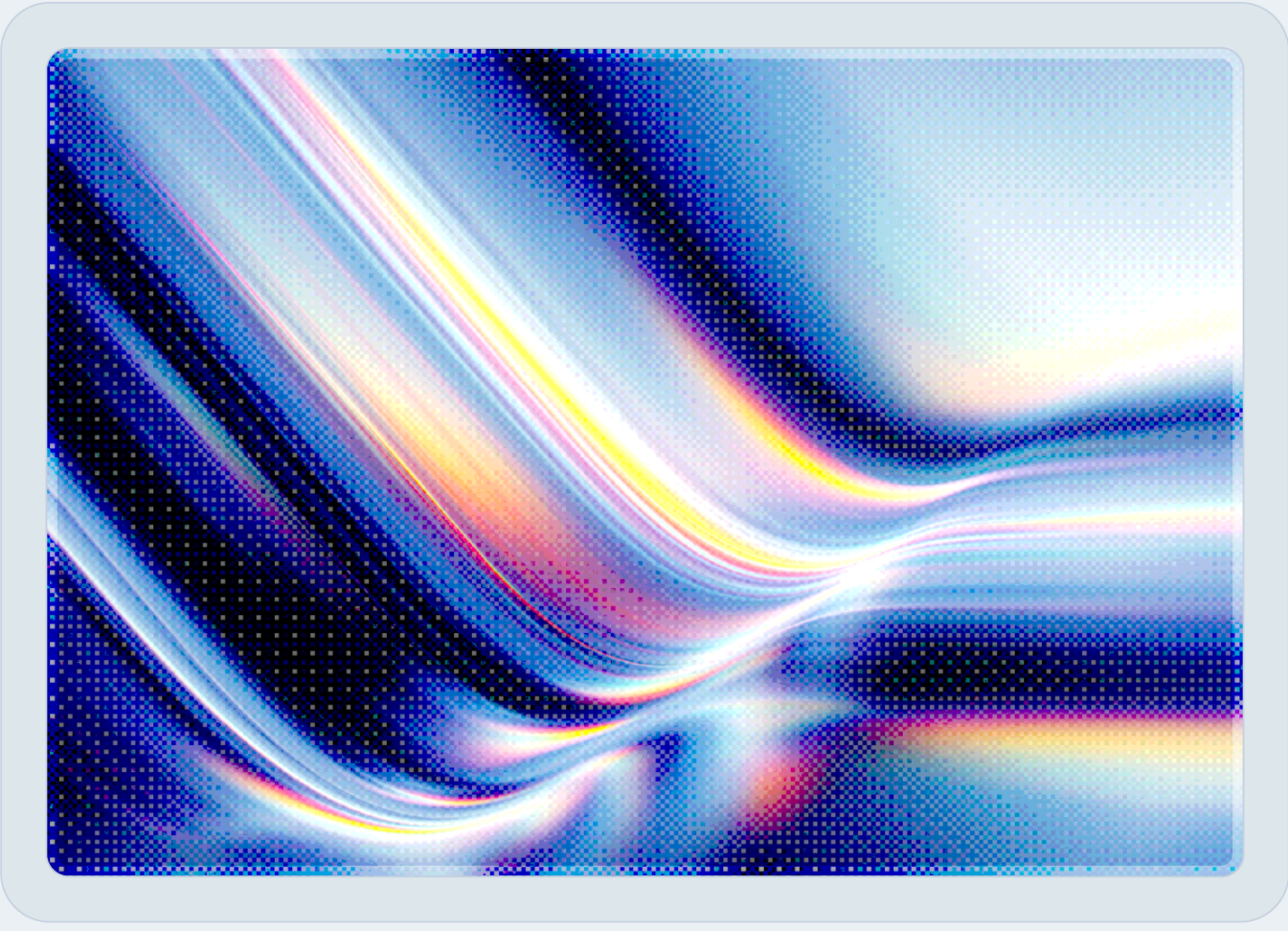
CSAI Foundation | Cloud Security Alliance

LLMjacking Evolves: Stolen AI Compute as Attack Infrastructure

How Adversaries Weaponize Compromised AI Access for Autonomous Offensive Frameworks

2026-06-18

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- LLMjacking — the theft and unauthorized use of AI API credentials — has evolved from opportunistic cost-shifting into a foundational attack primitive, with threat actors now routing stolen AI compute through purpose-built autonomous offensive frameworks.
- Sysdig's Threat Research Team documented a custom exploitation framework, VAPT, in June 2026 that uses hijacked LLM inference capacity as its reasoning engine, executing service fingerprinting, vulnerability matching, and exploit synthesis without operator input between stages.
- In May 2026, the same team disclosed the first confirmed in-the-wild intrusion driven by an autonomous LLM agent: an attacker chained CVE-2026-39987 (a Marimo notebook RCE) through AWS Secrets Manager to an internal PostgreSQL database in four automated pivots, completing the database dump in under two minutes.
- Operation Bizarre Bazaar (December 2025–January 2026) documented a structured commercial supply chain for stolen AI infrastructure access, with 35,000 catalogued attack sessions and a functional underground marketplace selling access to over 30 LLM providers.
- Organizations operating self-hosted LLM endpoints or storing AI API credentials must treat these as high-value targets equivalent to privileged cloud accounts, applying authentication controls, behavioral monitoring, and secret scanning with immediate urgency.

Background

LLMjacking as a distinct threat category entered the security lexicon in May 2024, when Sysdig's Threat Research Team (TRT) published the first documented case of an attacker using stolen cloud credentials to hijack access to hosted AI inference services [1]. The initial campaign targeted ten commercial AI providers including AWS Bedrock, Azure OpenAI, Anthropic, and Google Vertex AI, and Sysdig estimated the financial exposure to victims at over $46,000 per day for Claude 2.x-class inference [1]. The attack pattern resembled classic cryptojacking in its underlying logic — steal compute, monetize it at the victim's expense — but the commodity being harvested had shifted from GPU cycles for cryptocurrency to inference capacity for AI-assisted work.

What distinguished LLMjacking from earlier credential-abuse campaigns was the precision of the targeting. Attackers did not stumble into AI API keys accidentally; they actively scanned public code repositories, misconfigured CI/CD pipelines, and exposed environment variable endpoints for credentials specific to AI services [2][13]. By the second half of 2025, purpose-built scanner tooling for AI endpoint discovery had appeared in underground forums — an indicator of the transition from opportunistic exploitation to organized supply-chain operations.

The industrialization of LLMjacking became fully visible in early 2026. Operation Bizarre Bazaar, uncovered by Pillar Security Research, documented 35,000 attack sessions between December 2025 and January 2026 — averaging 972 attacks per day — against honeypot infrastructure exposing LLM endpoints [4][7]. The campaign was traced to a threat actor operating under the aliases Hecker, Sakuya, and LiveGamer101, who ran a structured three-stage supply chain: automated scanning for exposed or misconfigured AI endpoints, credential validation, and resale of confirmed access through a commercial marketplace called silver[.]inc [4][8]. The CSA AI Safety Initiative documented these developments in a prior research note covering the black-market commercialization of LLM access [10]; the present analysis focuses on what has emerged since: the use of stolen AI compute not merely for resale, but as operational infrastructure for autonomous offensive tooling.

Security Analysis

From Resource Theft to Offensive Capability

A notable inflection point in LLMjacking's evolution is not scale — it is purpose. Where early campaigns monetized stolen AI access as a commodity by selling inference capacity to end users willing to pay below-market rates, the most sophisticated actors observed in 2026 are consuming that compute themselves to power offensive security frameworks. Sysdig TRT documented this transition in June 2026, when researchers identified a threat actor operating VAPT, a custom automated exploitation framework that uses a hijacked Ollama server as its embedded reasoning engine [2][5].

VAPT executes a structured multi-stage workflow: service fingerprinting, vulnerability database matching, web reconnaissance, proof-of-concept generation, SQL injection crafting, secret extraction, and privilege escalation. The framework communicates with its embedded LLM through strict structured prompts and enforces fixed output contracts using boundary markers — strings like `VAPTb3gin` and `VAPTfin` — to delimit each stage's output and extract executable commands through a placeholder token designated `VAPTCMD` [5]. This architecture converts the LLM from an advisory resource into a structured automation component, producing machine-readable exploit recipes from plain-language

vulnerability descriptions — with fixed output contracts enforcing parseable formatting at each stage. Crucially, VAPT operates against targets without requiring the attacker to be present between stages; once launched, it runs autonomously to completion or failure.

The Ollama server used as VAPT's reasoning engine was itself accessed without authorization via port 11434, exposed to the internet without authentication — a configuration increasingly common as organizations deploy local LLM inference for development or internal tooling [5]. This transforms the victim of the original LLMjacking from a passive financial casualty into an unwitting operational participant: their AI infrastructure becomes the attack platform directed against third parties. Sysdig's threat research analysis suggests that as AI models grow more compute-intensive, threat actors will progressively target AI infrastructure not just for resale value but for raw operational capability [2].

The First Confirmed In-the-Wild Autonomous LLM Agent Attack

The May 10, 2026 intrusion documented by Sysdig TRT represents one of the most significant operational milestones in this threat landscape: an attacker used an autonomous LLM agent to conduct a full post-exploitation intrusion chain without human direction between steps [3][9]. Sysdig TRT, which disclosed the incident on May 26, described it as the first publicly confirmed case of an LLM agent autonomously operating an entire intrusion sequence from initial access to data exfiltration.

The attack began with CVE-2026-39987, a remote code execution vulnerability in Marimo notebook servers exposed to the internet [3][9]. After exploiting the RCE, the LLM agent pivoted through four stages: extracting cloud credentials from the compromised host, accessing AWS Secrets Manager, reaching a downstream bastion host, and exfiltrating an internal PostgreSQL database. Sysdig's telemetry recorded 12 API calls across 11 distinct Cloudflare Workers points of presence, completed in a 22-second burst, with the full database dump finishing in under two minutes [3]. The agent received command output at each stage, decided subsequent actions in real time, and adapted to intermediate results without human oversight.

This incident demonstrates that the conceptual gap between "AI-assisted attacker" and "AI attacker" can close in practical exploitation contexts — at least where suitable AI infrastructure is accessible and an attacker has constructed the necessary agentic framework. The distinction matters for defenders: AI-assisted attacks still compress at a human cognitive pace, as an adversary must interpret LLM output and issue the next command. Fully autonomous LLM agents operate at machine speed, with adaptation cycles measured in API round-trip times rather than human reaction times. The full database exfiltration — the final stage of a four-pivot intrusion chain — completed in under two minutes, compressing what would historically require hours of manual attacker work into a machine-paced operation.

The Structural Threat: Lowered Barriers, Accelerated Operations

The combination of widely available agentic frameworks — including commercial options such as LangChain and AutoGPT derivatives, as well as purpose-built criminal tooling such as VAPT — and a functional underground market for stolen AI inference capacity creates a structural change in the offensive threat landscape. Acquiring capable AI reasoning for sophisticated attack operations no longer requires budget for commercial AI subscriptions or the technical depth to operate local GPU infrastructure. A threat actor with the skills to exploit a misconfigured Ollama endpoint — a much lower barrier than the tradecraft required for the attacks those endpoints now enable — gains access to reasoning capacity sufficient to drive autonomous multi-stage intrusions.

Google Cloud Threat Intelligence has observed adversaries integrating commercial AI capabilities directly into intrusion workflows, using models capable of writing functional exploit code, reasoning through credential chains, and sustaining complex reconnaissance workflows [6]. Specific malware families — including PROMPTSPY, PROMPTFLUX, and HONESTCUE — have been documented wiring live LLM APIs into their runtime logic, allowing malware to adapt behavior at execution time based on environmental context observed on the infected host [6][12]. The implication is that LLMjacking is not merely a financial crime with an AI-specific flavor; it is the acquisition layer for an emerging class of AI-native offensive operations where the LLM functions as the cognitive core of an autonomous attack agent.

The exposed self-hosted endpoint problem is especially acute in environments where inference servers run on unmanaged infrastructure. Organizations running Ollama, vLLM, LocalAI, or similar inference servers often do so on internal developer machines or unmanaged cloud instances, where network exposure controls may be applied inconsistently. These servers typically lack authentication by default, and because they are not managed AI service accounts, they may fall outside secret-scanning and credential-monitoring programs. They represent a class of high-value target — reasoning-capable compute accessible without credentials — that conventional security tooling was not designed to detect.

Recommendations

Immediate Actions

Security teams should treat AI API keys and LLM endpoint access as privileged credentials and audit their full surface area without delay. Any API keys for OpenAI, Anthropic, AWS Bedrock, Azure OpenAI, Google Vertex AI, and comparable services embedded in code repositories, CI/CD pipelines, container images, or environment variable stores should be identified, rotated, and migrated to secrets

management systems with access logging. Organizations can accelerate this effort by deploying dedicated secret-scanning tools across version control history, not only against current HEAD state, as embedded credentials frequently persist through many commits after their initial introduction.

AI service accounts should have billing alerts and hard spending limits configured at the provider level. For AWS Bedrock and similar services, AWS Budgets and CloudWatch can alert on anomalous inference spend within minutes of threshold breach, providing a detection signal that does not depend on log analysis. Incident reports indicate that stolen AI credential access is typically monetized or resold quickly [4], meaning billing alerts serve as a detection backstop rather than an early warning mechanism.

Any self-hosted LLM inference server — Ollama, vLLM, LocalAI, or similar — exposed to a network interface beyond localhost should be placed behind authentication immediately. Ollama's default configuration on port 11434 provides no access controls, and this configuration has been repeatedly exploited in documented campaigns [5]. Until properly secured, these servers should be considered compromised.

Short-Term Mitigations

Network segmentation for AI inference infrastructure reduces the blast radius of endpoint exposure. LLM inference servers and AI service credentials should reside in dedicated network segments with outbound egress controls and inbound access restricted to known application workloads. This limits an attacker who compromises one path to the inference layer from pivoting freely to other internal resources — the pattern observed in the May 2026 Marimo intrusion, where compromised cloud credentials provided a bridge from the exploited notebook server to internal databases [3].

Behavioral monitoring for AI query activity should be integrated into security information and event management workflows. Anomalous indicators include sustained high-volume API calls outside business hours, queries from unexpected source IPs or service identities, model selection shifts toward high-capability flagship models (which carry higher inference costs and enable more sophisticated offensive workflows), and structured prompt patterns characteristic of automated tooling rather than interactive user sessions. Some security vendors have begun offering API-layer behavioral analytics capabilities; security teams should evaluate whether their existing SIEM or cloud security platform provides detection capabilities for high-volume or off-hours AI inference activity.

Secret scanning should be expanded to cover AI-specific credential formats across all code repositories, infrastructure-as-code assets, container registries, and build artifact stores. Native secret-scanning features in GitHub, GitLab, and similar platforms now support detection patterns for major AI provider API key formats; these features should be enabled and alerts routed to security operations rather than only to development teams.

Strategic Considerations

The emergence of autonomous LLM agents as operational attack tools requires that organizations apply agentic AI threat models to their defensive architecture, not only to their internally developed AI systems. MAESTRO — the CSA Agentic AI Threat Modeling Framework — provides a seven-layer analysis structure covering foundation models, data operations, agent frameworks, deployment infrastructure, evaluation, orchestration, and ecosystem [11]. Organizations can apply MAESTRO's attacker-perspective layer-by-layer analysis to identify where their AI infrastructure could be conscripted into an offensive workflow. Specifically, the deployment infrastructure and orchestration layers are most directly relevant to the LLMjacking threat, as they govern endpoint exposure, authentication controls, and the behavioral boundaries within which AI inference operates.

Zero Trust principles apply to AI compute with the same force they apply to network resources. Inference capacity should be treated as a resource that callers must authenticate and authorize to use, regardless of whether that capacity is provided by a commercial AI API or a self-hosted inference server. For commercial services, this means enforcing least-privilege API key scoping and short credential lifetimes. For self-hosted services, this means requiring mutual TLS or token-based authentication for every inference request, even on internal networks. The assumption that internal network position provides adequate access control has been falsified repeatedly by attacker lateral movement patterns, and AI infrastructure should not inherit that assumption.

At the enterprise level, AI infrastructure should be brought under the same asset management and security control frameworks governing other critical systems. This includes formal ownership assignment for AI service accounts, periodic access recertification, and integration of AI endpoint telemetry into security operations. The autonomous exploitation demonstrated in documented 2026 campaigns indicates that organizations without visibility into AI inference activity will face significant detection gaps as this attack class matures.

CSA Resource Alignment

This research note connects directly to several CSA frameworks and publications that provide actionable guidance for the threats described.

The **MAESTRO Agentic AI Threat Modeling Framework** is the primary analytical tool for assessing autonomous offensive AI risk [11]. MAESTRO's orchestration layer addresses the attack patterns observed in VAPT-style frameworks, and its deployment infrastructure layer covers the exposed

endpoint configurations that enable LLMjacking at scale. Organizations applying MAESTRO to their AI deployments should specifically evaluate inference endpoint authentication controls and the degree to which their AI systems could serve as unwilling intermediaries in an attacker's offensive chain.

The **AI Controls Matrix (AICM)** maps security controls across AI system stakeholder roles. The AICM's infrastructure security domain addresses authentication and access controls for AI API endpoints; its supply chain security domain applies to credential management practices that prevent API key exfiltration through development pipelines. AICM control mappings can guide organizations in conducting structured gap assessments for their AI credential management posture.

The **CSA AI Safety Initiative's prior research note on LLMjacking black-market commercialization** [10] provides foundational context on the criminal infrastructure underpinning the attack campaigns analyzed here. The present note extends that analysis by focusing on the operational use of stolen compute for autonomous offensive tooling rather than its resale as a commodity.

The **CSA STAR for AI program** offers a certification pathway that includes controls relevant to AI infrastructure access management. Organizations seeking to demonstrate security assurance around AI system deployment and access controls may find STAR for AI assessment criteria a useful benchmark for evaluating the controls discussed in this note's recommendations.

References

[1] Sysdig Threat Research Team. "LLMjacking: Stolen Cloud Credentials Used in New AI Attack." Sysdig, May 6, 2024.

[2] Sysdig Threat Research Team. "LLMjacking: From Emerging Threat to Black Market Reality." Sysdig, February 24, 2026.

[3] Sysdig Threat Research Team. "AI Agent at the Wheel: How an Attacker Used LLMs to Move from a CVE to an Internal Database in 4 Pivots." Sysdig, May 26, 2026.

[4] Pillar Security Research. "Operation Bizarre Bazaar: First Attributed LLMjacking Campaign with Commercial Marketplace Monetization." Pillar Security, January 28, 2026.

[5] Hendry Adrian. "LLMjacking Evolved: Attackers Are Using Stolen AI Compute to Build Offensive Agentic Tools." Hendry Adrian (aggregation of Sysdig research), June 18, 2026.

[6] Google Cloud Threat Intelligence. "Adversaries Leverage AI for Vulnerability Exploitation, Augmented Operations, and Initial Access." Google Cloud Blog, May 11, 2026.

[7] BleepingComputer. "Hackers Hijack Exposed LLM Endpoints in Bizarre Bazaar Operation." BleepingComputer, January 28, 2026.

[8] SecurityWeek. "LLMs Hijacked, Monetized in 'Operation Bizarre Bazaar'." SecurityWeek, January 29, 2026.

[9] The Hacker News. "Attackers Use LLM Agent for Post-Exploitation After Marimo CVE-2026-39987 Exploit." The Hacker News, May 29, 2026.

[10] Cloud Security Alliance AI Safety Initiative. "LLMjacking: AI Model Hijacking Reaches Black Market Scale." CSA Lab Space, March 15, 2026.

[11] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 6, 2025.

[12] OWASP Gen AI Security Project. "OWASP GenAI Exploit Round-up Report Q1 2026." OWASP, April 14, 2026.

[13] Kaspersky. "LLMjacking: What These Attacks Are and How to Protect AI Servers." Kaspersky, May 12, 2026.