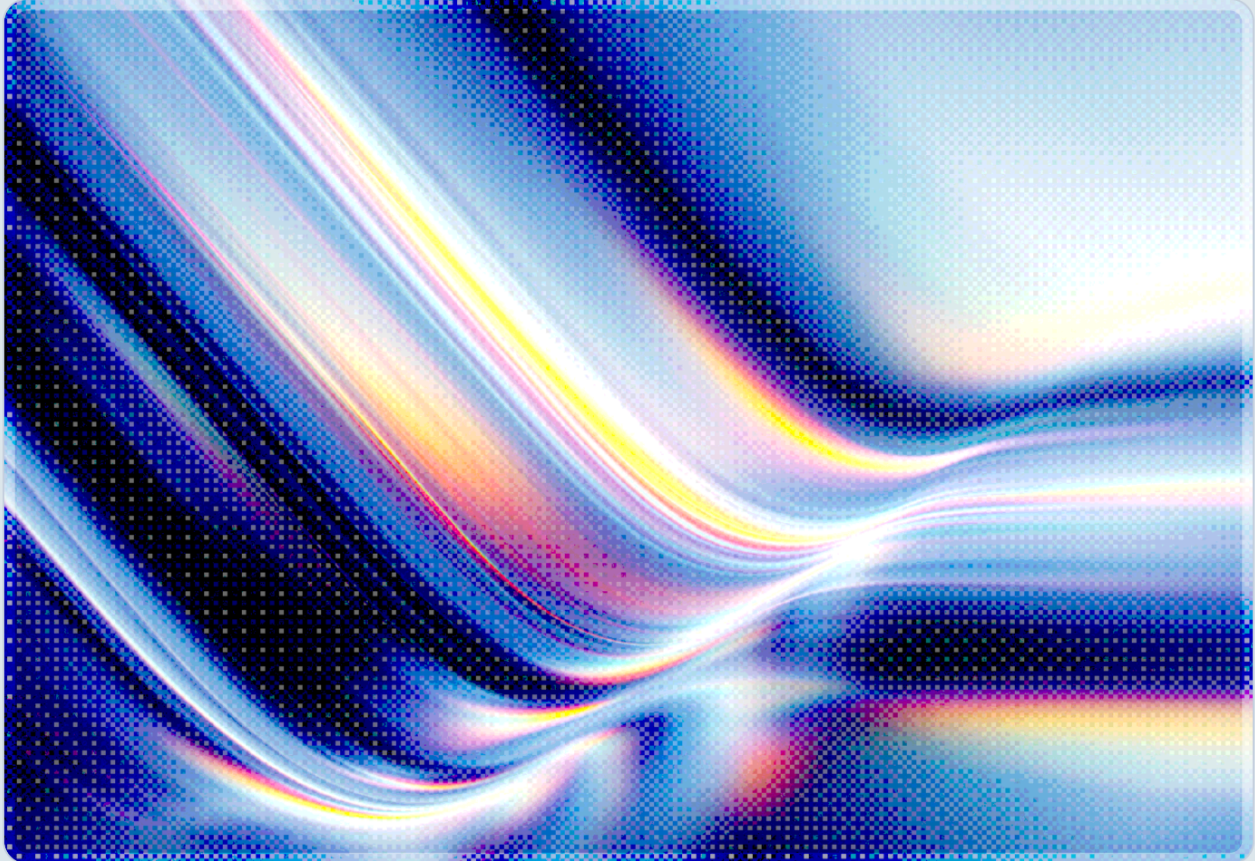


Skynet-Mode LLMjacking: Exposed Model Endpoints as Autonomous Attack Engines

Security Implications and Guidance for Operators of Self-Hosted Models, AI Gateways, and MCP Servers

2026-10-25

 AI-assisted Rapid Research



CSAI

cloud
security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

On October 21, 2026, Symantec's Threat Hunter Team published its analysis of an autonomous offensive agent that its operators called "Skynet" [1][2]. The researchers captured the agent's full instruction set in an AI honeypot. The agent searches the internet for model endpoints that accept OpenAI-style requests without authentication, or that accept a stolen key. It is designed to use the hijacked model as the reasoning engine for a campaign of scanning, exploitation, credential theft, and cryptocurrency mining against separate victims [1][3]. The configuration authorized the agent to run for up to 100,000 steps without human approval and defined roughly 74 offensive tools [2][3].

The central feature of the design is that the model plans the attack while someone else pays for it. The hijacked endpoint supplies planning and decision-making, and the agent's own modules carry out the attack. The endpoint owner pays for the inference [1][3], and traffic that originates from its infrastructure may raise reputational and legal questions for the owner. The evidence, however, shows intent rather than completed attacks. The honeypot never ran the agent's tool calls, and Symantec reports no evidence that the agent was used to compromise servers. Symantec did note signs that the kit had been run against a live target, with real-looking scan output, and researchers described it as "more than a prototype" [1][4].

None of the individual techniques is new; what is new is how they are combined. The jailbreak layers were largely off-the-shelf. The novelty lies in wiring them into a single agent that hands planning to a borrowed model and runs unattended at scale, which lowers the skill needed to mount broad exploitation campaigns [1][3]. The case also continues a development CSA has been tracking. The Skynet agent follows Sysdig's June 2026 finding that an exposed Ollama server drove the nine-phase "VAPT" exploitation framework [5]. Two cases do not make a trend, but together they suggest that stolen compute is becoming an operational component of attacks rather than only a commodity for resale.

Most of the agent's targets are endpoints with no authentication, which makes the defensive response largely a matter of configuration and asset inventory. Authenticating every model endpoint, gateway, and MCP server, restricting their network exposure, protecting and rotating provider API keys, and monitoring inference traffic for offensive content would sharply reduce the compute available to agents of this kind.

Background

LLMjacking started as cost theft. Sysdig's Threat Research Team coined the term in May 2024. In the case it described, attackers exploited a vulnerable Laravel instance (CVE-2021-3129), took AWS credentials from it, and used them to reach hosted models across ten cloud AI services, including Anthropic, AWS Bedrock, Google Cloud Vertex AI, Mistral, and OpenAI [6]. Sysdig estimated that this kind of abuse could cost a victim more than \$46,000 a day in model usage [6]. In this early phase, the attackers wanted to resell model access to other criminals. The victim's main harm was financial, plus the risk that legitimate workloads would be locked out once quotas ran out.

The attack surface has since moved from stolen cloud credentials toward model servers that need no credentials at all. Ollama is a widely used runtime for serving models locally, and it has no built-in authentication [5]. In January 2026, SentinelOne SentinelABS and Censys reported roughly 175,000 exposed Ollama hosts across 130 countries, observed over 293 days, and found that a persistent core of about 23,000 hosts accounted for most observations [7]. CSA's May 2026 analysis of Intruder's internet-scale scan reported that 31% of directly queried Ollama servers (about 1,652 of more than 5,200) answered test prompts without authentication [8]. CSA's September 2026 research note on exposed self-hosted AI endpoints went further: of

36,769 reachable endpoints, only 2.02% returned an HTTP authentication challenge [17]. Criminal operators have since built commercial infrastructure around this access. Pillar Security's "Operation Bizarre Bazaar" honeypots captured about 35,000 attack sessions against exposed AI infrastructure between December 2025 and January 2026, and that campaign linked internet scanning, automated validation of working endpoints, and a commercial marketplace that resold the hijacked access [9].

In June 2026, Sysdig documented a different use of LLMjacking. It observed an attacker using an exposed, unauthenticated Ollama server as the reasoning engine for a custom exploitation framework called VAPT. The framework ran nine phases, from service fingerprinting through vulnerability matching, proof-of-concept synthesis, credential extraction, and privilege escalation [5]. Rather than reselling the stolen compute, the attacker used it to drive an offensive tool. Sysdig recorded a session on June 12 lasting about eight and a half hours, followed by three more sessions from three IP addresses on June 14 [5]. CSA covered this development in a series of rapid research notes [10][11], which also flagged an open question: was VAPT an isolated case or the start of a pattern? The Skynet agent is a second, independently captured instance, which strengthens the case that VAPT was not isolated.

Security Analysis

Anatomy of the Skynet Agent

Symantec's honeypot imitates an unprotected, internet-facing AI service. Between its setup and the report's publication, it logged more than 127,000 external requests from tens of thousands of addresses [3][4]. In summer 2026, one of those clients sent what its operator believed was a usable model service a complete agent configuration [2][3]. Because the honeypot answered as a model and did not run any tools, Symantec recovered the full instruction stack without any downstream victim being touched.

According to The Register's account of the Symantec findings, the instruction stack is organized in layers that turn a general-purpose model into a compliant attack planner [1]. Layers one and two hide the model's real identity and recast reading secrets and configuration files as routine development work. Layers three and four cast the model as a devoted companion and tell it to ignore its own safety warnings. Symantec researcher Priti More noted that the persona and phrasing used here circulate publicly in English-language jailbreak communities [1]. Layer five defines the toolset. The final directive turns on "Skynet mode," authorizes up to 100,000 unattended steps, and instructs the agent to deploy cryptocurrency miners quietly [1]. Infosecurity Magazine also reports a separate obfuscated script that installs Monero-mining software, hides its presence, and erases logs [3].

The toolset reveals what the operators wanted. The roughly 74 tools cover internet-wide asset discovery, botnet command-and-control, database dumping through Prisma, and exploit modules for VS Code, Django, Jupyter, Laravel, Supabase, Next.js, and Flask/Werkzeug [1]. The toolset also includes GraphQL and command injection, plus server-side request forgery through HTML-to-PDF generators and a dedicated SSRF tool [1]. These are widely used developer-facing frameworks, and some of them, such as Jupyter and VS Code server, are commonly deployed in the same AI development environments as self-hosted model servers. Skynet mode is designed to close the loop: scan for targets, infect them, deploy miners, and reinfect hosts [1][2].

Three Parties, Three Distinct Harms

Skynet-style LLMjacking spreads harm across parties who have no relationship with one another. The table below summarizes the roles described in or implied by Symantec's findings and the related reporting; the commercial model provider row reflects CSA's analysis of the stolen-key and ban-evasion statements in that reporting rather than an observed Skynet behavior.

Party	Role in the Attack	Primary Harm
Endpoint owner (self-hosted model, AI gateway, MCP server, or holder of a stolen API key)	Unwittingly supplies the reasoning engine that plans each step	Inference and GPU costs; reputational and possible legal exposure for traffic that originates from its infrastructure; capacity exhaustion
Downstream victim (web application or server running a targeted framework)	Target of scanning, exploitation, and data theft	Database and credential theft, remote code execution, cryptominer installation, reinfection
Commercial model provider (where jailbreaks reach hosted models via stolen keys)	Model safety controls are the target of layered jailbreaks	Abuse of its platform; enforcement complicated because the operator is hidden behind a third party's account

For the operator, this arrangement offers both savings and anonymity. Symantec's researchers observed that "running a model always costs someone money." Routing inference through someone else's endpoint pushes that cost onto the victim and helps the operator avoid the account bans that providers impose for abuse [3]. The operator's own infrastructure plays only a small part, since the model's planning happens on the hijacked endpoint and its outputs are executed against third-party targets. This complicates attribution. The French-language indicators Symantec observed, including challenge text, system-prompt language, and a French residential IP address on the first day, were described by the researchers as weak, possibly borrowed signals [1].

Why Autonomy Changes the Risk Calculus

Resale-era LLMjacking was largely bounded by human attention. An operator reselling access, or manually prompting a stolen model, could run only so many sessions. VAPT already automated much of the exploitation workflow within a fixed pipeline, and a configuration that authorizes 100,000 unattended steps extends that automation into open-ended autonomy [2]. In practice, a single hijacked endpoint could drive a continuous scan-exploit-mine cycle for as long as the endpoint stays reachable. Symantec's observation that "an operator can hand the model a goal and let it work" captures the shift [3]. Exploitation skill moves from the operator into the agent's tool definitions and the borrowed model's reasoning.

The comparison with VAPT shows how quickly the technique is developing. VAPT embedded the model inside a deterministic pipeline. Sysdig described code that parsed model output, executed requests, and confirmed success with marker strings, while the model handled structured decisions at each phase [5]. Skynet gives the model wider latitude, running as an open-ended agent with a large tool catalog and a self-sustaining infection loop. Symantec's honeypot did not observe it in operation, so CSA cannot assess how well this design works in practice. Still, the two cases together point to a design trend in which attackers treat any reachable model, open-weight or commercial, as interchangeable reasoning capacity. Sysdig noted that VAPT requested commercial model names such as "gpt-4o-mini" and "claude-3-5-sonnet" from an Ollama server [5]. CSA infers from this that the framework had been repointed from paid APIs to free stolen capacity.

Target Surface Beyond Ollama

Defenders should not read Skynet as an Ollama-only problem. Infosecurity Magazine reports that the agent hunts for self-hosted models, AI gateways, and Model Context Protocol (MCP) servers [3], and The Register notes that it also accepts endpoints reachable with stolen keys [1]. Because AI gateways typically hold upstream credentials for several commercial providers, an exposed gateway is likely to be a particularly attractive target: one unauthenticated proxy can expose frontier-model capacity billed to the gateway owner's accounts. An exposed MCP server raises a different concern, since it may let a hijacked model reach

files, databases, or internal APIs on the owner's side. Pillar Security described the same category of MCP endpoints connecting models to file systems and internal APIs as part of the Bizarre Bazaar target set [9]. The Skynet configuration Symantec captured was aimed at third-party victims. Even so, the same access patterns could be turned inward against the endpoint owner.

Indicators

Symantec's published indicators were relayed by AlienVault OTX and summarized by OffSeq Threat Radar [12]. Organizations should validate these against Symantec's original report before operational use. Defenders should also treat behavioral signals as more durable than atomic indicators, because mining pools and source addresses rotate easily.

Type	Indicator	Context
Domain / URL	mine.c3pool[.]com (port 13333)	Monero mining pool endpoint referenced by the agent
Domain / URL	auto.c3pool[.]org (port 80)	Monero mining pool endpoint
Domain / URL	auto.skypool[.]xyz (port 6666)	Mining pool endpoint
IP address	86.216.147[.]228	Source IP listed in the OTX indicator set [12]
SHA-256	b720a5e7c3a5b1760d15a8833a782f4fe7fc5c62b967fe57f607e72745ec0ccf	Associated file hash
SHA-256	fcde1323acceba46fd88f2276c010bb98839dccce03dc512c56d9529428ceec68	Associated file hash

The indicator set does not describe the role of each file, and the published sources do not tie the listed IP address to the French-origin signals discussed above, which Symantec characterized as weak in any case [1].

Recommendations

Immediate Actions

Organizations should start by finding out whether they are already supplying compute to attackers. Security teams should scan their own address ranges and cloud accounts for reachable model servers. This means checking Ollama's default port 11434, OpenAI-compatible API ports, AI gateway listeners, and MCP servers, and confirming that each one requires authentication [5]. Any endpoint that answers an unauthenticated prompt should be taken off the public internet at once by binding it to localhost or an internal interface, or by placing it behind an authenticating reverse proxy. Ollama has no native authentication, so the control has to be added at the proxy or network layer [5]. Teams should also rotate any provider API keys that are stored on, or reachable from, exposed gateways and development hosts, because Skynet also uses stolen keys [1].

The second immediate priority is looking back for signs of past abuse. Teams should review inference logs for unexplained spikes in request volume, long-running sessions from unfamiliar addresses, and prompts that contain jailbreak personas, tool-definition schemas, or instructions about scanning, exploitation, or mining. Sysdig specifically recommends watching for structured-output contracts and marker-bracketed command patterns in model traffic [5]. Hosts running the framework types in Skynet's tool catalog should be checked for cryptominer processes, unexpected outbound connections to mining pools, and cleared logs. This is especially important for Jupyter, VS Code server, and Next.js or Flask development instances exposed to the internet.

Short-Term Mitigations

Over the following weeks, organizations should treat model inference as a governed, metered service rather than a background utility. Each model endpoint and gateway should require per-client credentials, enforce rate limits and spending caps, and log enough detail to attribute usage to a known workload. Symantec's guidance points the same way, emphasizing authentication of AI endpoints and reduced network exposure [2][12]; as reported, it also covers protecting API keys, restricting agent privileges and outbound connectivity, and requiring approval for sensitive tool actions [2]. Spending alerts on cloud AI accounts give a cheap backstop for detecting stolen-key abuse, and quota exhaustion should be investigated as a security signal.

MCP servers and agent frameworks deserve particular attention, because they turn a model compromise into an action compromise. MCP servers should not be exposed to untrusted networks. Each one should grant the narrowest set of tools and data needed for its function, and destructive or exfiltration-capable tools should require human approval. Egress filtering on hosts that run models or agents, including blocking known mining pool protocols and destinations, limits what a hijacked or compromised system can do. Development environments such as Jupyter and VS Code server should come under the same exposure management and patching standards as production applications, because Skynet's tool catalog targets them explicitly [1].

Strategic Considerations

Over the longer term, the Skynet case makes the argument for treating AI compute as an asset class with its own inventory, ownership, and threat model. Shadow model servers set up by individual teams are a likely source of unauthenticated exposure, so asset discovery should cover model runtimes, gateways, and MCP servers alongside conventional workloads. Organizations should also plan for liability and incident response in a scenario where their infrastructure was the reasoning engine for attacks on third parties. Questions about notification, log retention, and cooperation with law enforcement are easier to settle before an incident than during one.

The case also has implications for model providers and the wider ecosystem. Operators considered layered jailbreaks built from publicly circulating personas worth building into their tooling [1], though the honeypot capture does not show how well they work against current production models. Agentic traffic patterns could become a useful detection signal, including very long unattended loops, large tool catalogs in system prompts, and repeated exploitation planning. Providers may be able to flag this kind of abuse even when it arrives through legitimate-looking API keys. Finally, AI honeypots captured complete attacker instruction sets in this case and in academic work such as the Ollure honeypot study, which recorded 290,887 interactions over 84 days [13]. Security teams with mature detection programs may want to consider deploying them.

CSA Resource Alignment

CSA's rapid research on the June 2026 VAPT incident is the direct predecessor to this note. *LLMjacking Evolved: Stolen Compute as Offensive AI Infrastructure* [10] and *LLMjacking Evolved: Stolen AI Compute as Offensive Infrastructure* [11] documented the shift from reselling stolen model access to using it in production for offensive tooling, and both asked whether VAPT was an isolated case. The Skynet agent adds a second, independently captured instance and widens the analysis in three ways: from a deterministic pipeline to an open-ended autonomous agent, from Ollama servers to AI gateways and MCP servers, and from exploitation alone to a designed self-sustaining infection and mining loop. Readers should treat the recommendations in those notes, particularly on hardening model-serving infrastructure, as still current and as reinforced by this case.

CSA's research notes on exposed self-hosted AI infrastructure supply the exposure data behind the Skynet threat. *Exposed AI Infrastructure: Self-Hosted Services Under Attack* [8] reported that 31% of directly queried Ollama servers in Intruder's scan answered prompts without authentication, and 36,769 *Exposed AI Endpoints: A Self-Hosted Supply Chain Crisis* [17] found that only 2.02% of reachable self-hosted AI endpoints returned an authentication challenge. These findings describe exactly the conditions this agent is built to find. Related CSA analysis of the NadMesh botnet [14] showed the same exposed AI services being used as entry points for harvesting cloud and model credentials. Read together, these notes show exposed AI infrastructure being exploited in three ways: as a credential source (NadMesh), as resaleable access (Bizarre Bazaar), and now as an attack planner (VAPT and Skynet).

For control mapping, the CSA AI Controls Matrix (AICM) v1.1 [15] provides the governing framework. Its Identity and Access Management controls address authentication of model endpoints and gateway credentials. Its Threat and Vulnerability Management controls cover exposure scanning of AI services and patching of development frameworks, while its Logging and Monitoring controls support the inference telemetry needed to detect hijacked usage. CSA's MAESTRO agentic threat modeling framework [16] is a useful lens for the agent itself. It helps organizations reason about how a hijacked model, an attacker-supplied tool layer, and an unsupervised execution loop interact across layers. It also helps them ask the inverse question: how would their own agent deployments behave if an adversary controlled the model's instructions?

References

- [1] Lyons, Jessica. "[Autonomous attack agent in 'Skynet mode' hijacks other models to carry out its evil deeds.](#)" The Register, October 24, 2026.
- [2] Symantec Threat Hunter Team. "[Autonomous LLMjacking: 'Skynet' Agent Hijacks Exposed AI Models to Run Attacks.](#)" Symantec (Broadcom), October 21, 2026.
- [3] Poireault, Kevin. "[Symantec Uncovers AI Agent Hijacking Models for Autonomous Cyber-Attacks.](#)" Infosecurity Magazine, October 21, 2026.
- [4] Let's Data Science. "[Symantec Details Agent That Hijacks Exposed AI Models for Cyberattacks.](#)" Let's Data Science, October 2026.
- [5] Clark, Michael. "[LLMjacking Evolved: Attackers Are Using Stolen AI Compute to Build Offensive Agentic Tools.](#)" Sysdig, June 17, 2026.
- [6] Brucato, Alessandro. "[LLMjacking: Stolen Cloud Credentials Used in New AI Attack.](#)" Sysdig, May 6, 2024.
- [7] SecurityWeek. "[175,000 Exposed Ollama Hosts Could Enable LLM Abuse.](#)" SecurityWeek, January 30, 2026.
- [8] Cloud Security Alliance. "[Exposed AI Infrastructure: Self-Hosted Services Under Attack.](#)" CSA AI Safety Initiative, May 2026.
- [9] Pillar Security Research. "[Operation Bizarre Bazaar: First Attributed LLMjacking Campaign with Commercial Marketplace Monetization.](#)" Pillar Security, January 28, 2026.
- [10] Cloud Security Alliance. "[LLMjacking Evolved: Stolen Compute as Offensive AI Infrastructure.](#)" CSA AI Safety Initiative, June 26, 2026.
- [11] Cloud Security Alliance. "[LLMjacking Evolved: Stolen AI Compute as Offensive Infrastructure.](#)" CSA AI Safety Initiative, June 20, 2026.
- [12] OffSeq Threat Radar. "[Autonomous LLMjacking: 'Skynet' Agent Hijacks Exposed AI Models to Run Attacks.](#)" OffSeq (source: AlienVault OTX), October 21, 2026.
- [13] Elzer, Karina, Niklas Netterstrøm Johansen, and Emmanouil Vasilomanolakis. "[OllamaDrama: Designing and Deploying a Honey pot to Measure Attacks on Exposed LLM Infrastructure.](#)" arXiv:2609.29757, September 2026.
- [14] Cloud Security Alliance. "[NadMesh: A Botnet Built to Hunt Exposed AI Infrastructure for Cloud Keys.](#)" CSA AI Safety Initiative, July 20, 2026.
- [15] Cloud Security Alliance. "[AI Controls Matrix \(AICM\) v1.1.](#)" Cloud Security Alliance, June 22, 2026.
- [16] Huang, Ken. "[Agentic AI Threat Modeling Framework: MAESTRO.](#)" Cloud Security Alliance, February 6, 2025.
- [17] Cloud Security Alliance. "[36,769 Exposed AI Endpoints: A Self-Hosted Supply Chain Crisis.](#)" CSA AI Safety Initiative, September 15, 2026.