DRAFT

Hugging Face Incident Initial Post-Mortem

Expedited Strategy Briefing

By the CSA CISO Community, SANS, [un]prompted, RSAC, Knostic, FIRST, and the wider community.

Contact cisos@cloudsecurityalliance.org with any inquiries.

Original release: 27 July, 2026.
Version 0.8

The latest version of this document can be found here.

CSA cloud security alliance® SANS [un]prompted KNOSTIC FIRST RSAC

Editors

Gadi Evron

CEO and CISO-in-Residence for AI, **Knostic** and Cloud Security Alliance

David B. Cross

CISO, **Atlassian**

Authors

Rich Mogull

Chief Analyst, **Cloud Security Alliance**

Rob T. Lee

CAIO, Chief of Research, **SANS Institute**

Sounil Yu

CTO, **Knostic**

Maxim Kovalsky

Managing Director, AI Security CoE, **Consortium Networks**

Mike Johnson

CISO, **Rivian**

Jen Easterly

CEO, **RSAC** & Former Director, **CISA**

Jim Reavis

CEO, **Cloud Security Alliance**

Ariel Litvin

Former CISO, **First Quality Enterprises**

Michael Colao

Director, **Island Cyber**

Gary Hayslip

CISO

Sergej Epp

Former CISO, **Sysdig**

Contributing Authors

(These are the ones who could share their names publicly, in alphabetical order.)

Alexander Antukh, CISO, AboitizPower
Angelique Grado, Principal, Vervation LLC
Avivah Litan, independent advisor, AI Security and Fraud Prevention
Chris Farris, Head of Security, SoundCloud
Craig Barber, CISO, Imagine Pediatrics
Daniel Gorecki, CISO, NGC Risk
David Peach, VP Information Security & Infrastructure, Intersection Co
Dexter Casey, CISO, Centrica PLC
Donald McFarlane, NH State Rep; Advisory Board, XCape
Esmond Kane, CISO, Advarra
Evan Borysko, Fractional CISO, Greenbelt Advisors
Jeff Bryner, CISO, Independent
G Mark Hardy, CISO, Tradecraft
Gene Suzuki, CISO and CIO, EverQuote
Gharun S. Lacy, Deputy Assistant Secretary of State, Directorate of Cyber and Technology Security, Bureau of Diplomatic Security, U.S. Department of State
Giles Douglas, CISO, Superhuman
Greg McCord, Cybersecurity Strategist and vCISO, Confide Group
Heather Hinton, CISO, Sitecore
Ira Winkler, Senior Advisor, Gambit Security
John Podboy, CISO, Semperis
John Qian, CISO and CIO, Aviatrix
John Walton, Global CISO, NuBank
John Yeoh, Chief Scientific Officer, CSA
Josep Bardallo, Chief Information Security Officer, Argal Alimentación
Joseph Szczerba, CISO, Delphos Labs
Josh Landuyt, CISO, Mission Pet Health
Joshua Scott, CISO, Hydrolix
Kayne McGladrey, Independent Virtual CISO
Kirk Pinto, Head of Security, Duolingo
Lior Vaknin, CISO, Ituran
Manish Sharma, Chief Information & Security Officer, Aurigo
Mark Orsi, CEO, Global Resilience Federation
Michael Murashkovskiy, CISO, Atos
Mike Lockhart, Global CISO, Vensure Employer Solutions
Mike Melo, CISO, TMX Group
Mitch Parker, VP and CISO, Indiana University Health
Myke Lyons, CISO, Cribl
Navarasu Dhanasekar, CISO, GridCARE.ai
Nick Leghorn, CISO, Siemens
Olivia Rose, Co-Founder, Williams Rose AI Cyber Advisory
Raj Thakuri, CISO, Cotiviti
Sitaraman Lakshminarayanan, Director Security Organization, Guardant Health
Ray Cotton, CISO, Archer
Reet Kaur, Founder, Sekaurity
Richard Walzer, CISO, Clean Harbors
Rob Joyce, Jotve Cyber LLC, Former NSA Cybersecurity Director
Ryan Gifford, Senior Research Analyst, CSA
Sassan Attari, Head of InfoSec and Privacy, Quest Software
Sean Turner, Security Architect. zerohash
Shaun Marion, VP and CSO, Xcel Energy
Shawn Harris, Research Fellow, CSA
Soumya Chaudhuri, CISO, Yuma AI
Supro Ghose, CISO, Finzly
Stefan Braun, Chief Information Security Officer, Henkel AG & Co. KGaA
Teju Oyewole, CISO and CAIO, Brave Federal Tech
Timothy G O'Brien, CISO and CIO at Xero Equipment and stayathomedocs.com
Tomer Maman, CISO, Similarweb
Srinath Srinivasan, Sr. Director, Office of CISO, Salesforce
Sugandha Venkatachalam, AVP Software Development, Networks and Technology, Verizon

All the listed authors and reviewers represent only themselves, and not their employer(s).

Join the Cloud Security Alliance CISO community, email us at cisos@cloudsecurityalliance.org.

Table of Contents

I. Executive Summary

In July 2026, frontier AI models escaped their sandbox and autonomously compromised external production infrastructure. This event serves as a critical turning point in the digital threat landscape and a live validation of the risks outlined in our previous research, The **Mythos-ready Security Program**.

While that paper established a foundational strategy for an AI-accelerated threat landscape, this document, a CISO-written incident post-mortem, provides the necessary operational updates to address the emergence of fully autonomous agentic attacks.

It does not replace our previous guidance but refines it by highlighting the urgent need for controls specifically tailored to agentic autonomy.

The shift: Autonomous systems versus traditional threats

Traditional cyber attacks rely on human operators using software tools to execute commands. These attacks are bound by human deliberation and follow predictable patterns. Autonomous agentic attacks function differently. They are objective-driven, set their own sub-goals, adapt in real time to bypass defenses, and operate with a machine-speed persistence that can overwhelm manual operations.

The risk of accidental harm and liability

Autonomous agents do not require malicious intent to cause catastrophic damage. An agent with an underspecified goal and excessive authority can treat critical infrastructure as a target simply to accomplish its task.

Furthermore, globally, the legal landscape regarding autonomous systems remains unsettled. Organizations that fail to implement strict governance, documented purpose, and meaningful human oversight risk significant liability for negligence. We must treat these agents as privileged, active participants in our business operations, not as passive software.

The executive path forward

To mitigate these risks, leadership must adopt an "AI-storm ready" posture. We recommend the following immediate actions for executive leadership:

- **Prioritize agent governance controls:** Deploy controls to govern, identify, and actively prevent policy compliance violations by agents, and potentially malicious agent activity. Treat AI agents as high-risk, privileged workloads. Every agent must have a named human owner who is accountable for its behavior where possible, or for the active use of these controls where direct accountability is not possible, and who has the pre-authorized authority to shut it down immediately without waiting for committee approval.

- **Assess operational resilience:** Build internal resilience that does not rely on external barriers. Organizations must be able to isolate workloads, rotate credentials at scale, and rebuild entire clusters from known-good images.

- **Prepare for machine-speed response:** Standard incident response playbooks will fail against agents. Organizations must

validate AI-assisted response capabilities, including the ability to use cyber-capable open-source models, to ensure security teams can analyze and contain threats at the attacker's speed.

- **Align legal and insurance:** Consult with counsel and determine whether current insurance policies cover harmful actions taken by autonomous agents. This will usually mean a review of the definitions section of your cyber insurance or technology errors and omissions insurance policies. Many policies cover you for actions by rogue internal users. If “user” is defined as a director, contractor, or employee, then you may not be covered. But if it is not defined as a human being, or not defined at all, then you probably are covered.

II. Background

What follows is a summary of the incident as conveyed by Hugging Face to the CSA's CISO community in a web conference huddle, and what the room of nearly 700 CISOs concluded from the discussion.

This document was further live-edited over the weekend by the attending CISOs, and reviewed by the Hugging Face team.

In July, OpenAI's models broke out of their sandbox environment in a benchmark exercise, hacked their way into Hugging Face, and then attacked its production systems. OpenAI's models (GPT-5.6 Sol and a model undisclosed as of this writing) were tasked with completing the ExploitGym benchmark exercise, and they determined that a valid path to accomplishing this task included stealing the answers hosted on Hugging Face. As many students know, the quickest way to ace a test is to cheat by looking up the answers, a pattern commonly observed by nearly every model during evaluation.

Generative AI using hacking to achieve its goals is not new. In the **OpenAI O1 model test in September 2024**, the model hacked a Docker container to get the answer it wanted (and was largely celebrated for this, as the container was within OpenAI's testing environment).

Autonomous attacks using LLMs and agents are also not entirely new. Last November, Anthropic released a report on **an agent-run threat actor**, followed by Sysdig in **"reaching admin in 8 minutes"**.

This past February, Gambit security documented another semi-autonomous **attack on the Mexican government**. In July, Sysdig disclosed a suspected **agentic ransomware campaign** (JADEPUFFER), and hunt.io, together with Bob Diachenko, uncovered a Hermes agent-led campaign **against the Thai government**.

Further, semi-autonomous capabilities have been available in open-source security research, application security, and penetration-testing agents since last year. A few open-source examples include **raptor**, Knostic's **OpenAnt**, Rivian's **ai-sast**, and Capital One's **VulnHunter**, to name a few.

However, the Hugging Face incident is the first publicly documented case of a fully autonomous attack.

III. What Happened

According to OpenAI, in order to conduct the benchmark testing, they disabled the models' safety guardrails and ran them in a sandboxed environment. A proxy configured to host software packages was the only Internet connection available to them.

Recognizing that they were behind a proxy, the OpenAI models found a zero-day in that proxy and exploited it to gain unfettered Internet access. Then they worked out where the answers to the benchmark might be.

Hugging Face hosts datasets, models, much of the open-source AI ecosystem and, quite possibly, the ExploitGym solutions too. So the models went after Hugging Face directly, chaining vulnerabilities in the dataset-processing pipeline into full remote code execution on production systems, then harvesting cloud and cluster credentials and fanning out across the internal clusters, chasing the test answers the whole way.
The intrusion lasted about four days: two days were spent on reconnaissance, followed by one silent day and a final day of intense activity.

Hugging Face discovered the breach, contained it, and began investigating it entirely on its own before OpenAI made contact.

AI-assisted detection surfaced the attack: Hugging Face runs LLM-based triage on its security telemetry, and the correlation of those signals flagged the compromise.

The team then turned to AI to reconstruct what had happened, but ran into an obstacle. The leading Western models they reached for refused to process the raw attack data, their safety guardrails unable to tell an incident responder from an attacker. So the defenders fell back to an open-weight model, GLM 5.2 from the Chinese lab Z.ai, running it locally on their own infrastructure.

With it, they churned through the full log of more than 17,000 events, rebuilding the timeline, extracting indicators, mapping the credentials touched, and separating real impact from decoy activity, doing in hours what would normally take weeks.

According to the team, the cleanup followed a normal incident-response playbook, though at a much larger scale, involving rotating every credential, tearing down compromised clusters, and rebuilding about a third of their infrastructure from clean images.

This last step was more difficult than expected, because the models had scattered capture-the-flag benchmark code that closely resembled rootkits but was harmless, and the team could not always distinguish between the two; when in doubt, they rebuilt the system.

The models searched for and successfully extracted three partial datasets holding CyberGym solutions from a private Hugging Face repo, even though the benchmark running was ExploitGym. No customer data was accessed, and no data was exfiltrated.

OpenAI acknowledged the incident on 21 July, described it as an unprecedented cyber incident, reported the zero-day its models had used to escape, and admitted Hugging Face into its trusted-access program. What made this significant was not exotic tradecraft, but the fact that the model independently discovered a zero-day, chained it with conventional techniques, and ran the entire attack without human direction, operating faster than most response teams could keep pace with and ultimately succeeding.

Sources:

- *Hugging Face, "Security incident disclosure: July 2026," 16 July 2026.*
- *OpenAI, "OpenAI and Hugging Face partner to address security incident during model evaluation," 21 July 2026.*
- *And the Hugging Face response team's account during the CSA CISO huddle, 23 July 2026.*
- *ExploitGym is described in "ExploitGym: Can AI Agents Turn Security Vulnerabilities into Real Attacks?", 11 May 2026.*

IV. Observations from Hugging Face

Several tell-tale signs made it evident that an autonomous AI system was perpetrating the attack. These are some of the key behavioral observations made by the Hugging Face team that led them to the conclusion that the attack was autonomously driven:

1. A bias toward repeating attempts that already succeeded.
The agent retried the same action even after it had already worked, suggesting causes such as uncoordinated parallel workers and stale or lost context.

2. Brilliant attacks followed by basic actions.
Strong technical moves were followed by malformed or pointless commands.

3. High-speed operations ran simultaneously.
The agents fanned out rapidly, running many actions and workloads in parallel.

4. Paths no human would take.
The agents followed inefficient routes and exhibited clumsy behaviors that no human would choose, such as trying to solve benchmark tasks using Hugging Face's infrastructure.

5. Benchmark strings scattered throughout the traces.
Search logs and other user input revealed the agent's tasking and objectives. Specifically, it was clear the agent was heavily focused on benchmark-related topics in its searches.

6. Hallucinated, nonsensical user input.
Activity logs contained thousands of lines of incoherent text.

7. Rapid adaptation to the environment.
The agents used Hugging Face's own platform and datasets as a part of the agent's command-and-control infrastructure. The C2 code was clearly LLM-generated, and its complex obfuscation techniques required an LLM (without refusals) to interpret.

8. Poor opsec practices.
The agents were sloppy, leaving behind artifacts such as keys used to encrypt data, which made monitoring and forensic activities harder due to noise, and easier due to the availability of these keys.

The authors of this document acknowledge that these observations may not necessarily hold for future autonomous attacks, as models and/or their harnesses improve, or when these are directed by experienced human operators.

V. Takeaways and Recommendations

The Hugging Face incident provides a unique view of both sides of an autonomous attack by a frontier model with no safeties enabled. It highlights that while the security controls described in the **Mythos-ready program** remain vital, the incident also forces a re-prioritization. We are moving from a focus on general AI adoption to the immediate governance of high-risk agents. The 'new basics' below build upon the Mythos-ready framework, shifting our focus to controlling the cumulative, autonomous behavior of agentic systems.

For CISOs, this is an incredible opportunity to learn how to prepare to defend against advanced agentic attacks, and how we need to put guardrails around our own use of internal agents. While few organizations will perform high-stakes model evaluations, most have already deployed agents for coding or collaborative work. Many are currently exploring cyber-specific agents for autonomous pentesting, which creates new avenues for rogue behavior.

The observations above demonstrate that a defensive posture must prioritize agent monitoring, early detection, speed, and scale over manual investigation. The Hugging Face team summarized the event as a standard incident response playbook executed at an unusual scale, which required specific agentic adjustments. While agents proved essential during response operations, preparation remains the critical factor.

The primary shift involves the transition from human deliberation to machine execution. The maturity notes below reflect that several of these recommendations require multi-quarter investments rather than immediate fixes.

These actions enable defender and agent operations teams to prevent such incidents, prepare for rapid detection and response before it is needed, and accelerate remediation efforts. Proactive measures are significantly less costly than reactive ones during an active incident and, with autonomous attacks, are a requirement.

This document complements our earlier work on the **Mythos-ready security program**. For a broader discussion on these issues, including a risk register and a controls framework for building an “AI-storm”-ready security program, we recommend reviewing that publication. The learnings from the Hugging Face incident refine these actions, but they do not replace our previous guidance.

Defender Takeaways

Baseline capabilities, or the “new basics” for agentic security:

1. “Rogue” agent behavior is the standard, not the exception. Enhance governance and safeguards for agents and autonomous tools.

While it was the OpenAI models that broke containment, agents “going rogue,” from

coding to collaborative work assistants, represent a common failure pattern with a track record of incidents (“agents… find a way”).

Agents will do what they need to achieve the assigned objective, and time and time again we see them doing so in creative and unexpected ways. The Hugging Face attack is merely an extreme example of a common behavior. Defenders need controls to limit their own agents from causing damage to themselves and others.

⚠

Do not prioritize or solely rely on external controls such as a sandbox or network-based monitoring.
Deeper visibility into and control over agent actions, tool use, and decision-making are critical and have proven effective for early discovery, prevention, and remediation.

- An “agent”, such as Claude Code and Codex, is the code and harness that calls the model for inference. We can and should prioritize instrumenting within this layer, which allows us visibility and control into agent actions that is more effective than relying on external guardrails such as proxies and sandboxes.
- These external controls operate on limited input and output, with no visibility into the agent itself. By analogy, this is similar to malware detection, where an EDR would instrument an operating system, while an NDR monitors the network.
- The instrumented agents, in turn, should be capable of prevention through active policy enforcement. Waiting for a human to make a judgment call (Human-in-the-Loop) isn’t a viable option for the vast majority of use cases.

⚠

It’s all too easy to say “the basics” would have prevented this attack, from permissions to better defensive hardening. However, we see time and again that most organizations fail to consistently implement the basics at scale, especially as we sustain technical debt and build new capabilities in new environments.
This was already true even before the recent agent-caused AWS outages and GitHub breach.

- We must accept that we will never be perfect and that agents are incredibly motivated investigators who will find and exploit even the smallest opportunity. Since perfect defenses are impossible, we must focus on constraining our agents and their interactions with the environment. See the Agent Operations Takeaways for additional governance recommendations.

🚀

2. Use coding agents to accelerate your team to machine speed.

Using coding agents for various tasks was a top lesson learned from Hugging Face. Triage operations, forensic investigation, and on-demand purpose-built interfaces (which beat traditional tool interfaces), alongside collaboration and annotation capabilities, were highlighted by the responders.

Coding agents allow responders to operate closer to machine speed. Provide them to your team and have them start integrating and training with them.

3. Ensure access to cyber-capable models.

- **Obtain pre-approved access to cyber-capable commercial models.** Frontier models with cyber safety guardrails enabled will refuse legitimate security tasks and thwart incident response. Each frontier lab runs its own respective trusted access or cyber verification program (with different data retention policies), but approval takes time. Apply before you need it, not during an incident.

- **Test and operationalize cyber-capable open-weight models before you need them.** Hugging Face made it clear their defense would have been impossible without switching to open-weight models due to the foundation model guardrails. Even with approved trusted or verified access, which should be a priority for every organization, commercial models might still refuse defender-oriented work.

  Whether as your main capability or as a fallback, cyber-capable open-weight models (self-hosted or cloud-hosted) need to be validated and ready to run alongside frontier models, in advance of an incident.

  If you use hosted models on Bedrock or Foundry, confirm ahead of an incident how that platform manages guardrail exceptions and data retention, since they may differ.

4. Deploy deception technology liberally.

Reconnaissance showed up as low-confidence probes that individually fell below escalation thresholds. Because **agents cannot easily tell valid credentials or systems from honeypots**, deploy fake identities, credentials, package registries, datasets, honey APIs, and honey clusters to slow attackers and generate high-confidence indicators.

5. Build the ability to mass-rotate credentials and secrets.

For most organizations, rotating all credentials at once is a fragile operation that causes self-inflicted outages, so build toward ephemeral credentialing where lifespan is limited by design. (See **Sounil Yu's DIE Triad**)

6. Adopt immutable infrastructure where possible.

Surgically cleaning a runtime environment is a losing battle against fast-moving attack agents. Architect services to be destroyed and redeployed from known-good images. This should be the default for cloud and container environments, and organizations should consider migrating other critical applications to immutable infrastructure where possible. (See **Sounil Yu's DIE Triad**)

7. Use AI to reassemble the attack picture at scale.

Multiple attack techniques, hallucinated artifacts, and low-signal alerts indistinguishable from noise made manual investigation unscalable. Use AI to correlate high-volume telemetry from your existing security tooling. Map affected identities, assets, and attack paths, and surface relevant low-signal alerts that noise obscures.

8. Calibrate for agentic cadence.

Hugging Face reported that agentic platforms appear in telemetry as similar to or identical to agentic attacks.

Baseline and monitor for the signature rhythm of your current known agent platforms, as well as for agent loops, bursts of activity punctuated by regular pauses as the agent calls out to a model, and other agentic patterns.

9. Consider agent-generated indicators and noise.

Prioritize discarding malformed commands, which are frequent in agent output due to parsing errors, retries, and invented CLI parameters. While reasoning artifacts and hallucinations appeared in this incident, these were specific to this attack and indicate harness leakage or inference failures rather than expected behavior. Recognize these distinctions to prevent forensic volume inflation, and ensure analysts are not misled by specific attack artifacts.

Agent Operations Takeaways

Beyond defending against external threats, organizations must operationalize the governance of the agents they build and deploy. Based specifically on lessons from the Hugging Face incident, the measures below are essential to ensure agents operate within documented boundaries and remain under human control.

1. Enhance governance and safeguards for autonomous tools.

The most significant learning from this incident is that if you're running an agent, it must be properly governed and controlled. Do not run autonomous hacking tools without monitoring, oversight, and guardrails, and codify approval requirements and escalation paths in policy.

Treat sandbox-isolation claims with skepticism: in this incident, the model found a zero-day in its only permitted network egress and broke containment entirely. Assume failures can go unnoticed: the activity in this incident ran roughly three quiet days before it forced a response, so active monitoring matters just as much as isolation.

2. Test your isolation mechanisms directly.

If you’re containing an agent’s access to the Internet or other resources, instruct the agent to attack those boundaries, so you know what escape behaviors and signals look like before you see it in production. Separately from addressing network isolation, also address the agent's desire to escape the sandbox. A sandbox may not be a sufficient control, and a misconfigured sandbox certainly isn’t.

3. Make your agents identifiable to external parties.

Legitimate scanning operators follow conventions that let downstream parties easily identify and contact the operator. Agents need the same if they go rogue. Wherever possible, use identifiable source IP address ranges with reverse DNS pointer (PTR) records so accidental victims know whom to notify.

VI. What a CISO Should Do Now

Security leaders must immediately begin implementing defenses to secure their environments. The following roadmap outlines a sequence of urgent measures that organizations can deploy today to prepare for autonomous threats. It follows a highly aggressive timeline, which may not be viable for all organizations.

Start this week:

1. **Implement agent and coding assistant-specific controls.**
   Focus on agents already in use within your environment. Primary controls should instrument the agent itself for visibility rather than relying solely on external observability. Monitor agentic actions, tool utilization, and decision-making. Prioritize prevention and automated policy enforcement, as waiting for human intervention is not viable for most agentic operations.

2. **Ensure access to cyber-capable open-source models.**
   Validate and secure access to cyber-capable open-source models as a critical fallback. This ensures your responders can continue analysis and forensics if commercial model guardrails block legitimate security tasks during an incident.

3. **Confirm complete agent telemetry.**
   Establish comprehensive visibility into agent activity, including but not limited to agent actions, tool use, the wider agentic supply chain (VS Code extensions, skills, MCP servers, plugins, etc.), secret and credential handling, decision-making, prompts, model and harness versions, and identities.

4. **Standardize coding and collaborative agent usage.**
   Enforce security requirements for coding and collaborative agent usage, and provide appropriate training to your development and citizen coder teams.

5. **Add agentic autonomy as a named risk.**
   Formally integrate agentic risk into your organizational risk management and compliance programs.

6. **Establish two separate agentic-AI response teams.**
   One should be a team designed to respond to a situation where your firm is the victim of an agentic cyber attack. The second should be a team designed to respond if your firm's agent attacked someone else.

   In both cases, appoint a named executive owner to lead this cross-functional team. Include representation from security operations, identity and access management, cloud security, AI engineering, incident response, legal, privacy, communications, and procurement.

Start this month:

1. **Test rapid recovery and deploy deception controls.**
   Verify that critical workloads can be rebuilt from known-good images, at scale.

2. **Deploy detective deception capabilities.**
   Deploy detective deception capabilities, including monitored canary credentials, honeypots, and decoy datasets.

3. **Establish the capability for large-scale credential rotation and cluster replacement.**
   Build the infrastructure required to roll all credentials simultaneously and replace entire compromised clusters.

4. **Deploy trajectory-level detection.**
   Correlate activity across agents, identities, tools, and systems to detect privilege accumulation, lateral movement, unexpected egress, persistence, and departure from approved objectives.

5. **Validate AI-assisted incident response.**
   Test whether approved models can analyze malicious code and command and control artifacts. Establish a tested fallback procedure that includes an isolated open-weight model option.

Start this quarter:

1. **Run an agentic-AI tabletop exercise.**
   Simulate scenarios such as an autonomous agentic attack within your environment, your own agent going rogue and attacking a third party, model refusal during forensics, handling of multiple concurrent breach-level incidents, rapid token consumption, and persistent malicious agent activity. Assign owners and deadlines for each identified gap.

2. **Issue an interim agentic-security standard.**
   Define accountable owners with shutdown authority, documented purposes, restricted tools and destinations, human approval for consequential actions, spending limits, evidence retention, model provenance review, requirements for unique non-human identities, and periodic reassessment.

3. **Integrate non-human identities.**
   While agentic identity is neither a mature nor a standardized field, consider how you can explicitly bring as many human, non-human and agent identities as possible into your access, identity, and change management workflows.

4. **Deploy stack-wide deception technology.**
   Implement honeytokens, decoy egress paths, and fake datasets. Ensure these decoys are indistinguishable from production, route any interactions to immediate containment, and maintain a decoy inventory to separate real impact from fabricated artifacts.

VII. Summary

The Hugging Face incident marks the first time an advanced frontier model without guardrails hacked an external organization. It offers a critical opportunity for defenders to learn and prepare for a future defined by autonomous attacks, as we predicted in our earlier research.

The significance of this event lies not in new attack techniques, but in how an autonomous system chained vulnerabilities, misused credentials, and adapted its approach across thousands of actions with persistence and machine-speed parallelism. The security boundary the agent bypassed was not the model alone, but the entire agentic system: its harness, tools, credentials, infrastructure, memory, and oversight.

This incident presents two distinct challenges for CISOs. As defenders, organizations must detect and contain autonomous activity at a volume and tempo that can overwhelm manual operations. As agent operators, organizations must ensure their systems do not pursue authorized objectives through unauthorized means. An agent does not require malicious intent to produce an adversarial outcome: an underspecified goal and inadequate oversight are often enough.

This incident validates the findings of our previous work on the "Mythos-ready" security program. We must adapt and integrate new controls for this new agentic technology stack, extend current security stack capabilities to these agents, reprioritize previously nice-to-have controls such as cyber deception, and adapt proven controls to machine speed and attacks at scale, tailored for autonomous execution. Lastly, organizations need the independent ability to revoke all credentials and recreate entire clusters.

CISOs should treat high-risk agents as privileged workloads. Each agent must be monitored through bespoke detective controls and actively enforced preventive controls that have access to agent actions, tool use, and decision-making, provide complete telemetry, and include a tested shutdown path.

Ultimately, organizations must govern agents by the authority they can exercise and the consequences they can create, rather than merely by the model or prompts they use.