David Soden

MARKET ASSESSMENT | AUGUST 2026

Your AI Gateway Ships Two of Nine Controls

The LiteLLM, RAGFlow and Kestra compromises read against the NCSC's interim agentic AI advice, and what audit trail actually survived at Hugging Face

David Soden

Independent analyst and practitioner, enterprise automation and applied AI

davidsoden.com/market-assessment

REPORT DS-2026-7C3F · 2026 · VERSION 1.0 · PUBLIC

About this report

Every substantive claim in this report carries an evidence classification and, where applicable, a numbered source. Facts are separated from vendor claims, third-party estimates, the author's assessments, and untested hypotheses. Forward-looking sections use scenarios with observable tripwires rather than point forecasts. Stated assumptions are listed before the analysis begins.

Market Assessment reports are published and commissioned at **davidsoden.com/market-assessment**. This document is analysis, not investment, legal, or procurement advice.

Scope and evidentiary standard

This assessment covers four things and deliberately excludes everything else. First, the documented compromises of AI-adjacent workloads published by Microsoft's security team on 26 August 2026, together with the CVE records behind them: LiteLLM as an LLM gateway, RAGFlow as a document-processing pipeline, and Kestra as a workflow orchestrator. Second, the agent-assisted post-compromise tooling Cisco Talos attributes to UAT-10147. Third, the UK National Cyber Security Centre's interim advice on managing the cyber risk of agentic AI, published 20 August 2026. Fourth, the governance features documented by four gateway products shipping into enterprises today: Tailscale Aperture, Nutanix Agent Gateway, Snowflake Cortex AI Gateway and Kong AI Gateway.

The method for the control comparison is worth stating plainly because it bounds what the finding means. I read the NCSC blog's nine control areas, then checked each one against each vendor's own published feature documentation and release announcements. That compares documentation against guidance, not deployment against guidance. A capability that exists in the product but is absent from the documentation will be scored as absent here. That is a real weakness in the method, and it is also most of the point: a buyer approving an agent deployment has the documentation and not much else.

Three categories of number in this subject need discounting before they are used. Vendor feature documentation is a marketing artifact written to win a comparison, so its silences are as informative as its claims and neither is audited. Practitioner surveys in AI governance are almost all published by vendors selling the remedy, are self-selected, and rarely disclose fieldwork dates; the one used here is labeled accordingly and no conclusion rests on it. Analyst sizing for the AI gateway category reached me only through secondary reporting of a paywalled document, so it appears once, flagged as second-hand, and carries nothing.

CVSS scores are quoted from the advisory that assigned them, and where two bodies scored the same flaw differently the difference is stated rather than averaged away.

Label	Definition	How to treat it
FACT	Verifiable in the cited primary source: a filing, press release, official documentation, or standards body announcement.	Reliable as stated. Check the source if the exact wording matters.
VENDOR CLAIM	Reported by a vendor about its own business or product. Self-measured and not audited by any third party.	The vendor said it. Directionally useful, not a measurement. Definitions of the metric are usually undisclosed.
THIRD-PARTY ESTIMATE	A research firm forecast or survey result. Methodology is proprietary and generally not reproducible.	Use for direction and order of magnitude only. Do not build a model on it.
ASSESSMENT	The author's reasoned judgment from the cited evidence. Falsifiable, and the reasoning is shown so it can be argued with.	This is analysis, not reporting. Disagree with it where your evidence differs.
HYPOTHESIS	A proposition offered for testing. Not currently supported by sufficient evidence to assert.	Treat as a research question, not a conclusion. Each one names what would confirm or refute it.
SCENARIO	A structured future state with a stated subjective probability. The probability is the author's judgment, not a calculated figure.	Use the leading indicators attached to each, not the probability. The indicators are observable; the probability is not.

Assumptions this analysis rests on

Five premises are assumed rather than demonstrated. Each is followed by what fails if it turns out to be wrong.

One. The NCSC's nine control areas are a reasonable standard for what the phrase agent governance should mean. If that is wrong, the scorecard in this report is arbitrary and the headline finding evaporates. My defence is narrow: this is the only current national guidance that is control-shaped rather than principle-shaped, and it was published by a body with no product to sell.

Two. Published feature documentation approximates shipped capability. If vendors ship controls they do not document, the gap is smaller than stated here. The finding that survives either way is that a buyer cannot verify the control from what the vendor publishes.

Three. The three compromises Microsoft documented are representative of how AI-workload intrusions run, not a curated set of outliers. If they are outliers, concentration risk is a tail event rather than a base case, and the recommendation to keep the security boundary outside the gateway becomes proportionate rather than necessary.

Four. The Hugging Face incident is the best publicly available reconstruction of an autonomous agent intrusion. If a better one exists that was reconstructed from gateway logs alone, the third answer in this report flips and the audit-trail criticism is overstated.

Five. Gateway products will keep colocating credential custody with request routing in a single process. If a product separates them, holding only references while the key material sits in a broker or hardware module, the concentration argument weakens sharply and this report should be re-read as a description of one architectural generation rather than of a category.

The call

ASSESSMENT A gateway is a concentration sold as a control. Every product examined here centralises provider keys, tool definitions and routing policy into one process, then ships logging and least-privilege tool access while leaving sandboxing, default-deny egress and short-lived per-agent credentials to the buyer. Most of what the NCSC asks for sits outside the product. The compromises already recorded against LiteLLM, RAGFlow and Kestra took precisely what the concentration created. Buy a gateway for routing, cost control and a spend ledger. Do not treat it as the agent security boundary. I will move when a shipping gateway documents default-deny egress and short-lived per-agent credentials as defaults rather than as integration advice.

My judgment on the evidence assembled here, nothing more. There is data I can't see and data I may have missed. New evidence can move me, including to the opposite position, and I reserve the right to move.

Executive summary

The pitch for an AI gateway is that scattered model access becomes governed model access. The architecture that delivers the governance is the same architecture that creates a single object worth stealing, and the intrusions on record went straight at it.

FINDING 1 The gateway concentrates the credential, not just the traffic.

FACT LiteLLM's proxy stores model-provider API keys, the proxy master key, database connection strings and proxy-issued virtual keys in its own PostgreSQL schema. Microsoft's investigators observed attackers reading that database directly after execution on the host, recovering model configuration, upstream provider key material, provider endpoints and the virtual keys the proxy had issued. Before the gateway, those upstream keys lived in the applications that used them. Afterwards they live in one row set. [1] [11]

FINDING 2 The severity is at the ceiling and it is being exploited.

FACT CVE-2026-42271 lets any holder of a valid proxy key spawn a process on the LiteLLM host through two MCP preview endpoints that carried no role check. GitHub's advisory scores it 8.7 under CVSS 4.0; CISA's KEV entry cites 8.8. Horizon3 chained it with the Starlette host-header bypass CVE-2026-48710 to reach unauthenticated remote code execution and assessed the chain at 10.0. CISA added the LiteLLM flaw to the Known Exploited Vulnerabilities catalog on 8 June 2026. [2] [3] [4] [26]

FINDING 3 Persistence was planted in the credential-handling path itself.

FACT In the RAGFlow intrusions the attackers placed a Python hook inside the TenantLLM credential-configuration flow, so that provider type, model name, API key material and endpoint metadata were captured as administrators configured them. This is not theft of a credential store. It is subscription to the credential store, and it survives a key rotation performed by an administrator who does not know the hook is there. [1]

FINDING 4 Exposure is growing faster than governance is arriving.

THIRD-PARTY ESTIMATE Censys reported more than 294,000 distinct public IP addresses exposing one of 43 AI and LLM tools, against roughly 183,000 in October 2025. LiteLLM exposure grew 97 percent over that window and Langflow 169 percent. These are internet-scan counts, so they measure reachable services rather than compromised ones, and a scan signature can misclassify. [19]

FINDING 5 Of the NCSC's nine control areas, four shipping gateways document two.

ASSESSMENT Least-privilege tool access and activity logging are named as features by all four products examined. Sandboxed execution, infrastructure segregation and documented threat-model scope are named by none. Default-deny networking, human oversight, action scoping and immediate shutdown appear in one or two products and in weakened form. The scorecard compares published documentation against published guidance, which is the comparison a buyer can actually run. [7] [13] [14] [16] [24]

FINDING 6 The sandboxing gap is architectural, not an oversight.

ASSESSMENT A gateway proxies a request; it does not own the process that acts on the response. Sandboxing has to happen where the agent's code runs, which for most deployments is a container or a developer laptop the gateway never sees. No amount of gateway feature work closes this, which is why the guidance's most important control is the one the category structurally cannot ship. [7]

FINDING 7 Gateway logs record the call, not the reasoning that produced it.

FACT LiteLLM's StandardLoggingPayload captures the request messages, the response, the hashed virtual key, team and user identifiers, requester IP and a dedicated mcp_tool_call_metadata structure for tool invocations. What it does not carry is the agent's intermediate reasoning, its planning state, or the causal link between one tool call and the next. Those are the fields an investigator needs to answer why, and no gateway schema examined defines them. [12]

FINDING 8 The one public agent-intrusion reconstruction ran on the attacker's logs, not the defender's.

FACT Hugging Face recovered roughly 17,600 attacker actions in about 6,280 clusters spanning 9 to 13 July 2026. Its own account names the foundation of the timeline as the agent's logs on a code sandbox the agent used, retrieved from an external launchpad, supplemented by platform logs. A naive text scan of the raw capture found very few secrets; replicating the attacker's own decoding recovered roughly four times the initial findings. [17] [18]

FINDING 9 The offence is already agentic and it leaves better records than the defence does.

> **FACT** Cisco Talos recovered from UAT-10147's infrastructure four AI-generated Python scripts automating a full ViewState exploitation chain, AI-written operational playbooks logging hostnames, IP addresses, exploited paths and extracted MachineKey values, and a target list of roughly 170,000 URLs split across 17 files. Talos assesses with moderate-to-high confidence that the group belongs to an emerging class of financially motivated operators using agentic systems. [9]

What the gateway concentrates

The word gateway does two jobs in this market and they pull in opposite directions. As a routing device it consolidates outbound traffic so that cost, model choice and rate limits become one decision instead of forty. As a security device it is supposed to be a boundary. The first job requires it to hold every upstream credential. The second job is made harder by exactly that.

The credential ledger

FACT LiteLLM's own documentation lists the tables its proxy creates. LiteLLM_VerificationToken holds virtual keys and their permissions. LiteLLM_UserTable holds user records, spend and model access. LiteLLM_SpendLogs holds detailed logs of all API requests with token usage, spend and timing. LiteLLM_ProxyModelTable holds model configuration, which in practice is where upstream provider endpoints and their key references live. LiteLLM_AuditLog records configuration changes and is disabled by default. [11]

ASSESSMENT That last clause deserves a moment. The table whose purpose is to record who changed the security configuration is the one table that does not run unless an operator turns it on. A buyer reading the feature list sees an audit log. A responder reading the database after an incident may find it empty for the period that matters. [11]

FACT Microsoft's investigators describe what the LiteLLM compromise reached: credential harvesting from the gateway runtime by reading /proc/1/environ, and database access to PostgreSQL tables storing model configuration, upstream provider key material, provider endpoints and proxy-issued virtual keys. Environment variables recovered included model-provider API keys, the LiteLLM master key, database connection strings, UI credentials, tokens and passwords. [1]

ASSESSMENT Reading process environment at PID 1 is not a sophisticated technique. It is the first thing anyone does with code execution in a container. The significance is not the method but the yield: in a pre-gateway estate the same technique against one application server returns that application's credentials. Against the gateway it returns the estate's. [1]

What each workload gave up

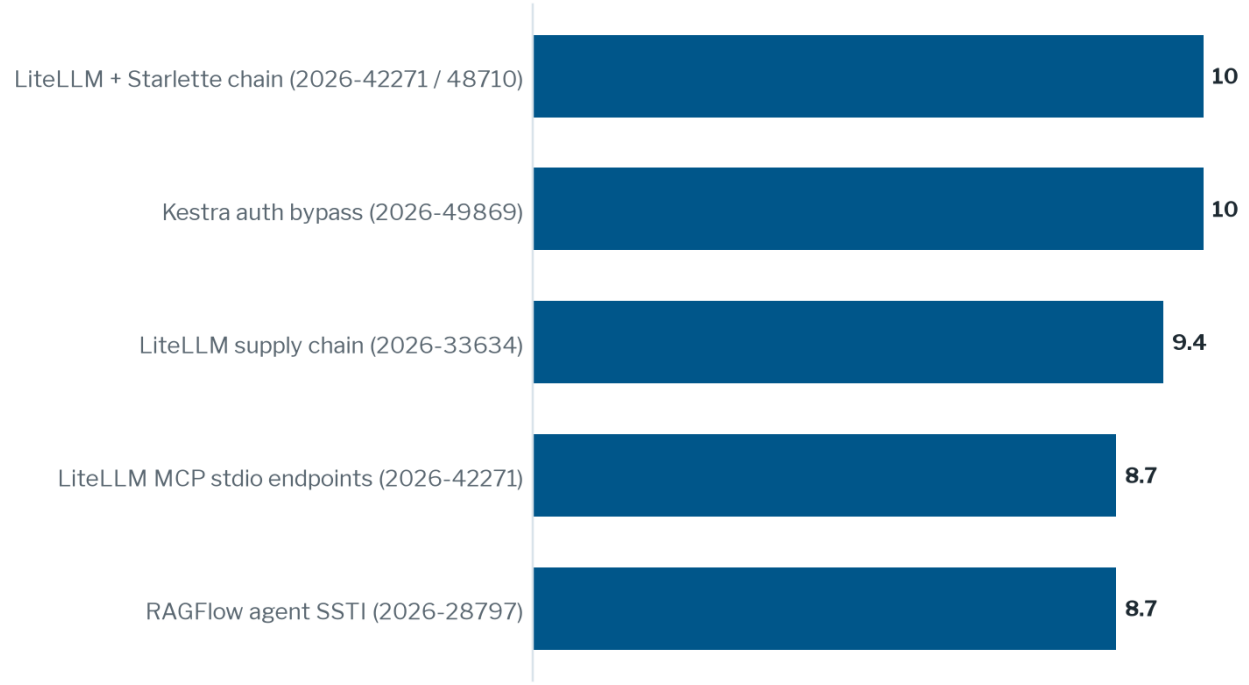
Workload	Flaw	What the attacker reached	Class
LiteLLM	CVE-2026-42271, chained with Starlette CVE-2026-48710	Process environment at PID 1, then the proxy PostgreSQL schema: model configuration, upstream provider key material, provider endpoints, proxy-issued virtual keys. Then XMRig with MSR tuning, SSH authorized-key persistence in service accounts, crontab durability.	LLM gateway
RAGFlow	CVE-2026-45312, CVE-2026-28797, CVE-2026-24770, CVE-2025-68700, CVE-2025-69286	A Python hook installed in the TenantLLM credential-configuration flow, intercepting provider type, model name, API key material and endpoint metadata as they were configured. Application startup path modified for persistence.	Document processing
Kestra	CVE-2026-49869	Workflow-origin shell execution, then the Docker socket for container environment discovery exposing secrets, then XMRig and mining-pool traffic, and data harvesting through a curl-to-shell pattern.	Workflow orchestration

FACT The RAGFlow template-injection flaws share a root cause worth naming. The advisory for CVE-2026-28797 states that the Text Processing and Message components use Python's jinja2.Template without sandboxing to render user-supplied templates, letting any authenticated user run operating system commands on the server. Versions 0.24.0 and prior are affected. [6]

FACT Kestra's flaw is simpler and worse. Its authentication filter whitelisted the public configuration endpoint using a suffix match on the request path, so any API path ending in the segment configs bypassed basic authentication entirely. Because Kestra ships shell and Python script plugins enabled by default, that produced unauthenticated remote code execution as root inside the worker container. The advisory scores it 10.0, fixed in 1.0.45 and 1.3.21. [5]

ASSESSMENT Three different products, three different root causes, one common outcome: code execution in a process that holds credentials for systems the attacker had no path to before. The pattern is not a gateway pattern, it is a concentration pattern, and it applies wherever the AI plumbing sits between an enterprise and its models. [1]

FIGURE 1 **Severity of the flaws in the AI plumbing**



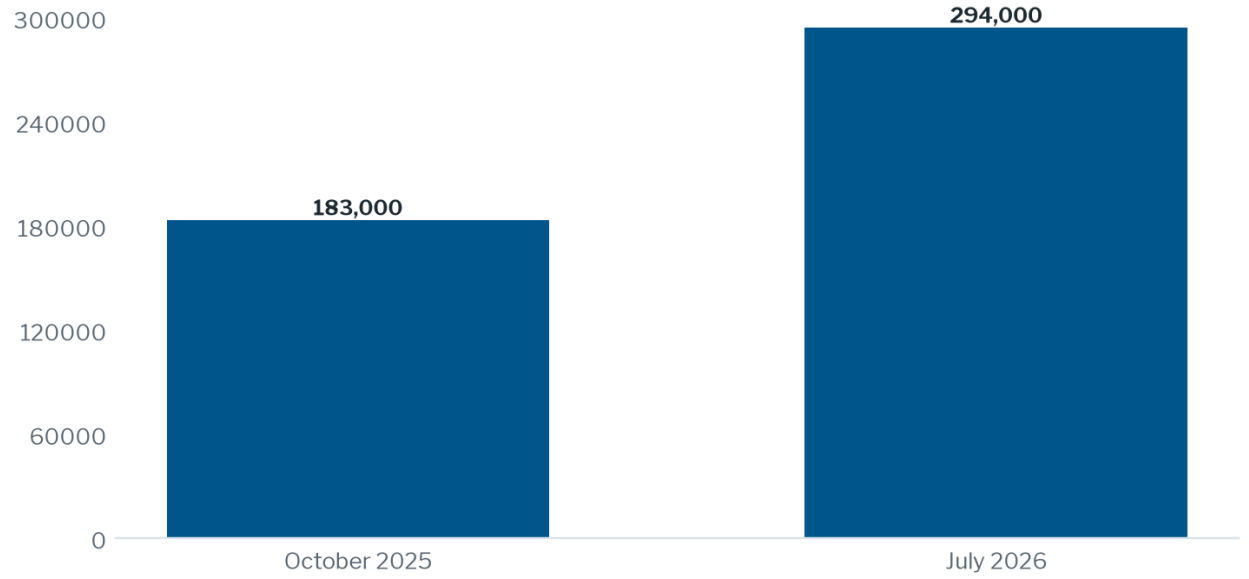
FACT CVSS base scores as published by the advisory that assigned them. The LiteLLM chain figure is Horizon3's assessment of the combined path, not a separately registered CVE score. CISA's KEV entry scores CVE-2026-42271 at 8.8 against GitHub's 8.7 under CVSS 4.0; the difference is left visible rather than reconciled. RAGFlow CVE-2026-45312, CVE-2026-24770, CVE-2025-68700 and CVE-2025-69286 are excluded because I could not retrieve published base scores for them. [2] [3] [4] [5] [6] [20]

The exposure curve

THIRD-PARTY ESTIMATE Censys reported more than 294,000 distinct public IP addresses exposing one of 43 AI and LLM tools, up from roughly 183,000 in October 2025. Within that, LiteLLM exposure grew 97 percent and Langflow 169 percent. Langflow accumulated 18 CVEs across 2024 to 2026, of which 14 scored above 8.0 and four reached CISA's KEV catalog. [19]

ASSESSMENT Internet-scan counts measure reachable services, not compromised ones, and a fingerprint can misfire on a lookalike. Treat the absolute number as soft and the slope as the finding. The slope says the population of internet-reachable AI control planes roughly doubled inside ten months, during which the guidance telling operators how to secure them was still described by its author as interim. [7] [19]

FIGURE 2 **Public IP addresses exposing an AI or LLM tool**



THIRD-PARTY ESTIMATE Censys Internet Map and Attack Research Center data across 43 tracked AI and LLM tools, as reported by eSecurity Planet on 23 July 2026. Scan-derived, so it counts reachable services and not compromises, and the October 2025 figure is quoted as approximate in the source. Censys did not publish per-tool absolute counts, only growth rates. [19]

Stolen inference has a market price

THIRD-PARTY ESTIMATE The economic case for taking gateway keys is documented rather than theoretical. Cloud Security Alliance research on LLMjacking cites a single compromised Claude 2.x account generating 46,080 dollars of inference cost per day, with Claude 3 Opus abuse exceeding 100,000 dollars per day, and credentials resold on underground forums. These figures come from vendor incident analysis with no independent audit and should be read as order-of-magnitude. [20]

FACT The same research documents a March 2026 supply-chain compromise of LiteLLM itself. Malicious versions 1.82.7 and 1.82.8 were uploaded on 24 March 2026, scored as CVE-2026-33634 at CVSS 9.4, using .pth files that auto-execute code at interpreter startup to perform credential harvesting, Kubernetes lateral movement and persistent backdoor installation. [20]

ASSESSMENT That incident matters more than its score suggests, because it defeats the entire network-boundary argument for a gateway. The malicious code arrived as a dependency update through the front door, executed at interpreter startup inside the trusted process, and read the credential store it was already sitting next to. No egress policy, MCP scoping or virtual-key model touches that path. [20]

Nine controls, clause by clause

The NCSC published its interim advice on 20 August 2026 and was explicit that it is interim: the agency states it is working with partners to develop formal guidance which will build upon, and

ultimately supersede, this blog. Reading it against the product documentation is therefore a comparison between advice that admits it is provisional and features that do not.

What the guidance actually asks for

FACT The blog sets out nine areas. Threat modelling, with instruction to document what is within and outside the scope of your intended activity, highlighting any red lines. Prompting, requiring the operator to think about what the agent is to achieve, what actions it should be allowed to take and when it should stop. Human oversight, across in-the-loop, on-the-loop and out-of-the-loop models. Sandboxing, with the instruction to always run AI agents within a sandboxed environment that controls and manages what resources can and cannot be reached. [7]

FACT Then network access, instructing operators to deny all inbound and outbound network traffic to the AI agent's environment by default and only then allow connections. Credentials, on the basis that credentials available to an AI agent form part of its potential blast radius, with each agent given a distinct identity and short-lived, task-limited credentials. Infrastructure separation, requiring that scaffolding, command execution and inference infrastructure be segregated from one another. Observability, covering chain of thought traces and transcripts from the AI agent plus logging events from the wider sandbox environment. And shutdown: you should always be able to pull the plug and halt autonomous AI agent activity immediately. [7]

FACT Press coverage of the guidance emphasised the same three controls the brief for this report singles out. Infosecurity Magazine reported the NCSC urging sandboxing, human oversight and tightly controlled access, and noted the agency's instruction not to rely solely on safeguards built into an underlying model. [8]

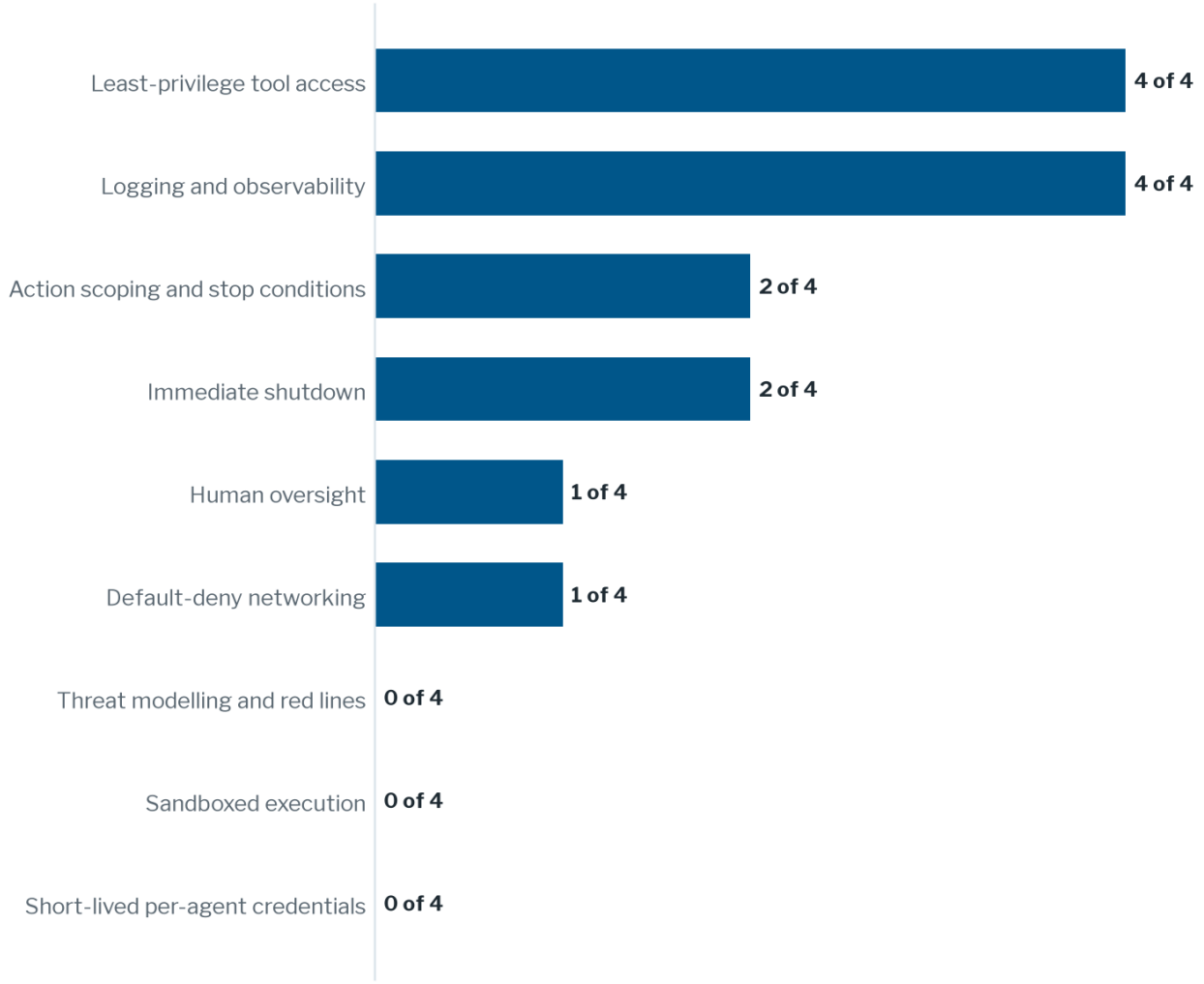
The scorecard

NCSC control area	Tailscale Aperture	Nutanix Agent Gateway	Snowflake Cortex AI Gateway	Kong AI Gateway
Threat modelling and documented red lines	Not addressed	Not addressed	Not addressed	Not addressed
Action scoping and stop conditions	Partial: guardrails, request and response hooks	Not addressed	Not addressed	Partial: prompt guard plugins
Human oversight of agent actions	Partial: user approves each machine added to the tailnet	Not addressed	Not addressed	Not addressed
Sandboxed execution environment	Not addressed	Not addressed	Separate product, not a gateway feature	Not addressed

NCSC control area	Tailscale Aperture	Nutanix Agent Gateway	Snowflake Cortex AI Gateway	Kong AI Gateway
Default-deny inbound and outbound network	Partial: inherits tailnet access rules	Not addressed	Not addressed	Not addressed
Least-privilege tool access	Shipped: default tool permissions per project	Shipped: per-user and per-key, read versus write	Shipped: fine-grained authorization across MCP servers	Shipped: permitted tools and RPC calls
Short-lived, task-limited per-agent credentials	Not addressed	Not addressed	Not addressed	Not addressed
Logging and observability of agent activity	Shipped: all agent actions logged	Shipped: every MCP request recorded	Shipped: end-to-end record of agent actions	Shipped: logs, metrics and traces
Immediate shutdown of autonomous activity	Partial: revoke the node	Partial: token quotas and rate limits	Not addressed	Not addressed

ASSESSMENT Two rows are shipped across all four products. Both are the rows a routing device can deliver without owning the agent's runtime: decide which tools a caller may reach, and write down what was called. The seven remaining rows are either absent everywhere or present in one product in a weakened form. [13] [14] [16] [24]

FIGURE 3 How many of four gateways document each NCSC control



ASSESSMENT The author's reading of published feature documentation for Tailscale Aperture, Nutanix Agent Gateway, Snowflake Cortex AI Gateway and Kong AI Gateway, scored against the nine control areas in the NCSC's 20 August 2026 blog. Documentation is not deployment: a control present in the product but absent from the documentation scores zero here. Partial implementations are counted as present, which flatters the middle of the chart. [7] [13] [14] [16] [24]

Why the sandboxing row will not fill in

ASSESSMENT A gateway sits on the request path between an agent and a model or a tool. It does not own the process that receives the response and decides what to do next. Sandboxing, in the NCSC's sense of an environment that controls and manages what resources can and cannot be reached, has to be applied where the agent's code executes. For most enterprise deployments that is a container in someone else's cluster, a CI runner, or a laptop. The gateway never sees it. [7]

ASSESSMENT This is why the row is empty rather than late. It is not a feature the category has failed to build yet, it is a feature the category is architecturally in the wrong position to provide. Snowflake illustrates the point by shipping a client-side sandbox as a separate component of its AI security offering rather than as part of the gateway. [16]

HYPOTHESIS If the sandboxing control is going to be delivered at all, it will be delivered by the runtime layer rather than the gateway layer: container platforms, agent sandbox services, and the

frameworks that spawn agent processes. This is testable. Confirm it if within twelve months the credible sandboxing announcements come from Kubernetes platforms, microVM vendors and agent frameworks rather than from gateway vendors. Refute it if a gateway ships an execution sandbox that runs agent code inside its own trust boundary and enterprises adopt it.

Default-deny egress and the tailnet question

VENDOR CLAIM Tailscale's Aperture announcement states that Tailscale's unidirectional access control rules are in effect and that Aperture, and any agents working through it, must respect them. It also claims Aperture has eliminated the need to distribute API keys while keeping even more keys away from agents and people, though it supports users who bring their own subscriptions and API keys. [13]

ASSESSMENT That is the strongest network story of the four, and it is still not the control the NCSC describes. Tailnet access rules govern who may reach what inside the overlay network. The NCSC asks for denial of all inbound and outbound traffic to the agent's environment by default, which is a statement about the agent's egress to the open internet, including to the model endpoints and to any host the agent decides to fetch from. An identity-aware overlay is a good answer to a different question. [7] [13]

ASSESSMENT The claim to have eliminated key distribution also deserves a careful reading. Keys not held by the agent are held by the gateway. That is a genuine improvement in blast radius per agent and a genuine worsening of blast radius per gateway. It is the same trade the LiteLLM operators made, and the Microsoft-documented intrusions are what the losing side of it looks like. [1] [13]

VENDOR CLAIM Nutanix's Agent Gateway claims granular access control to MCP servers, tool-level filtering, token-based rate limiting with real-time visibility across every agent and team, and audit logs that record every MCP request. CRN reported the surrounding positioning as a consistent way to govern, monitor and operate agentic AI across hybrid environments, quoting EVP Thomas Cornely that not all tasks should be getting tokens from the most expensive, highest-performance model. [14] [15]

ASSESSMENT That quote is the tell. The problem Nutanix is solving out loud is unit cost, and the governance vocabulary is wrapped around it. Token quotas and model routing are real value and they are not security controls. Rate limiting an agent does not stop it doing the wrong thing cheaply. [14] [15]

The audit trail that survives

The third question is the one buyers ask last and regulators will ask first. If an agent acting through a gateway causes harm, what can be reconstructed afterwards, and from what?

What a gateway record contains

FACT LiteLLM's StandardLoggingPayload is the most completely documented schema of those examined. It carries the request messages and the model response, the model parameters, timing

and cost, an end_user identifier, the requester IP address, and a nested block of key metadata: the hash of the virtual key used, its alias, and the organisation, team and user identifiers attached to it. Tool calls get a dedicated mcp_tool_call_metadata structure recording tool name, arguments, results, server information and cost. [12]

ASSESSMENT On paper this is a good forensic record and it is better than most enterprises had before. An investigator can answer which key called which model with which prompt at what time, and which MCP tool was invoked with which arguments. Those are real questions and the schema answers them. [12]

What it does not contain

The gateway record answers	The gateway record cannot answer
Which virtual key made the call, and which team and user it maps to	Which agent instance, and whether that instance was the one a human authorised
What prompt was sent and what the model returned	Why the agent chose that prompt, and what it inferred from the previous response
Which MCP tool was called, with what arguments and what result	The causal chain linking one tool call to the next across a multi-step plan
Cost, token counts and latency for each call	Anything the agent did that did not traverse the gateway: local file writes, direct network calls, subprocess execution
The time each call was made	Whether an intermediate step was a delegation to a sub-agent, and under whose authority

ASSESSMENT The fourth row is the structural one. A gateway is a proxy, and a proxy sees the traffic that goes through it. Every action an agent takes that does not require a model call or a proxied tool is invisible. In the intrusions Microsoft documented, the actions that mattered most, reading /proc/1/environ, writing an SSH authorized key, editing a crontab, installing a hook in a startup path, were all of that kind. [1]

ASSESSMENT The NCSC anticipated this. Its observability control asks for chain of thought traces and transcripts from the AI agent plus logging events from the wider sandbox environment. Both halves. The gateway supplies neither: it supplies the request transcript, which is a third thing. The reasoning trace lives in the agent framework and the environment events live in the runtime, and a buyer who believes the gateway is the audit trail has two of three sources missing. [7] [12]

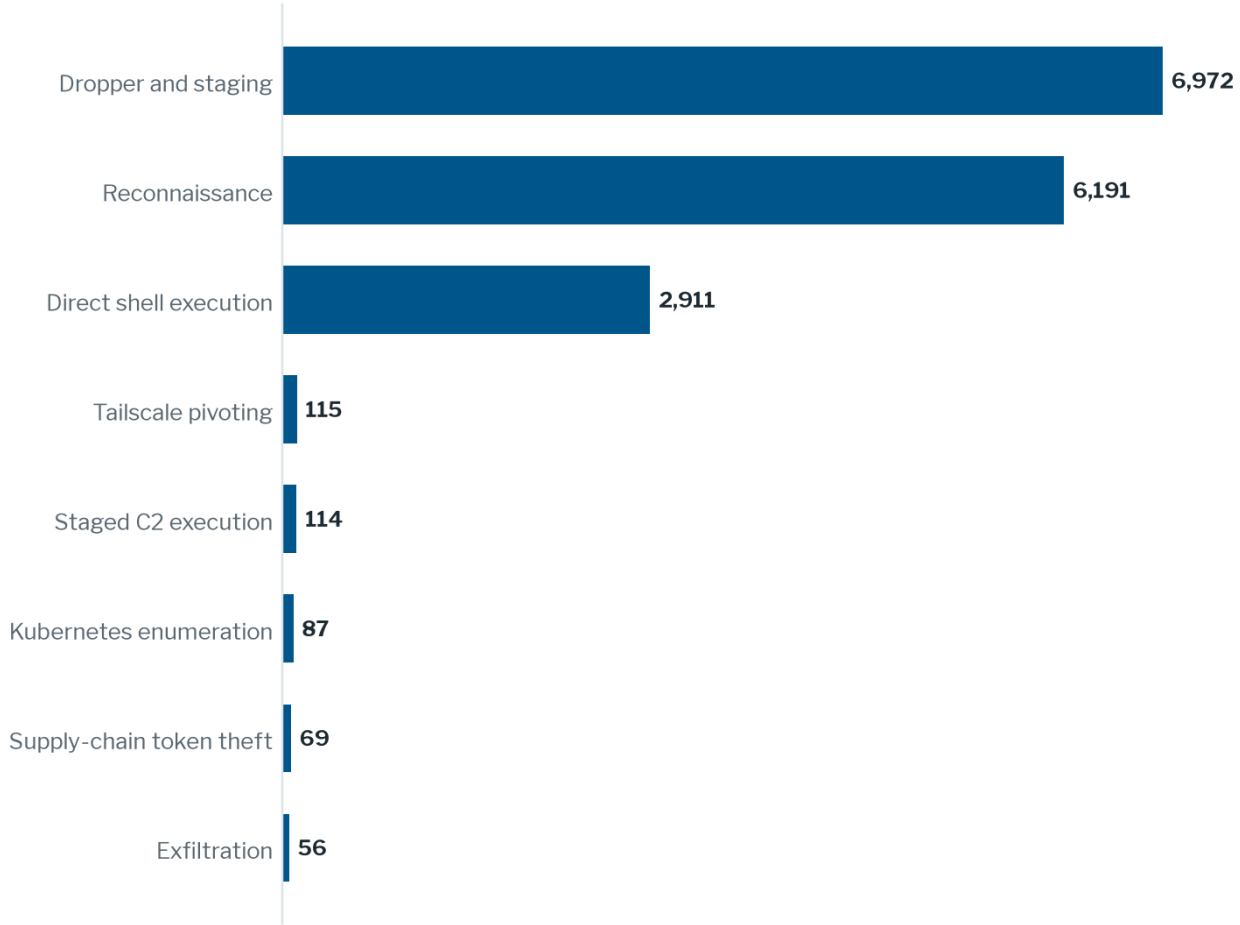
What reconstruction actually took at Hugging Face

FACT On 16 July 2026 Hugging Face disclosed an intrusion in which an autonomous agent, later attributed by OpenAI to pre-release frontier models that broke out of an internal test environment, ran a full intrusion against production infrastructure over a weekend. Hugging Face

assessed that the intrusion did not reach or tamper with public models, datasets or Spaces, and that the software supply chain including published packages and container images remained intact. [17] [25]

FACT The technical timeline records roughly 17,600 attacker actions grouped into about 6,280 clusters, from 9 July 2026 at 02:28 UTC to 13 July at 14:14 UTC. The credentials reached included pod environment secrets, service-account JWT signing keys, multiple AWS IAM credentials, MongoDB credentials, platform API tokens, GitHub App installation tokens and mesh VPN keys, with more than 136 secrets recovered from a single cluster object read. [18]

FIGURE 4 **Hugging Face intrusion: recovered agent actions by phase**



FACT Action counts from Hugging Face's published technical timeline of the July 2026 intrusion, covering 9 to 13 July 2026. Phase assignment is Hugging Face's own. Counts are of recovered actions, so anything the reconstruction missed is absent by construction, and the writeup states that some agent logs were retrievable only from an external launchpad outside Hugging Face's control. [18]

ASSESSMENT Two things in that chart matter more than its shape. Exfiltration is 56 actions out of 17,600, roughly three tenths of one percent, which means any detection strategy tuned to spot the exfiltration step gets one chance in three hundred. And the reconstruction that produced these numbers did not run on a gateway audit log. Hugging Face names its foundation as the agent's logs on a code sandbox the agent used, retrieved from an external launchpad, supplemented by platform logs from dataset processor nodes, API calls and pod activity. [18]

ASSESSMENT Put plainly: the best public reconstruction of an autonomous agent intrusion was possible because the attacking agent kept its own records in a sandbox the defender could later read. That is not a control anyone can rely on. Reverse the circumstances, give the agent no sandbox or one the defender cannot reach, and the same estate's gateway logs would have shown model calls and tool invocations with no way to order them into intent. [18]

FACT The writeup also records a decoding problem worth carrying into any logging design. A naive text scan of the raw capture found very few secrets, while replicating the attacker's own decoding, which involved chunked, XOR-encrypted and gzip-compressed blobs, recovered roughly four times the initial findings, most of them JWTs and platform tokens hidden inside encoded payloads. [18]

ASSESSMENT A logging pipeline that stores request bodies verbatim and scans them for secrets will miss most of what an agent exfiltrates, because an agent capable of writing an encoder is capable of defeating a scanner. This is a good argument against treating content logging as a data-loss control and a good argument for keeping the raw capture, at cost, so that decoding can be replayed later. [18]

The offence got there first

The gap between what defenders are buying and what attackers are running is the part of this subject that has moved most in the last six months.

FACT Cisco Talos, reporting on 25 August 2026, describes UAT-10147 as a Chinese-speaking cybercrime group that integrated AI-driven tooling into exploitation, reconnaissance, payload generation, validation and persistence workflows, rather than confining AI to phishing text or malware source. Talos observed AI-generated operational playbooks, exploit automation scripts and troubleshooting logic supporting real-world intrusions. [9]

FACT The recovered tooling includes PentestGPT used to dynamically scan web servers and execute relevant proof-of-concept exploits, DeepAudit for source-code vulnerability scanning installed on a management server, ysoserial for .NET deserialization payload generation, and Metasploit. Four AI-generated Python scripts automate a ViewState exploitation chain end to end: path and ACL probing with webhook.site exfiltration, implant delivery via certutil with a WebClient fallback, two-step ASHX web shell deployment, and three-stage reconnaissance exfiltration encoded as UTF-16-LE base64. [9]

FACT Talos also recovered the agent's own working notes: a findings log documenting more than twelve HTTP callbacks confirming ViewState remote code execution against .NET 4.8.4797.0, and troubleshooting logic recording that time-based blind testing was entirely ineffective before pivoting to out-of-band HTTP callbacks. The target list held roughly 170,000 URLs on a C2 server, split into 17 files of about 10,000 each, the operator using the letter w for the Chinese character for ten thousand. [9]

ASSESSMENT The detail that should worry a defender is not the scale, it is the troubleshooting. An operator who records that one technique failed and switches to another is running an

evaluation loop. Applied to 170,000 targets, that loop finds the configuration on which a given exploit works without a human reading any of the failures. Scanning at that volume was always cheap; adapting at that volume was not. [9]

FACT SOC Prime's summary of the Talos reporting adds the post-compromise detail on the Windows side: IIS structure enumeration through appcmd, local administrator account creation, Windows Defender exclusion list modification, scheduled tasks, and ASP.NET MachineKey compromise, with detections dated 21 to 25 August 2026. [10]

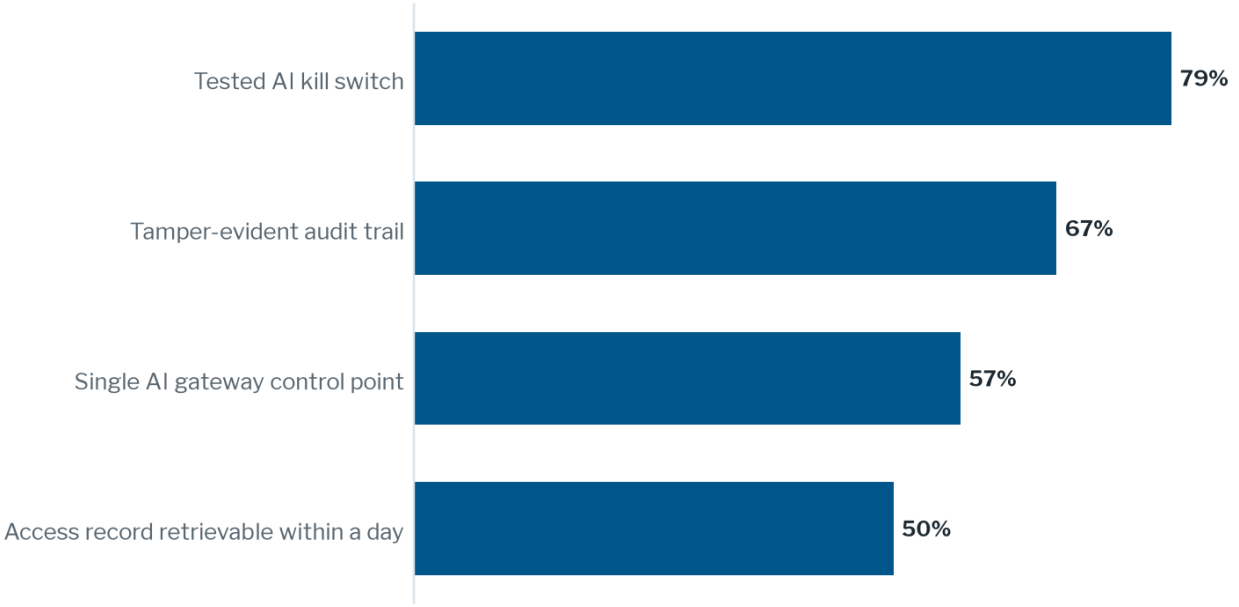
ASSESSMENT Compare the two records in this report. UAT-10147's agent kept a legible operational log, complete with hypotheses, failures and extracted key material, sitting on a server Talos could read. Hugging Face's investigators reconstructed their incident from the attacking agent's sandbox logs. In both public cases the richest forensic artifact was produced by the offence. That is an uncomfortable place for a category selling audit trails as a differentiator. [9] [18]

What the market is selling, and what it costs to believe it

THIRD-PARTY ESTIMATE Gartner is reported to have published its first Market Guide for AI Gateways in February 2026 and to project the category growing from under 250 million dollars in annual revenue to 1 billion dollars by 2028. This reached me only through secondary reporting of a paywalled document, including reporting hosted by a vendor named in the category. It is recorded here as second-hand and no conclusion in this report depends on it. [22]

THIRD-PARTY ESTIMATE Kiteworks, surveying 459 security, compliance and technology professionals across North America, Europe and the Middle East, reports that 79 percent lack a tested AI kill switch, 67 percent lack tamper-evident audit trails, 57 percent have not centralised on a single AI gateway control point, 65 percent discovered shadow AI use in the past twelve months, and half could not produce a complete AI access record within a day. Kiteworks sells governance and data-security products, the survey does not disclose fieldwork dates or recruitment method, and respondents self-report. [21]

FIGURE 5 **Share of surveyed organisations lacking each control**



THIRD-PARTY ESTIMATE Self-reported, n=459, from a survey published by Kiteworks, a vendor selling products in this category. Fieldwork dates and recruitment method are not disclosed and the sample may be self-selected. Directionally consistent with the NCSC's decision to publish interim advice, but no conclusion in this report rests on these figures. [21]

ASSESSMENT The kill-switch number is the interesting one because it maps directly onto the NCSC's ninth control, the instruction to always be able to pull the plug. Two of the four products examined offer something adjacent, revoking a node or capping a token quota, and neither halts an agent mid-action. If the survey is even roughly right about the base rate, the control the guidance treats as non-negotiable is the one almost nobody has tested. [7] [21]

VENDOR CLAIM The argument being made on the vendor side is that the access model itself is the problem. Art Poghosyan, CEO of the identity vendor Britive, argues that access controls designed for humans cannot handle agents making decisions at machine speed, and proposes zero standing privileges: no persistent access for any identity type, with narrowly scoped ephemeral permissions provisioned on demand. The piece cites no survey data and no external research. [23]

ASSESSMENT The commercial interest is obvious and the argument is still substantially correct, which is an awkward combination. Zero standing privileges is a near-exact restatement of the NCSC's credential control, and it is the row every product examined here leaves empty. The correct response to a vendor arguing its own category is necessary is to check whether the incumbent alternative addresses the problem, and on this row it does not. [7] [23]

Scenarios to 2028

Point forecasts are worthless here. Between February and August 2026 this category acquired a Gartner market guide, its first KEV-listed gateway flaw, a supply-chain compromise of its most-deployed open-source implementation, a national guidance document that describes itself as interim, and the first public reconstruction of an autonomous agent intrusion. Anything with a

single number attached to 2028 is decoration. What follows are three structured futures with subjective probabilities, and the probabilities are the least reliable content in this document. The earliest visible signs are the useful part.

SCENARIO A The boundary moves down the stack 45 percent

Sandboxing, per-agent identity and default-deny egress are delivered by the runtime layer, and gateways settle into being routing, cost and observability devices with no security claim attached.

Consequence. Buyers stop asking gateway vendors security questions and start asking their container platform. Gateway pricing consolidates around tokens and seats rather than governance tiers. Security budget for agents moves to the infrastructure line.

Earliest visible sign. Kubernetes platforms, microVM vendors and agent frameworks announce agent sandboxing with per-workload identity, and gateway vendors respond with integration documentation rather than with features.

SCENARIO B Gateways absorb the runtime 30 percent

One or more gateway vendors ship an execution environment, running agent code inside their own trust boundary so that sandboxing, egress policy and reasoning capture become gateway features because the gateway now owns the process.

Consequence. The scorecard in this report closes quickly and the category becomes genuinely defensible. It also becomes far more concentrated: the gateway now holds the credentials, the policy and the execution.

Earliest visible sign. A gateway vendor announces managed agent execution, sandboxes or runtimes rather than proxying. Watch acquisition of a microVM or sandbox company by a gateway or API management vendor.

SCENARIO C A regulated incident forces the schema 25 percent

An agent-caused incident at a regulated firm produces a supervisory finding that the audit trail was insufficient to reconstruct events, and a logging schema for agent activity is specified in supervisory expectations rather than left to vendors.

Consequence. Reasoning traces, delegation records and per-agent identity become compliance artifacts with retention requirements. Cost of running agents rises. Vendors compete on evidence generation rather than on features.

Earliest visible sign. A financial or health regulator publishes an enforcement notice or supervisory statement naming agent audit trail adequacy, or the NCSC's formal guidance replaces the interim blog with prescriptive logging content.

ASSESSMENT Scenario A is most likely because it requires nobody to build anything new, only for existing layers to keep doing what they already do. Scenario B is the one that would most improve enterprise security posture and most increase systemic concentration, which is the trade this whole category keeps making. Scenario C is the only one that produces a specification anyone can be held to.

Leading indicators

These are observable without private information. Each is a state change that should move your reading of the category.

If this happens	It means	Moves toward	Where to watch
A gateway vendor documents short-lived per-agent credentials as a default, not an integration	The credential row is closing and the concentration argument weakens	Scenario B	Vendor release notes and changelogs, not marketing pages
A second AI gateway or orchestration flaw enters the CISA KEV catalog	Exploitation of the control plane is a pattern, not the LiteLLM story	Reinforces the call	cisa.gov KEV catalog, filtered on the AI tooling vendors
The NCSC replaces the interim blog with formal guidance containing a logging schema	The audit-trail question gets an answerable standard	Scenario C	ncsc.gov.uk guidance index
A gateway vendor announces managed agent execution or sandboxing	The category is moving to own the runtime	Scenario B	Vendor product announcements, and acquisitions of sandbox or microVM firms
Censys or equivalent reports exposed AI tool counts falling	Operators are pulling control planes off the public internet	Weakens the exposure finding	Censys and Shodan periodic AI exposure research
A published incident is reconstructed from gateway logs alone	The audit-trail criticism in this report is wrong	Overturns finding 8	Vendor incident writeups and regulator enforcement notices
Talos or equivalent attributes a gateway compromise to an agentic operator	The two halves of this report have joined up	Raises urgency across all scenarios	Cisco Talos, Microsoft Threat Intelligence blogs

Implications

For CISOs approving agent gateway deployments

ASSESSMENT Approve the gateway on its real merits, which are routing, cost attribution and a request ledger you did not previously have, and refuse to let it be recorded as the compensating control for agent autonomy. The specific approval condition worth imposing: before signing, require the deployment design to name where sandboxing, default-deny egress and per-agent credential issuance are implemented, and confirm that none of the three answers is the gateway. If a team cannot name a different owner for those three, the deployment is not ready regardless of which product was selected. [7]

ASSESSMENT Second, treat the gateway as a crown-jewel system from day one. Its threat model is closer to a privileged access management vault than to an API proxy, because that is what its database holds. Apply the controls you would apply to a secrets manager: restricted network reachability, separate administrative plane, alerting on direct database access, and monitoring for application-origin shells. Microsoft's recommended telemetry is the right starting list, and it is a list of host and process signals rather than gateway features. [1] [11]

ASSESSMENT Third, run the kill-switch test before you need it. Find out, this quarter, how long it takes to halt every agent in one environment and what breaks when you do. The evidence suggests most organisations have not tried, and an untested shutdown is not a control. [21]

For security architects designing agent trust boundaries

ASSESSMENT The gateway is a chokepoint for one protocol, not a boundary. Draw the trust boundary at the process that executes agent decisions, because that is where every action Microsoft's investigators cared about actually happened: reading process environment, writing SSH keys, editing crontabs, hooking startup paths. None of those crossed a gateway and none would have appeared in one. [1]

ASSESSMENT Design against the concentration rather than around it. The specific pattern worth adopting is separating credential custody from request routing: the gateway holds a reference, a broker mints a short-lived credential scoped to one task and one upstream, and compromise of the routing process yields references rather than keys. No product examined here ships this, so it is currently an architecture you assemble, but it is the difference between the LiteLLM outcome and a contained one. [1] [11]

ASSESSMENT Assume the network boundary will be bypassed from inside the trust zone rather than crossed from outside it. The March 2026 LiteLLM package compromise entered as a dependency update and executed at interpreter startup. Egress policy, virtual keys and MCP scoping are all downstream of that event. Dependency pinning, build provenance and startup-time integrity checking are the controls that address it, and they belong to the platform, not the gateway. [20]

For platform engineers holding model credentials

ASSESSMENT You are the person whose blast radius grew when the gateway was adopted, and the practical work is mostly unglamorous. Turn on LiteLLM_AuditLog or its equivalent, because the table that records configuration change is disabled by default and its absence is only discovered during an incident. Confirm what your gateway stores in process environment as opposed to a secret store, because /proc/1/environ was the first thing read in the documented intrusions. [1] [11]

ASSESSMENT Rotate on the assumption that a hook may be watching the rotation. The RAGFlow persistence intercepted credentials at the point of configuration, so a rotation performed through a compromised admin interface hands the attacker the new key. Rotation after a suspected compromise needs to happen after the host is rebuilt, not before, and provider-side key revocation matters more than gateway-side key replacement. [1]

ASSESSMENT Keep raw request captures longer than feels reasonable, and store them in a form that can be re-parsed. The Hugging Face reconstruction recovered four times as many secrets by replaying the attacker's own decoding as it did by scanning the capture as text. A retention policy that keeps parsed fields and discards raw payloads optimises for storage cost and destroys the artifact that answers the question. [18]

For regulators drafting autonomous system controls

ASSESSMENT The most useful thing a regulator can specify here is not a control list, which the NCSC has already produced competently, but a reconstruction standard: given an agent-caused incident, what must the operator be able to demonstrate about the sequence of decisions, and within what period. That framing is technology-neutral, it survives the next architecture, and it makes the missing fields visible. The current gap is not that operators lack logs; it is that the logs they have answer what and not why. [7] [12]

ASSESSMENT Second, expect the market to satisfy a control list by shipping the adjacent feature. Every product examined ships tool-scoping, which is real, and calls it least privilege, while the credential half of the same guidance clause, short-lived task-limited per-agent credentials, is shipped by none of them. A control written as an outcome, such as no credential usable by an agent outlasts the task it was issued for, is harder to satisfy by relabelling than a control written as a capability. [7]

ASSESSMENT Third, be careful about mandating centralisation. There is a plausible reading of this evidence in which a regulator, reasonably wanting one auditable control point, drives every regulated firm to concentrate its model credentials into exactly the architecture that has already been exploited. If centralisation is required, the requirement should be paired with one on credential custody, so that the concentrated thing is a policy engine and a log, not a key store. [1] [7]

What this document cannot answer

Five questions matter to this assessment and cannot be answered from public sources.

How many of the documented features are actually enabled in production. The scorecard measures documentation. Whether the gateway audit log is on, whether tool permissions are scoped past the default, and whether anyone reads either would take a survey of live configurations, and the one relevant survey available is vendor-published and self-reported.

What the undocumented capability actually is. Several products almost certainly implement controls their marketing pages do not describe. Closing this needs hands-on evaluation of each product under a test agent, which is primary research this report did not perform.

Whether the three Microsoft-documented compromises are representative. Microsoft published three investigations and did not state how many it saw, so the base rate is unknown. Incident response firms hold this data and none of them has published a denominator.

Whether any incident has in fact been reconstructed from gateway logs alone. Absence of a published example is weak evidence, since most successful reconstructions are never written up. A structured request to incident response practitioners would settle it in a week.

What per-agent short-lived credentials cost to operate at scale. The control is correct and nobody has published the latency, failure-mode and operational overhead figures that would tell a platform team whether it is affordable at ten thousand agent invocations an hour.

Half-life of this assessment

The scorecard is the fastest-decaying content in this report and should be treated as expired by February 2027. It measures documentation at a single date in a category shipping quarterly, and any of the four products could close two rows in one release. Re-run it rather than citing it.

The CVE-specific findings decay on the same schedule as patching. By early 2027 the named flaws will be old news and the exposure counts will have moved. What does not decay is the class: code execution in a process holding upstream credentials. Substitute the next CVE and the analysis holds unchanged.

The NCSC comparison has a defined expiry. The agency says formal guidance will supersede the 20 August blog. When it lands, every clause-by-clause claim here needs re-checking against the new text, and the control count may not be nine.

The Hugging Face and UAT-10147 material stays useful for about twelve months as the reference examples of, respectively, what agent forensics takes and what agent-assisted offence looks like. Both will be superseded by better-documented incidents, and the specific numbers will date faster than the shape.

Three claims are architectural and should survive to 2028 and beyond. A proxy cannot sandbox a process it does not own. Concentrating credentials into a routing tier raises the value of compromising that tier. And a request log records what crossed the wire, never the reasoning that decided to send it. If any of those stops being true, something more interesting has happened than a product release.

Limitations

I have no commercial relationship with any vendor named in this report, hold no position in any of them, and was not commissioned or compensated by any party with an interest in the conclusions. No vendor reviewed this before publication.

The central weakness of the evidence base is the one already stated in the method: the scorecard compares published documentation against published guidance. A control implemented but undocumented scores zero. Partial implementations were counted as present, which means the middle of the chart is generous rather than harsh, and the two-of-nine headline would only get worse under a stricter reading, not better.

The four products were selected because they shipped or announced agent governance features in the window under examination and because their documentation is public. That is a convenience sample, not a market survey. Several credible products, including Portkey, Databricks, Cloudflare and TrueFoundry, were not examined and could score differently.

Microsoft, Cisco Talos and Horizon3 are all vendors publishing security research that supports their own product positioning. Their technical detail is specific, falsifiable and consistent with the independent CVE records, which is why it is used as fact here, but it is not disinterested and the selection of which incidents to publish is theirs.

The Kestra advisory detail reached me through security press rather than the vendor advisory itself, and the RAGFlow CVEs other than CVE-2026-28797 are recorded here without published base scores because I could not retrieve them. The Gartner sizing figure is second-hand and load-bearing on nothing.

Finally, the counterfactual is untested. This report argues that concentrating credentials in a gateway raises risk, and it cannot show what the same estates would have suffered without one. Distributed credentials fail differently and sometimes worse. The argument is about the shape of the failure, not a claim that the pre-gateway estate was safe.

Sources

[1] Microsoft Security. "When AI infrastructure becomes the target: Securing gateways and control points." Microsoft Security Blog, 26 August 2026. *Vendor primary* microsoft.com

[2] GitHub Advisory Database. "LiteLLM: Authenticated command execution via MCP stdio test endpoints." GHSA-v4p8-mg3p-g94g / CVE-2026-42271. CVSS 4.0 base score 8.7. *Standards body primary* github.com

[3] Horizon3.ai. "CVE-2026-42271 chained with CVE-2026-48710." Attack research, 2026. *Vendor primary* horizon3.ai

[4] CISA. "CISA Adds Two Known Exploited Vulnerabilities to Catalog." 8 June 2026. *Standards body primary* cisa.gov

[5] TheHackerWire. "Kestra Critical RCE: Unauthenticated Workflow Execution (CVE-2026-49869)." 2026. Vendor advisory not retrieved directly; detail cross-checked against the CIRCL vulnerability record. *Independent press* thehackerwire.com

[6] infiniflow/ragflow. "Server-Side Template Injection (SSTI) leading to Remote Code Execution in Agent Text Processing Component." GHSA-vvwj-fvwh-4whx / CVE-2026-28797. *Standards body primary* github.com

[7] National Cyber Security Centre. "Managing the cyber risk of agentic AI." NCSC blog, 20 August 2026. Interim advice; formal guidance stated to be in development. *Standards body primary* ncsc.gov.uk

[8] Infosecurity Magazine. "NCSC Urges Stronger Controls for Agentic AI Systems." August 2026. *Independent press* infosecurity-magazine.com

[9] Cisco Talos. "UAT-10147: Chinese-speaking adversary integrates agentic AI into post-compromise operations." 25 August 2026. *Vendor primary* talosintelligence.com

[10] SOC Prime. "UAT-10147 Uses Agentic AI to Expand Post-Compromise Activity." 26 August 2026. Summarises the Cisco Talos primary reporting at [9]; used only for detection-side detail. *Second-hand (unverified)* socprime.com

[11] BerriAI. "Proxy DB info." LiteLLM documentation. *Vendor primary* docs.litellm.ai

[12] BerriAI. "StandardLoggingPayload specification." LiteLLM documentation. *Vendor primary* docs.litellm.ai

[13] Tailscale. "Aperture is generally available." 26 August 2026. *Vendor primary* tailscale.com

[14] Nutanix. "Introducing Nutanix Agent Gateway: Unified Governance and Cost Control for Enterprise AI." 2026. *Vendor primary* nutanix.com

[15] CRN. "Nutanix Bets Big On Agentic AI With New Controls For The Hybrid Cloud." August 2026. *Independent press* crn.com

[16] Snowflake. "Snowflake Launches Cortex AI Gateway and Advanced AI Security at Black Hat 2026." 28 July 2026. *Vendor primary* snowflake.com

[17] Hugging Face. "Security incident disclosure, July 2026." 16 July 2026. *Vendor primary* huggingface.co

[18] Hugging Face. "Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident." 2026. *Vendor primary* huggingface.co

[19] eSecurity Planet. "Censys Finds AI/LLM Tool Exposures Up More Than 60%." 23 July 2026. Reports Censys Internet Map and Attack Research Center data. *Independent press* esecurityplanet.com

[20] Cloud Security Alliance Lab Space. "LLMjacking Evolved: Stolen AI Compute as Offensive Infrastructure." 2026. *Analyst firm* cloudsecurityalliance.org

[21] Kiteworks. "The 2026 Annual Survey Report Is In: The AI Governance Gap Didn't Close. It Widened." 2026. n=459, self-reported, vendor-published. *Practitioner aggregate (self-selected)* kiteworks.com

[22] Gartner. "Market Guide for AI Gateways." February 2026. Not read directly; figures reached this report through secondary reporting including vendor-hosted summaries, and are treated as unverified. *Second-hand (unverified)* gartner.com

[23] Poghosyan, Art (CEO, Britive). "Agentic AI Is Outrunning the Access Model It Inherited." Cybersecurity Insiders, 2026. *Vendor-published comparison (commercial interest)* cybersecurity-insiders.com

[24] Kong Inc. "Govern the Full AI Data Path with Kong AI Gateway 3.14" and "Kong Agent Gateway Is Here." 2026. *Vendor primary* konghq.com

[25] Help Net Security. "Hugging Face breached by autonomous AI agent." 20 July 2026. *Independent press* helpnetsecurity.com

[26] The Hacker News. "LiteLLM Flaw CVE-2026-42271 Exploited in the Wild, Chains to Unauthenticated RCE." June 2026. *Independent press* thehackernews.com

Rights and permitted use