angriff

PyPI-Pakete v1.82.7 & v1.82.8 kompromittiert durch TeamPCP

24. März 2026 · Lageeinordnung

~3–5 Std.
Exposure-Fenster
Quarantäne: ~13:38 UTC
Löschung: ~15 Uhr UTC
Konservativ: 10:39–~15 Uhr

3,4 Mio.
Downloads / Tag
überwiegend automatisierte CI/CD-Läufe, nicht Endgeräte

Phase 09
TeamPCP Kampagne
Laufende Angriffsserie über 5 Ökosysteme

PYSEC-2026-2 · SNYK-PYTHON-LITELLM-15762713 · sonatype-2026-001357

Hinweis: Kursierende Opferzahlen (500.000 Laptops) sind nicht belegt. Die 95 Mio. monatlichen Downloads sind überwiegend automatisierte CI/CD-Pipelines, nicht Endnutzer-Installationen.

02

Was passiert ist

Am 24. März 2026 wurden zwei kompromittierte Versionen der Python-Bibliothek LiteLLM auf PyPI veröffentlicht.

LiteLLM ist eine zentrale Schnittstelle zu über 100 LLM-Anbietern und wird in zahlreichen AI-Agent-Frameworks, MCP-Servern und LLM-Orchestrierungstools als Abhängigkeit verwendet.

⚠ Ausführungstrigger

v1.82.7: Schadcode wird bei Import von litellm.proxy.proxy_server ausgeführt (z. B. beim Start des LiteLLM-Proxy).

v1.82.8: Zusätzlich eine .pth-Datei, die bei jedem Python-Interpreter-Start ausgeführt wird – auch durch pip, pytest oder IDE-Language-Server. Der Trigger ist nicht pip install selbst, sondern der nächste Python-Prozessstart danach.

Der Erstentdecker (FutureSearch) hatte nie pip install litellm ausgeführt – es wurde durch ein **Cursor-MCP-Plugin als transitive Abhängigkeit installiert**.

Angriffsvektor

1 **Trivy CI/CD**
kompromittiert (19.03.)

2 **PYPI_PUBLISH Token**
aus GitHub Actions Runner gestohlen

3 **litellm 1.82.7 + 1.82.8**
direkt auf PyPI hochgeladen (kein GitHub-Release)

4 **Credential Harvesting**
SSH, Cloud, K8s, Wallets, LLM-Keys

03

Timeline

Ende Feb
Pwn Request gegen Trivy CI (GHSA-9p44-j4g5-cfx5)

19.03. 17:43
Trivy v0.69.4 Tags umgeschrieben (Snyk)

23.03. 12:58
KICS kompromittiert, C2-Domains registriert (Endor Labs)

24.03. 10:39
litellm 1.82.7 auf PyPI (Endor Labs, LiteLLM offiziell)

24.03. 10:52
litellm 1.82.8 auf PyPI mit .pth (FutureSearch)

24.03. 11:48
Disclosure-Issue #24512 eröffnet (GitHub)

24.03. ~12:44
Bot-Spam auf Issue; Issue geschlossen (Snyk, The Register)

24.03. ~13:38
PyPI-Quarantäne – keine Neuinstallation mehr möglich (Snyk)

24.03. ~15 Uhr
Maintainer bestätigt: alle Keys rotiert (GitHub #24518, laut Snyk)

24.03. kurz danach
Kompromittierte Versionen endgültig gelöscht (GitHub #24518, laut Snyk)

Quellen jeweils in Klammern angegeben

Dreistufiger Payload

Stufe 1 – Credential Harvesting (FutureSearch, Endor Labs, SafeDep)

Sammelt systematisch:

→ SSH-Keys (~/.ssh/) · AWS/GCP/Azure-Credentials

→ Kubernetes-Configs · Docker-Configs

→ Sämtliche Umgebungsvariablen + .env-Dateien

→ Git-Credentials · Shell-History · DB-Configs

→ npm/Vault/Netrc-Tokens

→ Krypto-Wallets (BTC, ETH, SOL, LTC, XMR, DOGE, ZEC, DASH, XRP, ADA)

Stufe 2 – Exfiltration (FutureSearch, Endor Labs)

AES-256-CBC + RSA-4096 Verschlüsselung

Versand als tpcp.tar.gz via HTTPS POST an

models.litellm[.]cloud

Stufe 3 – Persistenz & Lateral Movement (Endor Labs, ARMO)

→ Backdoor ~/.config/sysmon/sysmon.py + systemd-Service (Linux)

→ Pollt checkmarx[.]zone/raw alle 50 Min.

→ Kubernetes: Cluster-Secrets auslesen, privilegierte Pods deployen

→ Kill-Switch: youtube.com-URL in C2 → Abort

→ Windows/macOS-Artefakte: bisher nicht dokumentiert (Linux-fokussiert)

Zur Erkennung und Prävention:

→ Hash-Pinning (--require-hashes): Hätte hier nicht gegriffen, da die Wheel-Hashes korrekt waren – das Paket wurde mit legitimen Credentials veröffentlicht (Snyk).

→ Source-to-Artifact-Vergleich ist die stärkste Einzelkontrolle, aber keine alleinige Lösung. Trusted Publishing, Egress-Monitoring, SCA-Tools und reproduzierbare Builds adressieren unterschiedliche Kontrollziele.

05

Indicators of Compromise

Netzwerk-IOCs (alle techn. Quellen)

C2-Domain	**models.litellm[.]cloud**	Exfiltration (HTTPS POST)
C2-Domain	**checkmarx[.]zone**	Payload + Polling (/raw)
Registrar	**Spaceship, Inc.**	reg. 23.03.2026 (SafeDep)
Hosting	**DEMENIN B.V.**	Shared Provider (Snyk)

Dateisystem-IOCs (Linux – Windows/macOS bisher nicht dokumentiert)

- litellm_init.pth (site-packages/) – v1.82.8
- litellm/proxy/proxy_server.py – v1.82.7 + v1.82.8
- ~/.config/sysmon/sysmon.py – Persistente Backdoor
- ~/.config/systemd/user/sysmon.service – systemd Auto-Start
- /tmp/tpcp.tar.gz – Verschl. Exfiltrations-Archiv
- /tmp/session.key · /tmp/payload.enc – Temp. Artefakte
- /tmp/session.key.enc · /tmp/.pg_state · /tmp/pglog

Kubernetes-IOCs (Endor Labs, ARMO, FutureSearch)

- node-setup-* Pods in kube-system (privilegierte Alpine)
- Unautorisierter Secrets-Zugriff über alle Namespaces
- /root/.config/sysmon/sysmon.py auf Nodes via Pod-Mount

Hashes (Formate beachten)

Wheel SHA256 (1.82.8) – hex-kodiert (SafeDep)
d2a0d5f564628773b6af7b9c11f6b86531a875bd2d186d7081ab62748a800ebb

litellm_init.pth – Base64-kodierter SHA256 aus RECORD (GitHub #24512)
ceNa7wMJnNHy1kRnNCcwJaFjWX3pORLfMh7xGL8TUjg

RSA Public Key (Anfang) – eingebettet im Payload (GitHub #24512)
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvahaZDo8...

> Plattformhinweis: Alle IOCs und Persistenz-Mechanismen sind Linux-spezifisch. Für Windows/macOS liegen bisher keine dokumentierten Artefakte vor. Der .pth-Mechanismus selbst ist plattformübergreifend.

Bin ich betroffen?

✓ Nicht betroffen (LiteLLM offiziell bestätigt)

→ Offizielles Docker-Image (ghcr.io/berriai/litellm)

→ LiteLLM Cloud

→ Version ≤ 1.82.6 ohne Upgrade im Zeitfenster

→ Installation direkt aus GitHub-Repository

⚠ Potenziell betroffen

Konservatives Betroffenheitsfenster: 24.03., 10:39–~15 Uhr UTC

(10:39 = Upload v1.82.7 → ~13:38 = Quarantäne → ~15 Uhr = endgültige Löschung (laut Snyk))

1. pip install/upgrade von litellm in diesem Zeitfenster
2. Ungepinnte pip install litellm (v1.82.7 oder .8 erhalten)
3. Docker-Builds mit ungepinntem litellm in diesem Zeitfenster
4. Transitive Installation durch AI-Frameworks, MCP-Server, LLM-Tools
5. CI/CD-Pipelines, die in diesem Zeitfenster pip install ausgeführt haben

Prüfbefehle (Linux/macOS – Windows: Pfade anpassen)

```
$ pip show litellm

$ find / -name "litellm_init.pth" 2>/dev/null

$ ls -la ~/.config/sysmon/sysmon.py

$ systemctl --user status sysmon.service

$ kubectl get pods -n kube-system | grep node-setup

$ grep -r "models.litellm.cloud" /var/log/
```

Incident Response

Phase 1 – Eindämmung (sofort)

→ Netzwerkisolation (nicht herunterfahren – forensische Daten sichern)

→ C2-Domains blockieren: models.litellm[.]cloud + checkmarx[.]zone

→ Paket entfernen + Caches löschen (pip, uv)

→ Persistenz entfernen: sysmon.py, sysmon.service, /tmp-Artefakte

Phase 2 – Credential-Rotation (24h)

→ SSH-Keys, AWS/GCP/Azure-Credentials rotieren

→ Kubernetes: kubeconfig + Cluster Secrets + RBAC-Audit

→ Alle LLM API-Keys, DB-Passwörter, CI/CD-Tokens

→ Krypto-Wallets: Guthaben sofort auf neue Wallets transferieren

→ .env / Umgebungsvariablen: alle Secrets rotieren

Phase 3 – Forensik (24–72h)

→ Netzwerk-Logs auf C2-Domains prüfen (Proxy, DNS, Netflow)

→ Memory-Dumps + Disk-Images VOR Bereinigung erstellen

→ Dependency-Audit: requirements, Lock-Files, Docker, CI/CD (auch transitiv!)

→ Cloud-Audit-Logs: CloudTrail, Azure Activity Log, GCP Audit Log

Phase 4 – Rebuild & Härtung (72h+)

→ Betroffene Systeme aus sicherem Zustand neu aufbauen

→ Dependencies pinnen + Lock-Files verpflichtend machen

→ SCA-Tools (Snyk, Sonatype, Endor Labs) für Echtzeit-Überwachung

→ Trusted Publishing via JWT für eigene PyPI-Pakete

→ Egress-Monitoring für Dev-Workstations + CI/CD

Kontrollziele differenziert betrachten:

Hash-Pinning, Trusted Publishing, Egress-Kontrollen, reproduzierbare Builds und Runtime Detection adressieren unterschiedliche Angriffsvektoren. Keine Einzelmaßnahme hätte diesen Angriff allein vollständig verhindert.

Die Kampagne ist nicht vorbei

Phase 09 einer laufenden Supply-Chain-Kampagne. Jede Kompromittierung liefert Credentials für die nächste:

27.02.	**Trivy**	PR-Target Exploit	GitHub Actions
19.03.	**Trivy (erneut)**	Tag Force-Push	GH Actions, Docker Hub
21.03.	**Checkmarx/KICS**	GH Action Hijack	GitHub Actions
22.03.	**Aqua Security**	44 Repos defaced	GitHub
24.03.	**LiteLLM**	PyPI Account Takeover	PyPI

TeamPCP-Profil (Wiz Threat Center, Snyk)

Aliasse	PCPcat, Persy_PCP, ShellForce, DeadCatx3
Aktiv seit	mindestens Dezember 2025
5 Ökosysteme	GitHub Actions, Docker Hub, npm, Open VSX, PyPI
Signatur	"TeamPCP Cloud stealer" in Payloads
Zus. Tooling	CanisterWorm (Internet Computer Protocol als C2)

> **Endor Labs:**
> "Jede kompromittierte Umgebung liefert Credentials, die das nächste Ziel ermöglichen."

Was du jetzt tun solltest:

1. pip show litellm auf ALLEN Systemen (inkl. CI/CD)
2. Bei v1.82.7 / .8 → Credentials rotieren
3. C2-Domains in Firewall/Proxy blocken
4. Dependency Pinning + Lock-Files einführen
5. Diesen Post teilen – andere könnten betroffen sein

Teilen, wenn es jemandem helfen kann