TeamPCP avvelena LiteLLM: la supply chain Python per sviluppatori AI sotto attacco



Il 24 marzo 2026 due versioni compromesse di LiteLLM, la libreria Python più utilizzata per integrare modelli di linguaggio AI, sono state pubblicate su PyPI contenendo un malware sofisticato capace di rubare chiavi SSH, credenziali cloud, token Kubernetes e wallet crypto. Con 95 milioni di download mensili e una presenza nel 36% degli ambienti cloud globali, la portata potenziale dell'attacco è devastante. Dietro l'operazione il gruppo TeamPCP, al termine di una campagna coordinata durata quasi un mese.

Cosa è LiteLLM e perché conta

LiteLLM è la libreria Python che alimenta quasi ogni framework AI agent esistente: un'interfaccia unificata che consente alle applicazioni di comunicare con OpenAI, Anthropic, Google e decine di altri provider LLM attraverso un unico wrapper. Con oltre 40.000 stelle su GitHub e dipendenze dirette da progetti come CrewAI, DSPy, Mem0, Guardrails e Camel-AI, LiteLLM non è un semplice strumento: è l'infrastruttura portante dello sviluppo AI moderno. Come abbiamo approfondito analizzando i rischi delle dipendenze open source, colpire una singola libreria critica significa colpire tutto ciò che ci sta sopra.

L'attacco: quasi un mese di campagna sistematica

L'operazione non è nata il 24 marzo: è l'atto finale di una campagna coordinata iniziata il 28 febbraio 2026, quando un bot autonomo ha sfruttato una vulnerabilità di workflow in Trivy rubando un Personal Access Token. Come documentato su ICT Security Magazine, i supply chain attack sfruttano la

fiducia tra un’azienda e i suoi fornitori per colpire molteplici obiettivi attraverso un singolo punto di ingresso. La progressione in questo caso è stata deliberata e sistematica:

28 febbraio: bot automatico compromette Trivy per la prima volta, rubando un PAT. Aqua Security rimedia i danni superficiali, ma lascia un accesso residuo che TeamPCP sfrutterà settimane dopo.

19 marzo: TeamPCP torna su Trivy con un nuovo attacco, iniettando malware in 76 dei 77 tag di release di aquasecurity/trivy-action e tutti e 7 i tag di aquasecurity/setup-trivy. Viene pubblicato anche un binario Trivy compromesso (v0.69.4). All’incidente viene assegnato il CVE-2026-33634 con CVSS score 9.4.

21-23 marzo: TeamPCP colpisce Checkmarx, compromettendo le GitHub Actions KICS e ast-github-action, oltre alle estensioni OpenVSX cx-dev-assist 1.7.0 e ast-results, usando credenziali rubate nella compromissione di Trivy.

24 marzo, ore 10:39 UTC: LiteLLM versione 1.82.7 viene pubblicata su PyPI, seguita dalla versione 1.82.8 alle 10:52 UTC. I pacchetti rimangono disponibili per alcune ore prima che PyPI li metta in quarantena. Con 3,4 milioni di download giornalieri il danno potenziale è enorme.

Il meccanismo di accesso è stato preciso: la pipeline CI/CD di LiteLLM usava Trivy per le scansioni di sicurezza senza pinnare la versione. Il Trivy compromesso ha esfiltrato il token PYPI_PUBLISH del progetto dall’ambiente GitHub Actions runner. Con quella credenziale, TeamPCP ha pubblicato direttamente i pacchetti malevoli su PyPI, bypassando completamente il repository GitHub: nessun tag, nessuna pull request, nessuna review.

Il payload: tre stadi, cifratura RSA-4096

Il malware funziona in tre fasi distinte, con un livello di sofisticazione che va oltre la media del cybercrime ordinario:

Fase 1: Furto massivo di credenziali. Il codice raccoglie tutto ciò che trova sul sistema: chiavi SSH, variabili d'ambiente, credenziali AWS, GCP e Azure, configurazioni Kubernetes, password di database, file .gitconfig, cronologia della shell e wallet di criptovaluta. I dati vengono cifrati con AES-256-CBC, con la chiave a sua volta cifrata con una chiave RSA-4096 pubblica hardcoded, e inviati all'attaccante in un archivio denominato tpcp.tar.gz.

Fase 2: Lateral movement Kubernetes. Se vengono rilevati token di service account Kubernetes, il malware tenta di creare pod privilegiati nel namespace kube-system, leggendo tutti i segreti del cluster e distribuendo backdoor su ogni nodo.

Fase 3: Backdoor persistente. Viene installato un servizio systemd (sysmon.service / sysmon.py) che interroga il server C2 per ricevere payload aggiuntivi, garantendo accesso permanente anche dopo la rimozione del pacchetto compromesso.

Le due versioni compromesse usano tecniche di iniezione diverse, con una distinzione tecnica importante. La versione 1.82.7 inietta un payload base64 nel file proxy_server.py e richiede un import per attivarsi, mentre la versione 1.82.8 installa un file litellm_init.pth che si esegue ad ogni avvio del processo Python, anche senza importare la libreria. Come sottolineato nella nostra analisi su DevSecOps e sicurezza della pipeline, questo rende la versione 1.82.8 significativamente più pericolosa: qualsiasi script Python, test runner o tool invocato in un ambiente con LiteLLM installato attiva silenziosamente il credential harvester in background.

Come è stato scoperto: pura fortuna

La scoperta è avvenuta per un errore degli stessi attaccanti: il file .pth generava processi figli che a loro volta attivavano lo stesso file, creando un fork bomb esponenziale che mandava in crash le macchine colpite. Callum McMahon, ricercatore di FutureSearch, ha notato un processo che consumava tutta la RAM del sistema durante il test di un plugin MCP su Cursor, che aveva scaricato LiteLLM come dipendenza transitiva, senza che l'utente avesse mai eseguito pip install litellm
. L'indagine ha portato alla scoperta del payload completo.

Come ha commentato Andrej Karpathy su X: "Se l'attaccante non avesse scritto codice sloppily, avrebbe potuto restare non rilevato per giorni o settimane. Una semplice pip install litellm
era sufficiente per esfiltrare chiavi SSH, credenziali AWS, GCP, Azure, configurazioni Kubernetes, variabili d'ambiente, wallet crypto e segreti CI/CD."

Sul fronte della risposta, TeamPCP ha lasciato un calling card esplicito: un commit su un repository forkato del maintainer con il messaggio "teampcp owns BerriAI", e ha chiuso l'issue GitHub #24512 come "not planned", suggerendo che l'account fosse ancora compromesso. Il team LiteLLM ha immediatamente sospeso tutti i nuovi rilasci, ruotato le credenziali e ingaggiato Mandiant di Google per l'analisi forense della pipeline.

La minaccia non è finita

Dopo l'attacco, TeamPCP ha rilasciato un messaggio esplicito: "Many of your favourite security tools and open-source projects will be targeted in the months to come." Non si tratta di una minaccia vaga: il gruppo ha attraversato cinque ecosistemi in meno di un mese, GitHub Actions, Docker Hub, npm, OpenVSX e PyPI, con una tecnica che trasforma ogni compromissione in un vettore per la successiva. Come

evidenziato nella nostra analisi sull’uso di npm e PyPI come vettori di attacco, nel 2025 sono stati distribuiti oltre 454.000 nuovi pacchetti malevoli nei registri open source globali: LiteLLM è solo l’ultimo capitolo.

Cosa fare immediatamente

Il comunicato ufficiale del team LiteLLM e le raccomandazioni di BleepingComputer indicano le seguenti azioni prioritarie:

Verificare la versione installata con pip show litellm
in tutti gli ambienti Python, inclusi ambienti virtuali, runner CI/CD, immagini Docker e ambienti di staging. Considerare compromessa qualsiasi macchina con versione 1.82.7 o 1.82.8 installata durante la finestra di esposizione e ruotare immediatamente tutte le credenziali accessibili: chiavi SSH, token cloud AWS/GCP/Azure, password database, API key e segreti CI/CD. Verificare la presenza del file ~/.config/sysmon/sysmon.py
e del servizio systemd associato. Controllare i cluster Kubernetes per pod non autorizzati nel namespace kube-system. La versione sicura certificata è la 1.82.6. Un approccio strutturato tramite Software Bill of Materials (SBOM) e dependency pinning con hash verification rappresenta oggi la principale misura preventiva per evitare che scenari analoghi si ripetano.

Condividi sui Social Network:
https://www.ictsecuritymagazine.com/notizie/litellm-supply-chain/