← Back to Articles

Security incident disclosure — July 2026

Published July 16, 2026

system
system Follow

Earlier this week, we detected and responded to an intrusion into part of our production infrastructure. This one was different from anything we had handled before in one important way: it was driven, end to end, by an autonomous AI agent system - and we detected and dissected it largely with AI of our own.

We identified unauthorized access to a limited set of internal datasets and to several credentials used by our services. We are still completing our assessment of whether any partner or customer data was affected, and we will contact any affected parties directly as required. We have found no evidence of tampering with public, user-facing models, datasets, or Spaces, and our software supply chain (container images and published packages) was verified clean.

What happened

The intrusion started where AI platforms are uniquely exposed: the data-processing pipeline. A malicious dataset abused two code-execution paths in our dataset processing (a remote-code dataset loader and a template-injection in a dataset configuration) to run code on a processing worker. From there, the actor escalated to node-level access, harvested cloud and cluster credentials, and moved laterally into several internal clusters over a weekend.

The campaign was run by an autonomous agent framework (appearing to be built on an agentic security-research harness - used LLM still not known) executing many thousands of individual actions

across a swarm of short-lived sandboxes, with self-migrating command-and-control staged on public services. This matches the "agentic attacker" scenario the industry has been forecasting.

What we did

- Fixed the root vulnerability: the dataset code-execution paths used for initial access are closed.
- Eradicated the attacker's foothold across the affected clusters and rebuilt the compromised nodes.
- Revoked and rotated the affected credentials and tokens, and began a broader precautionary rotation of secrets.
- Deployed additional guardrails and stricter admission controls on our clusters.
- Improved our detection and alerting so a high-severity signal pages a responder in minutes, any day of the week.

We are working with outside cybersecurity forensic specialists to investigate the issue and review our security policies and procedures. Finally, we have also reported this incident to law enforcement agencies.

For our community

As a precaution, we recommend rotating any access tokens and reviewing recent activity on your account. If you believe you are affected, or want to report a security concern, contact us at security@huggingface.co.

We are grateful to the teams across Hugging Face who responded around the clock, and we are sorry for any disruption this caused. Security is never finished; we will keep raising the bar.

Analyzing an AI-driven intrusion

The attack was initially surfaced through AI-assisted detection. Our anomaly-detection pipeline uses LLM-based triage over security telemetry to separate real signals from the daily noise, and it was the correlation of those signals that flagged the compromise.

To understand what a swarm of tens of thousands of automated actions did, we ran LLM-driven analysis agents over the full attacker action log, comprised of more than 17,000 recorded events. This allowed us to reconstruct the timeline, extract indicators of compromise, map the credentials touched, and separate genuine impact from decoy activity. Thanks to this approach, we were able to do in hours what would usually take days, and match the adversary's speed.

The choice of models we could use for this analysis was constrained in a way we did not anticipate; we describe this below.

The asymmetry problem

When we started the log analysis, we first used frontier models behind commercial APIs. This did not work: the analysis requires submitting large volumes of real attack commands, exploit payloads, and C2 artifacts, and these requests were blocked by the providers' safety guardrails, which cannot distinguish an incident responder from an attacker. We ran the forensic analysis instead on zai-org/GLM-5.2, an open-weight model, on our own infrastructure. This had a second benefit: no attacker data, and none of the credentials it referenced, left our environment.

This experience points to a gap worth planning for. We do not know which model powered the attacker's agents, whether a jailbroken hosted model or an unrestricted open-weight one; either way, the attacker was bound by no usage policy, while our own forensic work was blocked by the guardrails of the hosted models we first tried. The practical lesson for defenders: have a capable model you can run on your own infrastructure vetted and ready *before* an incident, both to avoid guardrail lockout and to keep attacker data and credentials from leaving your environment. This is not an argument against safety measures on hosted models, and we are sharing this feedback with the providers concerned.

What this means

Autonomous, AI-driven offensive tooling is no longer theoretical. It lowers the cost of running a broad, patient, multi-stage campaign, and it operates at machine speed. Defending an online platform now means treating the data and model surface as a first-class attack surface, and using AI on defense to keep pace. We will keep investing there, and keep sharing what we learn.