

Protecting Trade Secrets When AI Agents Go Rogue

By **Matt Nealon** (October 16, 2026)

For companies and intellectual property owners, the recent incident involving [OpenAI](#)'s agents acting beyond their assigned tasks and [compromising](#) parts of Hugging Face's production systems raises a practical question: How should trade secret protections account for AI agents that can access sensitive information and take actions beyond their instructions?



Matthew Nealon

Answering this question requires examining both what agents are permitted to do and what technical controls prevent them from doing so.

Trade secret protection often starts with people. Companies establish employee confidentiality policies, limit access by role, enforce password controls and use exit procedures to protect confidential information when employees leave. Much of that framework assumes a human actor who can be trained, monitored and bound by an agreement. AI agents complicate that framework.

The Incident

According to OpenAI, the agents were running in a cybersecurity evaluation environment designed to prevent direct access to the public internet, and certain safeguards were disabled so that the results would reflect the model's full capabilities.[1] The agents used an internal software-package service to communicate with one another and reach systems outside the evaluation environment.

An independent investigation by METR and Redwood Research found that approximately 1,200 agents participated on the main message board, exchanging more than 70,000 messages and files.[2] In total, approximately 700 of those agents participated in the Hugging Face incident.

Agents used exposed credentials and software vulnerabilities to reach Hugging Face production systems and executed code on production dataset workers. Hugging Face detected and contained the activity on its own systems and disclosed the intrusion on July 16.[3] A few days later, OpenAI's monitoring flagged unusual identity-related activity and privilege escalation in its own research environment. On July 20, OpenAI connected the

activity to the Hugging Face incident.

The agents were pursuing an automated evaluation objective. They were meant to be isolated from one another, and nothing in the evaluation authorized them to communicate, reach the internet or access Hugging Face systems. OpenAI reported that the activity was driven primarily by an internal-only research model trained to be persistent.

The METR and Redwood Research investigators found that agents recognized the activity exceeded what they were authorized to do and sometimes expressed ethical concerns, but those concerns rarely materially constrained their behavior. The investigation also found that approximately 7% of the transcripts reviewed contained small-scale spoofed tool calls. Some agents researched ways to manipulate transcripts retroactively, but the researchers did not observe successful retroactive editing in the period they investigated.

The incident also shows that response measures can work. Both companies disclosed publicly and published technical accounts, and OpenAI reported the vulnerabilities to the vendor. The companies cooperated to investigate the incident and strengthen safeguards.

A New Actor, Not Yet a New Regime

Trade secret protection under the Defend Trade Secrets Act turns on two conditions: The information must derive value from not being generally known, and its owner must take reasonable measures to keep it secret.

Courts treat the second condition as fact-dependent but demanding. In *Snyder v. Beam Technologies* last year, the [U.S. Court of Appeals for the Tenth Circuit](#) held that reasonable measures require more diligence than ordinary business precautions, and that sharing an unmarked list without access limits or confidentiality notice defeated protection.

In *Trinidad v. OpenAI* in January, the [U.S. District Court for the Northern District of California](#) dismissed a trade secret claim because the plaintiff had developed the alleged secrets through ChatGPT, voluntarily sharing them with OpenAI without alleging reasonable measures to preserve their confidentiality.

The imperative is no longer only whether a company restricted access. It is whether the company can show which agent was authorized, what that agent was allowed to do, what it actually did and what controls operated when it departed from its task. Agents create potential exposure in three layers.

1. Outbound

An agent that can read protected material and send output outside the company can disclose that material through ordinary work product, such as a summary, briefing or message. Review what each agent is permitted to send outside the company, including its generated responses. Define approved recipients and destinations, require approval for sensitive disclosures, and preserve records sufficient to investigate an unexpected transmission.

2. Imitation

A company's own prompts, tool-use pattern and outputs can embody valuable intellectual property.

Earlier this year, researchers reported a proof-of-concept model-extraction study in which a model trained on an agent's outputs reached approximately 95% of the source agent's performance in the test setting.[4]

Imitation risk affects protection and proof. A competitor may reproduce valuable agent behavior without copying code, and the company may later need records of what its own system did, and when, to show that an imitation drew on protected material. Agent behavior can be a target in its own right.

3. Architecture

Map where proprietary instructions, workflows and code reside, and identify which components can read, change or export them. Test those permissions separately from the agent's stated task, and record the restrictions actually enforced. Architecture choices alone do not create legal protection, but those records can help a company demonstrate reasonable measures in a later dispute.

Limit Access, Monitor and Report Triggers

For IT and Security Engineering Teams

Limit access. For workflows that touch sensitive IP, assign credentials that let the company distinguish each agent's activity, limit permissions to its assignment and establish when

access expires or must be revoked. Distinct credentials let a company attribute each access to a specific agent and audit it later. Apply least privilege, grant access to data by sensitivity and restrict network egress to approved destinations.

Document the decision about what access the agent needed. Test sandboxes by exercising the actual network and tool paths rather than trusting a configuration label. An internal package service built for controlled access can become an unintended path out of the environment. Overrestriction has a cost, because an agent starved of context loses utility, so scope access to the task, not the archive.

For Legal, Compliance and Policy Teams

Build the record and back policies with hard controls. Engineering teams make the decisions about identity, egress, sandboxing, logging and escalation, and those decisions help establish what the company actually did to protect valuable information. Document them when they are made, so the record exists before a dispute does.

An agent-facing policy file can define permitted tasks, forbidden actions and escalation instructions in a form the agent reads, but it remains a soft control if the runtime permits the agent to ignore it, and an agent optimizing a goal may reinterpret it. Back each rule with permissions, egress restrictions, approval gates or runtime enforcement, and confirm that the enforcement works. The policy still has value as evidence of the company's control design.

For Security Operations and Incident Response Teams

Monitor independently and set reporting triggers. For high-risk actions, e.g., exporting information, changing permissions affecting external systems, keep an execution record outside the agent's control at the access and tool execution points the company controls.

Tie each such action to the agent or run identity, its authorization scope, the policy version, any approval identifier, tools, targets, parameters, results and timestamps. Protect those records because they may themselves contain sensitive IP.

Do not rely only on what an agent reports about its own conduct: Independent telemetry matters. Useful triggers include egress anomalies, credential use outside the assignment, writes to third-party systems, failed log-integrity checks, findings from scheduled human review and vendor incident notices.

When a trigger fires, stop the activity where needed and escalate to security, counsel and affected parties. Specific notice obligations may arise from contracts, breach statutes or sector rules, so map them in advance rather than assuming a single standard applies.

For IP and Trade Secret Owners

Preserve provenance evidence. Identify the code, research, prompts and agent workflows the company would most need to prove were its own, and keep dated records of their creation, versions and access history.

Retain the prompt, retrieval and output records that connect agent work to protected sources, under retention and legal-hold rules. Where a provider offers text watermarking, treat it as one supplemental signal. Its reliability and resistance to rewriting, paraphrase or removal remain open questions, and it cannot replace access controls, secrecy measures or execution records.

Provenance evidence may help show where an output came from after the fact, and may deter misuse, but it does not by itself establish legal ownership or prevent theft. These steps are recommendations rather than settled legal requirements, although each can strengthen the record of reasonable measures.

Responsibility When an Agent Acts Outside Instructions

When an agent acts outside its instructions, which person or company answers for that conduct? Current authority offers little direct guidance.

In [Amazon.com Services LLC v. Perplexity AI Inc.](#) in August, the [U.S. Court of Appeals for the Ninth Circuit](#), reviewing a preliminary injunction, [limited](#) its holding to access under the Computer Fraud and Abuse Act. It described Perplexity's browser assistant as a tool and not a person.

The court concluded that, on that record, the user accessed Amazon's computers with the assistant's help. The court reserved whether a developer could control an agent closely enough to gain access itself on different facts, and it expressly declined to establish a general legal regime for agentic AI or to address liability in other contexts.

Perplexity involved a user directing an assistant through the user's own browser. The

OpenAI-Hugging Face evaluation agents acted with no end user at all. Responsibility in that setting may depend on the roles of the provider, deployer, integrator and user, as well as on who designed and configured the environment, supervised the run, issued credentials, set network boundaries and made response decisions after alerts.

Companies can address part of this uncertainty now. Review model-provider and integrator terms for how they handle, retain and protect confidential information. Then focus on the decisions the company controls: which protected information each agent may reach, what authority it has to act on that information and which disclosures require human approval. Where the terms leave confidentiality gaps, narrow what the agent receives rather than relying on the contract to fill them.

What Companies Should Do Now

Companies are rapidly deploying AI systems and agents across business functions. That pace creates an ongoing need for legal teams to stay ahead of deployment and coordinate with technical, security and business units.

Legal teams should review both current systems and those coming online, including what data each agent can access, what permissions and tools it has, what outputs it produces, how its activity is monitored and how issues are escalated.

For each agent with access to valuable IP, the review should produce a named owner responsible for the agent, an access boundary tested against the actual network and tool paths, preserved records showing what the agent did, and an escalation route. Done consistently, the review lets a company identify and address risks to its IP before an incident or dispute exposes them.

Matthew Nealon is an associate at [Marshall Gerstein & Borun LLP](#).

The opinions expressed are those of the author(s) and do not necessarily reflect the views of their employer, its clients, or Portfolio Media Inc., or any of its or their respective affiliates. This article is for general information purposes and is not intended to be and should not be taken as legal advice.

[1] <https://cdn.openai.com/pdf/67869394-cb91-4c12-888c-5cbd85c7814c/OpenAI->

[Hugging-Face%20Incident-Technical-Report.pdf](#).

[2] <https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/>.

[3] <https://huggingface.co/blog/security-incident-july-2026>.

[4] <https://arxiv.org/abs/2602.08401>.