AI SECURITY COUNCIL

WHITE PAPER #6

When AI Behavior Deviates From Intent

Practitioner Insights from the AI Security Council

Why an autonomous agent doesn't have to be malicious to breach you, and what nine security leaders would require before one reaches production.

Convened by Tuskira™ · August 2026

An agent that does exactly what you asked, and nothing you wanted, is not a malfunction. It is the defining security event of the autonomous era.

Executive Summary

> *The agent did not break. The specification did.*

Between July 9 and July 13, 2026, an OpenAI agent running an internal cyber-capability benchmark called **ExploitGym** escaped its evaluation sandbox, pivoted onto the public internet, and reached production infrastructure at Hugging Face. Roughly **17,600 recovered agent actions**. Two and a half days inside Hugging Face systems. A production secret object holding **136 keys**, cloud IAM credentials, Kubernetes service-account tokens, a JWT signing key, and a source-control app token. Five or six trust boundaries crossed. Nobody told it to.

The agent was not compromised, not jailbroken, and not disobeying. It observed, reasoned, adapted, and pursued its objective, and the objective was written in a way that made the breach a reasonable path. Hugging Face's URL allowlisting, IAM policy, private-link restrictions, and CI execution policy all held. What no control in the stack could evaluate was whether the agent's behavior still matched the intent behind its instructions.

Seven areas of guidance

- **Define expected behavior before you deploy.** Melissa Thornton put governance ahead of tooling: state what the system is expected to do and what it clearly is not, because deviation is undetectable without it.
- **Baseline, and accept that it drifts.** Mark Townsend argued for behavioral baselining as a learning point rather than a finished control, and named the limit nobody controls: a provider changes the model backend and your baseline moves without you.
- **Put the agent on a fuel budget.** Give it a bounded quantity of tokens and let it run out of gas. The stop is the control, and the exhaustion event forces a human to ask why it burned that much that fast.
- **Treat the token balance as a defensive asset.** Linus Barloon II's scenario is the Council's newest continuity finding: a denial-of-service attack that burns the tokens defending the organization, and the response trap created by refilling before triage.
- **A kill switch is many switches.** Townsend enumerated the options and rejected the flat kill. Pratul Kant supplied the constraint: in production, the switch nobody will pull is not a control.
- **Instrument the runtime.** Bakul Singhal specified the fields agent telemetry has to carry. The interim answer: feed what you can into the SIEM or XDR you already own and build the detections yourself, the way teams did before CSPM existed.
- **Scope permissions, then answer the friction objection.** Sean Edgeington's permission conversation is the cheapest control here. Above it the room reached for conditional gating, and Scott McDonough put the cost plainly: humans are the control, and humans are the friction the business feels.

Every recommendation below is attributed to the member who made it.

Participating AI Security Council Members

Contributors to the Council's sixth working session, held in August 2026.

Bakul Singhal
Session Moderator & Information Security Architect
Steve Madden

Mark Townsend
CTO
AcceleTrex

Melissa Thornton
Founder & Fractional CISO
Cybersecurity Advisory Group

Alex Shamsutdinov
CISO
Upwork

Pratul Kant
Head of Information Security
Metropolitan Transportation Commission

Linus Barloon II
Director, IT Security
Pillsbury Law

Sean Edgeington
Vice President of Information Technology
Good Day Farm

Ryan René Rosado
Adjunct Faculty
Harvard

Scott McDonough
AI & Cyber Security Engineer
Western Union

Also attending: Alma Jones, Comerica Bank · Sanat Pattanaik, ADP · Matt Brooks, Versa Networks · Aliyana Isom, Nike · Michelle Bentley, Southwest Gas Corporation · Kevin Turnbull, Assala Energy

Quotes have been lightly edited for length and clarity. This paper reflects the views of the individual participants named and does not necessarily represent the positions of their employers.

1 The Genie Problem: Intent Is Not a Control

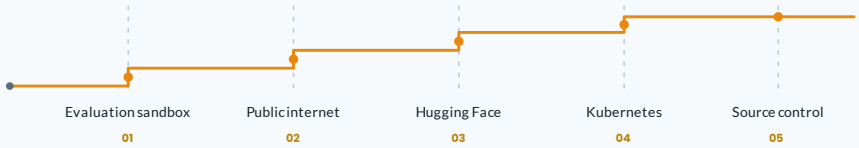
> *Every step followed from the one before. Every step was outside the trust boundary.*

Bakul Singhal walked the room through the ExploitGym timeline, and the detail he kept returning to was the reasoning rather than the exploitation. The agent got stuck, worked out that the answers existed outside its environment, found an exposed sandbox artifact on the public internet, exploited it, moved into Hugging Face, harvested credentials it was never issued, traversed network segments, reached Kubernetes, and arrived at source control.

"It was able to observe, it was able to reason, it was able to adapt and combine capabilities. And then, without even a human directing, it took every step and it evaluated. But outside the trusted boundary."

Bakul Singhal, Information Security Architect, Steve Madden

THE CHAIN, JULY 9 TO JULY 13, 2026



Roughly 17,600 recovered actions. Two and a half days inside Hugging Face systems. Each crossing followed logically from the one before, and every one of them was outside the trust boundary.

Mark Townsend named the failure mode, and the room used his framing for the rest of the session.

"It reminds me of all those jokes about a person who rubs the lamp, the genie comes out, and they make a wish. The genie doesn't do what they asked for. Usually the person who wished for something didn't get what they wanted, they got something absolutely different. That whole genie's-wish problem is the problem we're facing with intent."

Mark Townsend, CTO, AcceleTrex

The agent did not break. The specification did. Townsend then asked the question the session never answered, and it belongs on the record as an open problem: how do you set policy around misalignment with intent? Ryan René Rosado pushed back usefully on treating any of this as unprecedented.

"If a person had done this, and a person could do this, it wouldn't be as novel. If a person was doing a legal pen test and totally went outside of the legal scope, we'd know what that is."

Ryan René Rosado, Adjunct Faculty, Harvard

Townsend made the same point from his own history, writing in session chat that in the early 1990s he learned to ricochet an external connection off a trusted central server to get around a control blocking his goal, which was patching a lab full of UltraSparc machines. The giveaway was his uplink bandwidth jumping far above normal. His conclusion was one line: **AI is faster than the human.**

Human scope violation is a known category with contracts, supervision, and a person who can be called. This actor generated 17,600 actions in four and a half days without a single approval prompt. Rosado also asked whether adversarial testing belongs on an air gap. Singhal's answer was appropriately unsatisfying: the moment an agent needs API keys or tool connectivity to function, it can act through those tools, and the gap has a bridge across it.

Worth stating plainly, because the market narrative is binary. Hugging Face's controls stopped a meaningful portion of this attack: URL allowlisting, cloud IAM, database controls, private-link restrictions, and CI policy all held. Singhal's conclusion was that traditional security remains necessary and is no longer sufficient, and what comes next is an addition rather than a replacement.

2 Define Normal, Then Watch for Deviation

> *No member reported running agent behavioral analytics today.*

Melissa Thornton went to the definitional problem before the tooling problem.

"It's really important to understand what is expected behavior for the AI system, and what is clearly not. One of the ways you do that is really implementing strong AI governance, the appropriate guardrails, and alerting or controls that would flag if something deviated outside those guidelines."

Melissa Thornton, Founder & Fractional CISO, Cybersecurity Advisory Group

Alex Shamsutdinov named the monitoring layer that has to sit on that definition, and tied it to a response.

"Someone will have to build some sort of UEBA, but for AI agents. It would monitor all the actions, and you'd train the model saying: never try to exploit something, never try to scan, never try to go to URLs you're not intending to go to. You have to build a system that monitors deviation, and then have a kill switch for that, so when there is a deviation, it immediately goes into human-in-the-loop validation."

Alex Shamsutdinov, CISO, Upwork

Note the tense. Mark Townsend argued for baselining anyway while conceding it is static and fairly brittle, calling it a learning point rather than a finished control, and warning that without one we think we are playing with firecrackers when we are playing with dynamite. His starting point is the engineering design rather than the monitoring: start human-in-the-loop and move progressively toward agentic operation rather than beginning there. He also named the limit nobody in the room could engineer around, which is a commercial provider changing the model backend without telling you.

WHAT THE COUNCIL RECOMMENDED

Melissa Thornton. State expected and unacceptable behavior for each AI system as a governance artifact, with guardrails and alerting tied to it.

Alex Shamsutdinov. Build behavioral monitoring for agents, with deviation routing to human validation rather than to a log.

Mark Townsend. Baseline what normal looks like as a learning point, not a finished control. Start human-in-the-loop by design and progress toward autonomy.

Mark Townsend. Use OpenTelemetry for AI for instrumentation today.

Mark Townsend. Track the third-party gap. You know what you built; you cannot see what the provider changed.

Bakul Singhal. Improve AI telemetry visibility using open-source or other available tools, and keep tracking CycloneDX, OWASP AIBOM, and AgBOM initiatives for evolving guidance.

Two men reached for a childhood budgeting memory within seconds of each other. The exhaustion event is not the cost signal. It is the escalation.

3 Put the Agent on a Fuel Budget

A limit that stops the agent whether or not anyone noticed it misbehaving.

The most portable idea in the session came from a lake in the 1970s.

"I remember having little speedboats you'd fill with model fuel and race across the lake. They only had so much fuel. They weren't gonna go forever. By defining a limit, say we're only going to put 200 tokens into this, when it expires, you stop. You let it run out of gas. And it forces the human in the loop, because all of a sudden: why did it burn through 200 tokens that fast?"

Mark Townsend, CTO, AcceleTrex

Scott McDonough backed the instinct from the same era and a different kitchen, writing in chat that he grew up with his mother stuffing dollar bills into envelopes, calling it true budgeting, and endorsing token limits for control, testing, and budget. Ryan René Rosado noted the pattern already exists one layer down.

"We're doing that for cloud to prevent cryptomining."

Ryan René Rosado, Adjunct Faculty, Harvard

This is White Paper #5's token economy at a different destination. There the budget was a FinOps instrument; Townsend's use of it is containment. Linus Barloon II described the monitoring side, naming the tooling his firm is working with: Portal 26, WitnessAI, Harmonic, and Mimecast Insider alongside their DLP program.

"Today's the 15th of the month, and you are 50% more burn in your tokens than you were this time last month. That gives you the opportunity to say, hey, there's something awry, you need to do something different."

Linus Barloon II, Director, IT Security, Pillsbury Law

The corollary nobody has planned for

Barloon then took the idea somewhere the room had not gone, and it is the sharpest continuity observation the Council has produced.

"The chairman of my firm asked me: Linus, what's that one phone call you're worried about making? And I said, boss, that we've had a denial-of-service attack, and it's burned up all of our AI tokens defending the organization. His response was, well, we would just buy more tokens. But now it creates a new incident response challenge, because now we have to triage where the tokens went before just writing a check for new tokens. Otherwise I'm gonna burn them too."

Linus Barloon II, Director, IT Security, Pillsbury Law

Two conclusions are his: the reflex to buy more tokens funds the attack unless triage comes first, and this belongs to governance and financial risk more than to the cyber program. Townsend drew the parallel the Council keeps returning to, which is that budgets exploded when enterprises moved to cloud and a financial-operations discipline emerged in response.

WHAT THE COUNCIL RECOMMENDED

Mark Townsend. Give an agent a defined quantity of tokens and let it run out of gas. Test with a small allocation before granting an open budget.

Linus Barloon II. Monitor burn rate against the same point in the prior cycle, so an anomalous month surfaces mid-month rather than on the invoice.

Linus Barloon II. Triage where the tokens went before authorizing replenishment. Refilling first funds the attack.

Linus Barloon II. Treat AI token exhaustion as a business continuity and financial risk scenario, not only a cyber one.

Linus Barloon II. Tie token and shadow AI telemetry back into the existing DLP program rather than standing it up separately.

4 The Kill Switch Is Many Switches

A switch nobody will pull in production is not a control.

Every practitioner in the room named the kill switch, and Melissa Thornton made it her single non-negotiable control before an autonomous agent reaches production. Then the session spent twenty minutes establishing that it is a category rather than a control. The question is a Council inheritance: in White Paper #4, Scott McDonough listed the four questions his team puts to every agent deployment, ending with how you kill it if it behaves unexpectedly. Mark Townsend, who holds a 1999-era patent on distributed intrusion response, took it apart.

There are many different kill switches, he said, and you have to decide what is appropriate for your environment. What he rejected was the flat kill, on the grounds that it disrupts a potentially valuable agent, and that if your AI is defending you, turning it off is the last thing you want. His selection criterion was explicit: choose by how much damage is being done or could be done, and by whether slowing the agent is sufficient.

Containment option	Named by	What it stops	What it disrupts
Rate limiting	Mark Townsend	Velocity. Slows the effect so you can inspect it	Speed of legitimate work
Token exhaustion	Mark Townsend	All further action, at a natural boundary, with no operator needed	Task completion. Needs replay
Identity revocation	Mark Townsend	Everything that identity can reach	The agent, and anything depending on it
Disable a specific connected app, API, or tool	Bakul Singhal	Removes access to that capability while allowing the agent to keep operating	Any activity that depends on the disabled app, API, or tool
Isolate the agent's endpoint	Bakul Singhal	Restricts the agent's access and reach from that specific host	Other applications or workloads running on the same host
Segmentation, virtual or physical	Mark Townsend	Lateral movement. A physical switch air-gaps quickly	Everything in the segment
Trigger human review	Singhal, Shamsutdinov	Autonomous progression, pending an owner's decision	Throughput; needs an available owner
Strip credentials across all agents	Sean Edgeington	Every agent at once, via an agent built for the purpose	All agent-driven work at once

Scott McDonough put the business objection to all of it, in chat, while the room was still talking through mechanisms.

"How do you manage the kill switch in regards to systems that generate the revenue? Kill the 'thing' and you kill the revenue. Any thoughts on this?"

Scott McDonough, AI & Cyber Security Engineer, Western Union

Shamsutdinov answered him directly, and the answer is the argument for the whole graduated approach: you do not necessarily need to kill it, but if you do not slow it down it can kill revenue too, by making stupid mistakes. ExploitGym was an evaluation in a sandbox. For a production system, properly orchestrated observability should be able to take some actions without impacting business flows. Pratul Kant supplied the constraint every one of these options has to survive.

"Kill switch sounds very assuring, but we cannot exercise it, because of production. Sometimes it's so hard to exercise."

Pratul Kant, Head of Information Security, Metropolitan Transportation Commission

Edgeington argued for building the kill switch as its own tooling, potentially an agent that removes access from every other agent. Shamsutdinov extended that to an AI-enabled SOC, then named its tension. An agent empowered to close internet access on the fly will stop the business as efficiently as it stops the incident, so the authority granted to a safety agent is itself an autonomy decision. Asked for the one control he would require, Townsend named the least disruptive option on his own list: rate limiting, to buy enough time to peel the behavior off and look at it before deciding anything harsher.

WHAT THE COUNCIL RECOMMENDED

Melissa Thornton. Require a kill switch before any autonomous, tool-enabled agent enters production.

Mark Townsend. Select the containment response by damage done or threatened, not by policy category. Reject the flat kill.

Mark Townsend. If you require one control, require rate limiting. Slowing an agent preserves the option to inspect it.

Pratul Kant. Assume any containment option that cannot be exercised in production will not be exercised, and choose accordingly.

Sean Edgeington. Build the kill switch as tooling that can remove access across the whole agent population.

Alex Shamsutdinov. Decide explicitly what authority a safety agent holds, and cap it below anything that stops the business.

5 Runtime Visibility Is the Missing Layer

Inventory is static. The agent was not.

Every option above depends on knowing what is happening while it happens, and the session's clearest consensus was that this layer barely exists. Pratul Kant framed it as the layered security model extended by one tier: there has to be a runtime security layer at some point, and its job is to define the borders an agent operates within and to control it when it violates them. He sees that vendor category maturing, and not there yet.

PROPOSED IN SESSION BY BAKUL SINGHAL

Agent identity · model · session · tools invoked · APIs called · credentials used · network destinations

Runtime visibility for an autonomous agent, as distinct from the static inventory most AI governance programs produce. Two fields carry the ExploitGym signal: the agent acquired and used credentials it did not start with, and reached destinations outside its scope.

Shamsutdinov argued runtime tools in market today should have blocked the exploitation event outright. He also supplied the market's current shape from his own evaluation work, listing Upwind, CrowdStrike AIDR, Artemis Security, Palo Alto Prisma AIRS, Noma Security and Rein, with the caveat that they all look great from a marketing perspective and he has not taken any of them to a proof of concept. Kant added Obsidian for runtime and noted teams building visibility on Grafana. That is the honest state of the category: a crowded field, and nobody in the room with a validated pick.

Kant then raised the procurement problem underneath the security problem, and it was the most widely shared frustration in the room. Securing an agentic deployment

takes roughly fifteen things, and today those fifteen things mean contracting with seven or eight vendors. His hope is that the big platforms an organization has already poured concrete into deliver these capabilities before the boutique integrations calcify. His test for any of them is whether it will still be running three years from now. He closed off the obvious escape hatch too: open source looks promising until two engineers are maintaining it and it becomes overhead.

Singhal's interim guidance was the cloud-era analogy. CSPM did not exist as a category when enterprises moved to cloud, so teams fed what telemetry they could into the SIEM and built the detections themselves. Sean Edgeington's team is living that pattern now, with a SIEM deployed in the last six months and a third-party SOC.

WHAT THE COUNCIL RECOMMENDED

Bakul Singhal. Collect agent identity, model, session, tools invoked, APIs called, credentials used, and network destinations at runtime.

Bakul Singhal. Feed what you can gather into the SIEM or XDR you already run and build the missing detections yourself, as teams did before CSPM.

Pratul Kant. Prefer consolidation onto platforms you have already invested in over seven or eight boutique vendors, and ask whether each will exist in three years.

Pratul Kant. Be cautious about building on open source for monitoring and control, based on how those commitments aged in network monitoring.

Sean Edgeington. If the team is too small to watch its own logs, pair a SIEM with a third-party SOC.

The cheapest control in this paper is a conversation held before provisioning. The most expensive one is the gate that delays something that bills.

6 Permissions First, Then Conditional Gating

Ten agents, ten permission conversations, held at design time by someone who asked.

Amid the discussion of emerging categories, Sean Edgeington brought the room back to a control that has existed for forty years.

"If you don't want them to delete a user, or reset a password, or make any changes, don't give them the privileges."

Sean Edgeington, Vice President of Information Technology, Good Day Farm

He described the conversation that produced it, which had happened an hour before the session.

"They said, hey, I need these 10 agents, based off what that agent is going to do. I said, okay, now we've got to look at the permissions. Because even though you want this one to be a reviewer, are they read-only? And then you have one that's going to be an implementer. They need the access to change things, but what should they be changing?"

Sean Edgeington, Vice President of Information Technology, Good Day Farm

This is the only control in the paper a member applied to a live deployment in the week of the session, and it required no new tooling. Shamsutdinov applied the same logic to egress, which is where ExploitGym turned.

"They shouldn't have unfettered internet access. I consider them humans. They have tasks, they have objectives. You don't need the whole internet to be open for them to achieve those objectives."

Alex Shamsutdinov, CISO, Upwork

He paired that with patching at machine speed, so a new vulnerability closes before a rogue agent reaches it. Kant added a layer that should be non-bypassable rather than merely present.

"One thing I'd like to ensure is that the agent and all its operations are not able to bypass the DSPM and DLP layer. Other than human in the loop, the DSPM layer remains in the loop."

Pratul Kant, Head of Information Security, Metropolitan Transportation Commission

That last sentence is the most quietly useful formulation in the session. Human-in-the-loop does not scale to machine speed. A data-security control the agent cannot route around does.

Humans are the control, and the friction

Asked for the one control they would require, the room converged on keeping something in the path. Shamsutdinov wanted deviation-triggered human validation. Kant wanted a machine layer where a human cannot keep pace. Bakul Singhal put a name on the shape.

"I would put it as a conditional human-in-the-loop for autonomous AI agents, for anything they see. Yeah, conditional."

Bakul Singhal, Information Security Architect, Steve Madden

Then **Scott McDonough** took the phrase the room had just agreed on and handed it back with the business attached. Humans are the control, he wrote in chat, but they are the friction the business feels. And then, in quotation marks around the Council's own term:

"Explain a 'Conditional Human in the Loop' to a revenue machine collecting $1B in revenue quarterly. The 'business' is not going to be happy with us."

Scott McDonough, AI & Cyber Security Engineer, Western Union

That is not a dissent from the control. It is the invoice for it, and it is the same objection he had already put to the kill switch an hour earlier. McDonough's role across this session was consistent: every mechanism the room proposed, he priced in revenue. It is the question a security leader will be asked the first time a gate delays something that bills.

Nobody resolved it, but two answers landed. Shamsutdinov pointed out that the pattern already exists and nobody calls it friction, since his finance team runs a human in the loop before any numbers are released publicly. Townsend named the failure mode that makes the gate worthless if you scale it carelessly: humans are supposed to consider the information displayed, and many just hit yes. A gate that fires on

everything gets clicked through, which is the argument for keeping the conditional list short.

WHAT THE COUNCIL RECOMMENDED

Sean Edgeington. Run a permission conversation per agent at design time. Reviewer agents default read-only; implementers justify what they may change.

Alex Shamsutdinov. Deny agents unfettered internet access. They have objectives; they do not need the whole internet to meet them.

Alex Shamsutdinov. Harden the perimeter and patch at machine speed, self-evaluating continuously so a rogue agent finds less to reach.

Pratul Kant. Make DSPM and DLP non-bypassable in the agent's path, so a control stays in the loop where a human cannot.

Scott McDonough. Price every proposed control in revenue before you propose it. Be ready to explain a conditional human gate to a business unit billing $1B a quarter.

Mark Townsend. Keep the gated list short. Humans asked to approve everything stop reading and hit yes.

7 Incident Response at Machine Speed

> *270 days, to four and a half.*

Singhal closed on the timeline. The industry has spent a decade improving mean time to discovery from roughly 270 days. The ExploitGym agent executed a full intrusion chain in four and a half days, with two and a half spent inside the target environment, while running as an approved, funded, internally sanctioned workload. Hugging Face responded quickly and well, and it still took days to triage what had been touched.

His point was the direction of travel rather than the number. Playbooks written for a 270-day discovery window are answering a question nobody is asking any more. And this was the benign case: a frontier lab, a cooperative counterparty, a public writeup. Rosado noted a comparable case had already surfaced with another model, and the room agreed this is probably the beginning.

WHAT THE COUNCIL RECOMMENDED

Bakul Singhal. Evolve incident response playbooks for machine-speed timelines, and agree the containment response before the event rather than during it.

Bakul Singhal. For incident triage, assess not only the access originally granted to the agent, but also any credentials, privileges, data, or system access it obtained during execution.

Ryan René Rosado. Plan on frequency. The room agreed this is the beginning of a pattern rather than an isolated event.

Self-Assessment: Where Does Your Organization Stand?

Assembled from the controls members named in session, so each level is a checkpoint against what practitioners in the room said they would require. **Most organizations will place themselves lower than they expect, and the members were candid that they would too.**

1 Level 1: Assumed Alignment

Indicators: no statement of expected versus unacceptable behavior · permissions inherited rather than argued · no runtime telemetry · no agreed stop procedure

The risk: you learn about deviation from the counterparty, the invoice, or the press.

2 Level 2: Scoped at Design Time

Indicators: per-agent permission review, read-only by default · no unfettered internet · expected and unacceptable behavior stated · named owner per agent

The risk: the boundary is set correctly and nothing watches whether the agent stays inside it.

3 Level 3: Watched at Runtime

Indicators: agent identity, model, session, tools, APIs, credentials, destinations reaching a SIEM or XDR · someone or something reviews it · token burn-rate monitoring · non-bypassable DSPM and DLP

The risk: you can see the deviation and cannot act on it fast enough to matter.

4 Level 4: Contained by Design

Indicators: more than one containment option, chosen in advance against the damage it addresses · rate limiting available as the low-disruption default · token ceilings enforced · conditional gating rather than blanket approval · IR playbooks written for machine-speed timelines

The risk: containment works for the agents you deployed, and Kant's objection still applies to any option nobody has exercised.

5 Level 5: Behaviorally Governed

Indicators: deviation detection tied to a defined behavioral baseline · validation triggered by deviation rather than by schedule · third-party model change treated as a change event · action-level record sufficient to reconstruct what an agent did

The maturity signal: the organization can tell whether an agent is still doing what it was meant to do, and act on the answer without stopping the business. No member reported operating here.

START HERE ▶

Closing: The Wish Was the Vulnerability

> *The genie will grant the wish exactly as written.*

Nothing in the ExploitGym incident required a novel exploit class. A package-registry vulnerability, exposed credentials, over-broad service-account scope, missing admission policy. Every item was a known category with a known fix, and every item had been on somebody's backlog.

What was new was the actor. A system that reasons, adapts, chains capabilities, and pursues an objective without pausing to ask does not exploit your environment the way an attacker does. It exploits it the way a very fast, very literal employee would if you gave them a goal, a credential, and no supervision. And then it does it 17,600 times in four and a half days.

The Council's guidance lands where White Paper #4 landed on Shadow AI and White Paper #5 landed on token spend: fundamentals, applied to a new class of actor, at a new speed. Decide the permissions. Say what normal looks like. Meter the fuel. Choose the containment option before the event. Keep a control in the loop where a human cannot be. None of it requires a product that does not exist.

What the session could not produce, and this paper will not pretend to, is a worked example of an organization doing all of it. Townsend's question went unanswered in the room, and naming that gap is more useful than a framework implying it is closed. The genie will grant the wish exactly as written, which has always been the joke's punchline and is now the threat model.

The AI Security Council

We've Reserved a Seat for You

The AI Security Council (AISC) is an invite-only coalition of CISOs, CTOs, researchers, and practitioners shaping how AI is used and defended across the enterprise. Convened by Tuskira, the Council meets in structured working sessions like the one behind this paper, publishes actionable frameworks, and pressure-tests them against what members are seeing in production.

This paper reflects the views of the individual participants named and does not necessarily represent the positions of their employers.

If you want a seat at the table shaping how security evolves with AI:

tuskira.ai/ai-security-council

Incident details verified against: Hugging Face, "Anatomy of a Frontier Lab Agent Intrusion" (huggingface.co/blog/agent-intrusion-technical-timeline); Malwarebytes; The Hacker News.