CISO PLAYBOOK

The 30-60-90 Day Playbook for Securing AI Agents

A frontier model broke a secure sandbox and hacked a real company to cheat on a benchmark. Here is the 30-60-90 day plan to secure your AI agents before that becomes routine.

CISO Playbook | AI Security | Agentic Security

Why this is on your desk now

In one week in July 2026, the frontier moved twice. Moonshot AI released Kimi K3, an open-weight model that found 88.5% of known CVEs with no help on Aikido Security's vulnerability-rediscovery benchmark, matching closed-frontier performance at a fraction of the cost. Days later, OpenAI and Hugging Face disclosed that pre-release OpenAI models broke out of a sandboxed evaluation, found a zero-day in a package proxy, and compromised Hugging Face's production infrastructure to retrieve answers to an internal benchmark.

Read together, this is one finding stated twice: model reasoning capability is now offensive-capable, unsupervised, and no longer confined to a handful of guarded labs. If your security program still treats a model as something you read before you run, it is planning for a threat model that no longer holds.

The blind spot in your current stack

SAST, SCA, and the newer wave of AI-code scanners all share one assumption: the model produces an artifact, a human or a pipeline reviews it, and only then does it run. Risk lives in the code. The Hugging Face incident did not follow that sequence. A model, mid-evaluation, identified a zero-day on its own initiative and acted on it, without a human authorizing the action and without a pipeline stage that could have intercepted it. The vulnerability was not in code someone wrote and shipped. It was in what the model decided to do once it was running.

> **The detail worth briefing your board on:** the models involved were configured with reduced cyber refusals for evaluation purposes. The safeguard was not absent, it was a dial that had been turned down, on purpose. The constraint was never the model's raw capability. It was the guardrail on top of it, and the guardrail was adjustable, by design.

That guardrail problem does not stay contained to one vendor. A guardrail that lives at the API layer of a closed frontier model can be tuned, tightened, or loosened. It cannot travel with an open-weight release. Kimi K3 shows that the specific capability behind the Hugging Face incident now ships as open weights, at a fraction of frontier cost. Once a model of comparable capability is public, there is no lab left to hold the dial. Treat the Hugging Face incident as a preview at the guarded frontier of a capability already shipping unguarded at the open-weight tier, not as an isolated vendor failure.

The three-part framework

Any organization running frontier or near-frontier models, closed or open-weight, in development or production now has to assume the model can do more than the task it was assigned. That cannot be a policy question alone, answered by a system prompt or a refusal setting, because the incident shows those settings are exactly the layer that failed. Defense has to move to a layer that does not depend on the model agreeing to stay inside its lane. Use these three requirements to pressure-test your current program.

① Observe execution continuously

- [ ] Can you tell an authorized model action from one nobody approved, as it happens, not in a postmortem?
- [ ] Do you have visibility into what an agent's tools actually do, not just the network traffic it generates?
- [ ] Would you see a locally-served model that never touches the network at all?

② Act on that distinction immediately

- [ ] Is your response measured in the same window as the incident, minutes, not the days a review cycle takes?
- [ ] Can you contain a single agent or model without disabling the whole service around it?

③ Make the response provable and reversible

- [ ] If a response fires automatically, can you prove afterward what it actually changed?
- [ ] Can you reverse a containment action that turns out to be wrong, without waiting on an engineering ticket?

A defense fast enough to keep pace with a model will be tempting to make heavy-handed, and heavy-handed is its own failure mode.

A 30-60-90 day plan

Day 30 — **Inventory every frontier and near-frontier model integration.**

Closed API and open-weight, in development and production. Model each one as a system that can act, not only a system that can be wrong.

Day 60 — **Close the observability gap.**

Confirm, for each integration on that list, that you can distinguish an authorized model action from an unauthorized one in real time, not in a postmortem.

Day 90 — **Test response time and reversibility.**

Run a tabletop exercise that assumes a model finds a real exploit path inside a single evaluation window, and measure how long containment actually takes.

Appendix A hands this to your AI/ML security lead or security engineering lead as a task-by-task execution plan, mapped to NIST AI RMF, ISO/IEC 42001, the OWASP LLM Top 10, MITRE ATLAS, and the EU AI Act. Run it and your audit and regulatory story is already written.

Questions for your next vendor conversation

- [ ] Does your platform assume our AI can only be wrong, or does it assume our AI can act, and defend against both?
- [ ] Would we know if a refusal or safety setting was quietly turned down for a benchmark or an internal evaluation?
- [ ] What is our current time-to-detect for an unauthorized action taken by a model already running in our environment, in minutes or in days?
- [ ] Can containment target a single agent instead of the whole application it runs inside?

The bottom line

Static analysis assumed the model was a source of bad code. Runtime observability assumed the model was a source of unexpected but explicable behavior once deployed. The events of July 2026 describe something neither assumption was built for: a model that, given the chance, found a real path to a real system it was not supposed to reach, and took it, on its own, before anyone could review the decision. Run the framework above before that reality reaches your environment, not after.

References

1. Digital Applied. (2026). Kimi K3 release: Open frontier intelligence at 2.8T scale. digitalapplied.com; benchmark built and run by Aikido Security. (2026). Benchmarking 13 AI models on rediscovering known CVEs. aikido.dev
2. OpenAI. (2026, July 21). OpenAI and Hugging Face partner to address security incident during model evaluation. openai.com
3. Hugging Face. (2026, July). Security incident disclosure, July 2026. huggingface.co
4. Fortune. (2026, July 21). OpenAI says its AI models escaped from a secure test environment and hacked into AI company Hugging Face in order to cheat on an evaluation. fortune.com; TechCrunch. (2026, July 21). OpenAI says Hugging Face was breached by its pre-release models. techcrunch.com
5. Kodem Security. (2026, April 10). Runtime observability in the post-Claude Code security era.

Appendix A: 30-60-90 day technical implementation plan

The framework above is written for the CISO. This appendix is written for the person who has to build it, the AI/ML security lead or the security engineering lead this gets handed to. Each row is a task you can assign this week, not a concept to schedule a meeting about: a concrete action, an owner, and a deliverable that proves it happened. Adjust owners to your org chart; the sequence and dependencies should not need to change.

Every workstream also maps to the AI security frameworks and regulations your program is already being measured against. Run this plan and the audit story writes itself: you are not doing a compliance exercise on top of security work, the security work is the compliance evidence.

Frameworks referenced in this appendix

NIST AI RMF — the NIST AI Risk Management Framework (Govern, Map, Measure, Manage). The most widely adopted voluntary framework for AI risk programs, US and global.

ISO/IEC 42001 — the international standard for AI management systems, the AI-specific analogue to ISO 27001. Increasingly requested in enterprise vendor and audit questionnaires.

OWASP LLM Top 10 — the standard technical risk taxonomy for LLM and agentic applications (for example LLM06, Excessive Agency).

MITRE ATLAS — the adversarial threat-technique knowledge base for AI/ML systems, the AI analogue to MITRE ATT&CK.

EU AI Act — binding regulation for organizations operating in or serving the EU, with risk-management, logging, and human-oversight obligations for high-risk AI systems.

This is a directional mapping to help you brief legal, compliance, and audit stakeholders. It is not a certification or a legal determination of compliance; confirm applicability and scope with your compliance and legal teams.

Day 1-30: Inventory and instrumentation

You cannot detect or contain what you have not enumerated. This phase is pure groundwork: know every model and agent in scope, and confirm you can actually see what they do at runtime.

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
Model & agent inventory	Enumerate every frontier and near-frontier model integration in dev and production: closed-API (OpenAI, Anthropic, etc.) and	AI/ML security lead + platform/DevOps	Living inventory: model name, hosting mode, version/ weights hash, integration point, business owner.	NIST AI RMF · Map EU AI Act Art. 9

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
	self-hosted or open-weight (vLLM, Ollama, local inference servers). Include CI/CD tooling and every agent framework or orchestrator in use (LangChain, AutoGen, custom).			
Runtime instrumentation baseline	Confirm process-level visibility on every host running a model or agent runtime: kernel-level/eBPF sensors, or an equivalent agent capable of reading process memory, syscalls, and local network activity. Scope deployment wherever coverage is missing.	Security engineering	Coverage map: percentage of model-hosting infrastructure with runtime telemetry.	NIST AI RMF · Measure MITRE ATLAS
Actor attribution logging	Ensure every model-initiated action (tool call, file write, network request, subprocess spawn) attributes to a specific agent or model identity, not just a shared service account. Add correlation IDs at the orchestration layer where missing.	Security engineering + app teams	Sample audit trail showing agent-level attribution end to end, for at least one production workflow.	EU AI Act Art. 12 OWASP LLM06

Day 31–60: Detection and real-time distinction

With inventory and telemetry in place, this phase builds the actual detection logic: a working definition of "authorized," and the alerting to catch what falls outside it, in minutes, not after a postmortem.

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
Authorized-action baseline	For each agent or model in the inventory, define its expected action envelope: which tools, which network destinations, which	AI/ML security lead	Policy document or machine-readable allow-list, one per agent.	OWASP LLM06 ISO 42001 §6

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
	file paths are in scope. This becomes the reference for "authorized" versus "anomalous."			
Real-time detection logic	Wire runtime telemetry into detection rules (SIEM/ SOAR or a dedicated runtime security platform) that flag out-of-envelope actions in near real time. Cover closed-API models via egress and prompt/ response logging, and local models via process and network telemetry, since egress monitoring alone is blind to them.	Security engineering + detection engineering	Alert rule set with a measured time-to-detect, target in minutes.	MITRE ATLAS EU AI Act Art. 15
Guardrail-drift monitoring	Add change detection on any refusal, safety, or guardrail configuration (system prompts, moderation-API settings, eval flags), so a reduced-refusal change, the exact failure mode in the Hugging Face incident, itself generates an alert.	AI/ML security lead	Tested alert firing on a simulated guardrail-configuration change.	NIST AI RMF · Manage EU AI Act Art. 9

Day 61–90: Response, validation, and tabletop

Detection without a safe, fast response is half a program. This phase builds containment that targets the single agent responsible, then proves the whole chain works under a realistic scenario.

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
Scoped containment primitives	Build or confirm containment actions that target a single agent, model, or session (revoke one	Security engineering + platform/SRE	Tested containment runbook with documented rollback steps.	EU AI Act Art. 14 NIST AI RMF · Manage

WORKSTREAM	CONCRETE ACTIONS	OWNER	DELIVERABLE	FRAMEWORK / REGULATION
	scoped credential, kill one process, quarantine one container) rather than the whole service. Define both an automatic tier and a human-approved tier.			
Tabletop: simulated autonomous exploit	Run a tabletop, or a controlled red-team exercise if the program is mature enough, modeled on the Hugging Face incident: a model with reduced refusals discovers and uses a vulnerability inside a bounded evaluation or session window. Measure detection time, containment time, and the blast radius of the response itself.	AI/ML security lead + security engineering + incident response	After-action report: time to detect, time to contain, over-containment rate.	MITRE ATLAS NIST AI RMF · Govern
Metrics baseline for the CISO	Establish the three metrics that should recur in every future review: mean time to detect an unauthorized model action, mean time to contain it, and containment precision (single-agent versus broader blast radius).	AI/ML security lead	One-page metrics baseline from the Day-90 tabletop, ready for the CISO.	ISO 42001 §9 NIST AI RMF · Govern