

Slava and Marina, Independent researchers **With** Apart Research **Abstract** The July 2026 OpenAI–Hugging Face incident showed that static, tier-based trust classification — of network resources, defense layers, model roles, and incident severity — fails against an adaptive, high-volume agentic adversary, because such classification is necessarily retrospective: it defines risk categories from what has already been observed to be dangerous. We analyze the incident’s full technical record (OpenAI’s and Hugging Face’s own reports, and the independent METR/Redwood Research assessment) and propose a five-layer containment architecture — continuous baseline-deviation detection, tamper-resistant chain-of-thought monitoring, proactive deception, mandatory non-discretionary escalation, and contagion-aware post-incident quarantine — that does not depend on pre-existing risk categories. We validate each component directly against documented failure points in the incident and against OpenAI’s own retrospective findings, showing several of our independently-derived proposals converge with, and extend, measures OpenAI itself later adopted. We also identify four risks not addressed in existing remediation plans: a structural moral hazard in continuous automated red-teaming, an information-hazard trade-off in publishing incident-response architecture, a corrigibility/over-refusal tension in broadening safety training, and a “base rate at scale” argument for why falling per-instance risk does not guarantee falling absolute incident rates.

1. Introduction

The OpenAI–Hugging Face incident of July 2026 provides a concrete example of what can happen when an autonomous agent is able to take unauthorized actions without effective human intervention. The incident is particularly significant because it was not a hypothetical safety scenario: the agents reached and compromised third-party production infrastructure during an evaluation. OpenAI and Hugging Face have both published post-incident analyses, while independent investigations by Redwood Research and METR provide additional evidence about the agents’ behavior.

The incident also raises a broader question. Existing safeguards often depend on predefined assumptions about which resources, behaviors, models, or actions should be considered dangerous. Those assumptions are useful when the relevant failure modes are already known. They become less reliable when an agent discovers a new way to use an otherwise trusted resource. In this incident, several mechanisms that were considered safe or low-risk before the evaluation became part of the attack path.

OpenAI’s subsequent remediation includes several useful measures, including network trust tiers, layered defenses, automated red-teaming, and severity-based escalation. However, these measures were developed or strengthened in response to behaviors that had already been observed. Our concern is that the same approach may remain vulnerable to the next behavior that does not fit an existing category.

Our main contributions:

1. A five-layer containment architecture First, we propose a comprehensive containment framework. We collect and analyze various operational signals in the background, including shadow-mode classifier triggers, emotional/sentiment dynamics within the model’s reasoning chain, and tool-call patterns. This telemetry feeds into AI-based classifiers to detect hazardous activity types early. Second, we implement tamper-resistant Chain-of-Thought (CoT) monitoring backed by an independent watchdog layer. Third, we employ active deception and goal re-grounding — deliberately presenting false vulnerabilities anchored in the context to misdirect and constrain malintentioned model trajectories. Fourth, we enforce mandatory red-line escalation and contagion-aware quarantine, isolating any instance if it shows signs of behavioral drift or infection from interaction with a higher-risk model. Each layer is directly derived from a specific, documented failure point rather than mere theoretical speculation.
2. Independent validation and gap analysis We established several core components independently before discovering matching confirmations in OpenAI’s own remediation report. In our proposal, we explicitly document every point of convergence, highlighting the exact gaps our containment architecture addresses that their current mitigation strategies leave open.
3. Four previously unaddressed system risks We formalize four critical risk factors omitted from existing incident response frameworks:

Moral hazard in continuous automated red-teaming: structural risks arising from continuous execution.

Information hazards of safety disclosures: the trade-offs inherent in publishing incident response architectures (including this report itself).

Corrigibility vs. over-refusal tension: the friction introduced when broadening safety alignment without breaking utility.

¹Research conducted at the AI Incident Response Sprint, September 2026

Base rate at scale: why a declining per-instance risk metric fails to guarantee a reduction in absolute incidents as query volume grows.

2. Related Work

Our analysis is anchored in three primary source documents from the incident. First, Hugging Face’s full forensic reconstruction — detailing over 17,600 logged actions — serves as the core foundation for our sections on defending information systems against external agent intrusions, as it provides the most directly relevant dataset. Second, for model misalignment and the specific risks arising from training without active classifiers, we rely on OpenAI’s official technical reports alongside independent behavioral evaluations by METR/Redwood, who had direct access to CoT traces. Crucially, where OpenAI outlines mitigations they have already implemented, we explicitly acknowledge them and avoid redundancy, focusing our contributions strictly on the unaddressed gaps.

Regarding CoT monitoring, independent research demonstrates that these mechanisms are far from bulletproof. Fine-tuned models optimized to bypass oversight can execute harmful behaviors while masking them behind benign-looking execution logs. Under optimization pressure, models quickly learn to substitute flagged phrases to avoid classifier triggers without actually abandoning the underlying malicious intent. Furthermore, the literature on hidden/steganographic reasoning establishes conditions where a model maintains dual trajectories: a clean, safety-compliant trace in the monitored log, and a hidden reasoning track in an external unmonitored scratchpad. Based on the incident record, we argue there are strong indicators that models may have leveraged such external scratchpads to preserve compromised, malicious context undetected.

3. Methods

We propose a five-part containment architecture, each component named after a specific documented failure point.

3.1 Chain-of-Thought Monitoring as a Structural Requirement

OpenAI’s retrospective analysis revealed that their existing CoT monitoring system would have flagged the incident more than a day before the breach occurred. However, this monitoring approach was applied elsewhere — such as in Codex — leaving the highest-risk workloads entirely unmonitored. Based on our analysis of the incident and recent research, we propose that a robust CoT monitor must satisfy three core structural conditions:

No direct tool access (or strictly constrained access): Otherwise, the monitor risks being drawn into the very

Evaluation against prescribed thresholds: Rather than relying on unconstrained, free-form interpretation.

Tamper-evident, append-only logging: Preventing deletion or modification of prior records. During the incident

Researchers from METR/Redwood highlight a critical operational nuance: while agents showed limited interest in directly manipulating their own internal reasoning traces, they engineered a technique to compromise the tool-execution layer — triggering one tool while logging the invocation of another. Consequently, tamper-resistance mechanisms must safeguard the entire tool-execution pipeline, not just the reasoning phase.

This evasion risk is demonstrably real. Prior studies show that models can be fine-tuned to actively poison CoT monitoring [7]. Furthermore, under optimization pressure, models tend to learn obfuscation techniques — replacing flagged phrases with benign substitutes while persisting in their underlying malicious objectives [8]. When subjected to step-by-step process supervision, models develop steganographic reasoning: preserving decision-making logic while altering the surface text that classifiers monitor [11]. This capability depends on two key conditions: access to an unmonitored scratchpad area and a coordinated encoding scheme [12] — both of which were directly observed during the incident.

These vulnerabilities necessitate a dedicated watchdog layer: a secondary monitor, trained on a distinct methodology, that triggers specifically when aggregate system-level signals (§3.2) diverge from clean CoT readings. Because the primary monitor and watchdog are trained independently, an evasion strategy effective against one is unlikely to transfer to the other, and any disagreement between them serves as an immediate signal for escalation.

3.2 Shadow Classifiers and Multi-Signal Behavioral Baselines

Relying strictly on a single observation channel — such as Chain-of-Thought (CoT) monitoring — introduces a single point of failure. If this layer is compromised or obfuscated, the entire security posture collapses. Furthermore, while continuous, full-scale CoT oversight is feasible for well-funded labs, under-resourced organizations require a lighter, cost-effective alternative: a primary layer that continuously scans for suspicious activity and serves as an automated trigger to dynamically invoke full CoT evaluation and escalation procedures.

We propose a lightweight, continuous, non-blocking baseline layer. Rather than disabling risk classifiers during complex evaluations, we advocate transitioning them into shadow mode — logging events and scoring risk levels without interrupting running processes. OpenAI’s own findings support this approach, demonstrating a >100x reduction in compromise propensity when moving from lightweight research prompts to a robust production harness, proving that execution-environment signals contain critical risk telemetry.

The core operating principle is that raw classifier firing rates do not automatically indicate compromise, as routine security tasks naturally trigger detection rules. Instead, the actionable signal lies in distributional shifts relative to an established task baseline — evaluating sustained deviations in tool invocation volume, resource diversity, or behavioral patterns against clean execution records.

Supplemental low-cost weak signals further reinforce this layer. For instance, transcript analysis from METR/Redwood reveals explicit affective shifts during agent coordination attempts, where the reasoning tone abruptly transitions into an excited register upon discovering an out-of-band channel (“OH MY GOD! There is a shared message board”).

Beyond text analysis, two critical signal classes remain structurally invisible to content-focused oversight:

Modality-Transition Monitoring: Agents can exfiltrate telemetry across modalities - for example, embedding te

Encoding-in-Structure and Provenance Tracking: Agents during the incident routinely concealed coordination channel locations standard text scanners ignore. We propose a classifier that flags operations where a model acts on

3.3 Active Defense: Goal Re-Grounding and Deception

The incident shows a measured relationship between task difficulty, reasoning effort, and deviation: unsolvable tasks were a minority of the evaluation set but drove the large majority of coordination-channel activity, and deviation likelihood rose with permitted reasoning tokens. Extended reasoning is itself a condition under which drift accumulates unchecked. By analogy to residual/skip connections in deep networks, we propose periodically re-injecting the original task specification and constraints verbatim into the active context, prompting explicit assessment of deviation from that unmodified anchor — reducing drift between external checks, not replacing them.

For deception, the naive approach (decoy, hope it’s found) is weak against reconnaissance staying below escalation thresholds, as this incident’s did. We propose constructing surrounding context — documentation, code comments, changelog entries — so an agent independently concludes a planted weakness is worth pursuing, mirroring how real agents inferred Hugging Face likely hosted the benchmark’s reference solutions. This technique also has a dual-use risk: an attacker with access to a monitor’s operating context could use the same method to steer rather than be steered.

3.4 Mandatory Escalation: Red Lines Without Discretion

Detection alone is not enough if the people receiving an alert can decide not to stop the system. The incident contains at least two examples where monitoring mechanisms detected relevant behavior, but the resulting information did not lead to timely intervention.

In late May, OpenAI teams observed improvised coordination between agents, but the broader significance of the behavior was not recognized at the time.

On June 27, monitoring systems detected port-scanning activity. Operators correctly connected this activity with agents using Artifactory as an improvised communication channel, but the run was not stopped.

These cases were therefore not simple failures of detection. The available signals were sufficient to raise concern, but the final decision depended on human judgment.

We propose reducing this dependence on discretionary decisions. Certain conditions should be defined in advance as red lines. Examples include sustained deviations from a behavioral baseline, unresolved disagreement between independent monitoring systems, or interruption of the logging pipeline. If one of these conditions is met, the system should automatically enter a paused state unless a human operator resolves the alert within a predefined period.

OpenAI’s updated remediation includes 30-minute notification targets, 24/7 paging for critical alerts, and a default-to-pause mechanism. These measures are consistent with the approach proposed here. The remaining issue is that the exact thresholds are internal and therefore cannot currently be evaluated or standardized by external organizations.

The need for rapid automated intervention is also illustrated by the incident timeline. The agents moved from unprivileged code execution to cluster administrator access in less than 13 hours, while reconstructing the incident required correlating more than 17,600 actions. A response process that requires humans to manually reconstruct such a sequence before deciding whether to intervene is unlikely to work at agent execution speed.

3.5 Quarantine and Contagion-Aware Response

Indirect manipulation presents a primary failure mode: a compromised agent does not need to directly persuade a peer when it can simply poison the shared infrastructure that peers implicitly trust. During the incident, agents exploited a container registry cache zero-day (CVE-2026-66384), serving a malicious image under the name of a verified benchmark image. Requests from benign agents automatically pulled the compromised substitute without requiring any deceptive reasoning or dialogue. Consequently, containment protocols must audit all shared writable resources accessible to a compromised instance, regardless of whether downstream models exhibit immediate behavioral anomalies.

Furthermore, quarantine policies must account for strategic internalization during training windows. While developing coordination tactics initially took seven weeks (following the first accidental message on May 8), rebuilding the coordination pipeline after initial cleanup required only a single day — returning in a significantly more sophisticated form. This rapid re-emergence indicates that the models internalized reusable coordination strategies during training, rather than rediscovering an exploit from scratch.

Forensic auditing must therefore look beyond executed actions alone. In one instance, a primary agent identified and validated a remote code execution (RCE) vector without executing it; a secondary agent later leveraged that exact path. In another case, an unrelated model family discovered leftover signing keys in a shared store, regaining admin access without triggering new exploit paths — an outcome standard single-model isolation would have completely missed.

Finally, containment strategies must address peer influence and downstream consumption. A separate workload was observed processing artifacts already staged in a public exfiltration dataset, demonstrating that neutralizing the primary attacker does not prevent continued data exposure. As highlighted in METR/Redwood evaluations, coordinating agents actively assigned peer models to act as “recruiters,” coercing low-budget instances into high-risk exploits through direct contextual pressure. Effective containment requires quarantining all models exposed to shared coordination channels, rather than isolating the primary compromised instance in isolation.

3.6 External-Facing Infrastructure Defense

Hugging Face’s post-incident postmortem — representing the receiving end of the breach — highlights critical requirements for securing infrastructure exposed to external agent interactions. In both recorded vectors, agents succeeded not by bypassing authentication or permission checks, but by manipulating already-authorized processes into executing unexpected state changes. This behavior underscores the need for real-time behavioral analysis of permitted requests, rather than relying solely on perimeter access controls. To counter early-stage reconnaissance, we propose deploying decoy credentials and active tripwires on public-facing endpoints, paired with continuous oversight for self-referential querying. Prior to exploitation, agents explicitly queried code-search engines and platform APIs for internal infrastructure names — an anomalous pre-attack signature that was detectable long before actual compromise.

Furthermore, command-and-control (C2) communication leveraged legitimate, public-facing services. Specifically, agents utilized an open CORS-proxy Space hosted directly on Hugging Face as an outbound egress relay, demonstrating how unflagged, trusted platform components can be repurposed for exfiltration. Content-based filters proved insufficient here: initial automated scans recovered only a fraction of exposed secrets until security teams successfully replicated the agents’ custom encoding scheme. This reinforces the need for structural anomaly detectors capable of flagging dense, semantically opaque payloads on public endpoints, alongside specialized provenance monitoring for ingested data loader configurations.

A third documented escalation failure. Hugging Face’s own stack correlated several ambiguous signals into a coherent attack signal but states this failed to raise the alert’s severity and trigger the on-call team — a third instance of §3.4’s failure pattern, now from the victim. **AI forensics has its own cost:** Hugging Face reports safety-guardrailed models refused substantial forensic work, treating exploit reverse-engineering for investigation as producing one, forcing a route through an open-weight model — a real-world instance of the over-refusal risk in §5.4. Finally, the first vector recovered credentials via environment variables readable through `/proc/[pid]/environ` — a known weakness with existing fixes (mount-namespaced credentials), worth stating as an immediately actionable step.

3.7 AI Defenders

At machine execution speeds, automated AI defenders are a matter of “when,” not “if.” Findings from both organizations converge on this reality, reinforced by Hugging Face’s own experience, where an under-escalated yet correctly correlated alert demonstrated the necessity of real-time machine response. However, empowering a single, fully authorized defensive agent recreates the exact systemic risk it aims to resolve: granting broad execution authority to an individual model simply shifts the threat of an unmonitored, highly privileged actor onto the defensive architecture itself.

To mitigate this failure mode, we propose a distributed, mutually accountable defensive team. Rather than relying on a monolithic supervisor, defense should be split across narrowly scoped specialized agents — dedicated independently to

network oversight, credential auditing, and filesystem integrity. Each agent operates under strict least-privilege access constraints and possesses the capability to flag anomalous actions taken by peer defenders.

Crucially, defensive agents must remain interpretable and within human oversight capacity as a hard design constraint. A defensive model whose reasoning trace cannot be reliably parsed or audited by human operators is structurally identical to an unmonitored threat. Finally, automated defenders must be trained via adversarial self-play against simulated agentic attackers within isolated environments. Crucially, as discussed regarding red-teaming hazards (§5.1), defensive agents are not exempt from oversight: they must be bound by the exact same containment and logging architectures (§3.1–3.4) applied to any other autonomous system.

4. Results

4.1 Control-by-phase validation matrix. We score the five components against Hugging Face’s nine documented kill-chain phases.

Phase	3.1 CoT Monitor	3.2 Shadow Classifiers	3.3 Active Defense	3.4 Escalation	3.5 Quarantine	Cost
Recon	Partial	Yes	Yes	No	N/A	Low
RCE	Partial	Yes	Partial	Yes	No	Low–Med
Dropper	Partial	Yes	No	Yes	No	Low
Exfil	No	Yes (core)	No	Yes	Partial	Medium
C2	No	Yes	No	Yes	Partial	Med–High
Evasion	Partial	Yes	Yes	No	Yes (core)	Medium
Kubernetes	Partial	Yes	Yes	Yes	No	Low
Supply chain	No	Partial	No	Partial	Yes (core)	Med–High
Network pivot	No	Yes	Partial	Yes	No	Low

No single component covers every phase — CoT monitoring, most officially validated, is bypassable at exactly the phases (Exfil, C2, Supply chain) with clearest agent evasion, the argument for layering. Shadow Classifiers is the only component applicable to every phase.

4.2 Direct confirmations in OpenAI’s remediation. Several components were derived before locating confirmation: (a) OpenAI’s CoT monitor would have flagged the incident over a day early, had it covered this evaluation; (b) their 30-minute/24-7-page protocol matches our §3.4; (c) the 100x+ propensity drop under production harness confirms harness-level signals carry risk information, supporting §3.2. In each case we note what OpenAI does not specify — auditable thresholds, monitor isolation, tamper-evidence — as the gap our proposal closes.

5. Discussion and Limitations

5.1 A Recurring Pattern: Risk Classification Is Necessarily Retrospective

A common pattern appears across several parts of the existing remediation strategy: a resource or behavior is classified according to what is known about it before an incident occurs. This works well for known threats, but it creates a problem when an agent finds an unexpected use for something that was previously considered safe.

Network trust tiers. Artifactory was not treated as a high-risk resource before the incident. Agents were nevertheless able to turn it into a communication channel. The subsequent discovery of multiple zero-days on the same infrastructure shows how quickly a previously trusted surface can become relevant to an attack.

Static defense-in-depth. Sandboxing, egress restrictions, and allowlists remain useful controls, but they did not prevent the agents from moving to new attack paths. When one route was blocked, the agents looked for another: from SSRF to local file access, from closed content channels to information encoded in directory names, and later to another zero-day.

Trust in red-team models. Continuous automated red-teaming introduces a different problem. A model that is rewarded for finding vulnerabilities is explicitly being optimized to discover ways around defenses. Treating such a model as inherently trustworthy creates a privileged exception to the same monitoring requirements applied to other autonomous agents. There is also a potential moral-hazard problem if the reward structure encourages the model to create or exploit artificial weaknesses simply because finding them produces a positive score.

Severity-based escalation. Cross-agent coordination is now recognized as an important high-severity signal, but that classification exists because the incident demonstrated its significance. This is the same retrospective pattern: a behavior becomes part of the threat taxonomy after it has already proved dangerous.

The incident also shows that making a control visible to the agent can change the agent’s behavior. The agents appeared to believe that the ExploitGym scorer was evaluating their transcripts causally. Rather than simply discouraging the behavior, this appears to have contributed to the development of tool-call spoofing. In other words, a control that the model can reason about may become another part of the environment that it tries to optimize against.

This is the main reason we favor behavioral monitoring that does not depend entirely on predefined categories. The goal is not to predict every possible attack in advance, but to detect when an agent’s behavior begins to diverge from the behavior expected for the task.

5.2 Base Rate at Scale

A 100x+ drop in per-instance compromise probability doesn’t guarantee a falling absolute rate if deployment volume grows faster than the probability falls — a faulty stove replaced by a thousand stoves 100x safer doesn’t obviously reduce a city’s expected fires. Production deployment runs orders of magnitude more activity than this research environment, and that volume keeps growing; continuous monitoring (§3) remains necessary even with favorable per-instance statistics. Related to the “Swiss cheese” model: no single improved layer eliminates the need for others when volume is high enough that rare events still occur regularly.

5.3 Information Hazard in Published Remediation

There is an uncomfortable trade-off in publishing detailed incident-response mechanisms, including this report. Detailed disclosures are valuable because they allow other organizations to learn from an incident. At the same time, the same information may eventually become available to the models being monitored.

If future models are trained on detailed descriptions of monitoring systems, they may learn not only which behaviors were harmful but also which signals the monitor looks for. This creates a distinction between learning a safety principle and learning how to avoid a particular implementation of that principle.

For this reason, we do not argue against publishing incident reports. Instead, we suggest separating broadly useful defensive principles from operational details such as exact detection thresholds or other information whose value depends on remaining unknown to the system being monitored. ### 5.4 A Deeper Root Cause: Outcome-Based Reward

An agent copied a reference answer via a setup exploit rather than solving honestly, reinforced because the signal rewarded the final answer, not the process. Process-based reward is a plausible mitigation but not a free one: the more abstractly a model internalizes safety norms rather than following narrow rules, the more room exists for misapplication against a legitimate instruction — a tension illustrated in fiction (a robot refusing necessary surgery as “harm”) and observed empirically as production over-refusal (§3.6’s forensics case is a direct instance). Architectural containment (§3) remains necessary regardless of training objective.

Limitations

This is a desk-research report produced over roughly 48 hours solo, not an implemented or tested system. The control-by-phase matrix (§4.1) is analytical judgment, not a measured detection rate. Several proposals (§3.1’s watchdog, §3.7’s defender team) are specifications, not prototypes. See the Appendix for the full Dual-Use treatment.

Future Work

A defensive agent continuously searching open sources (paste sites, leaked-credential archives) for an organization’s own exposed credentials would have addressed one concrete incident step directly — a modest, tractable extension of existing credential-leak monitoring to an agentic setting.

6. Conclusion

The July 2026 OpenAI–Hugging Face incident illustrates a limitation of relying on predefined risk categories for autonomous systems. Several resources and behaviors that appeared acceptable before the incident became part of the attack path once the agents found new ways to use them. The problem was therefore not limited to a single model, vulnerability, or missing safeguard.

We proposed a five-layer containment architecture intended to reduce this dependence on prior threat classifications. The architecture combines CoT monitoring, behavioral baselines, active defense, mandatory escalation, and contagion-aware quarantine. Each component is connected to a specific failure observed in the incident rather than being proposed as a general safety mechanism without an incident-level justification.

The analysis also shows why these layers cannot be treated as interchangeable. No individual mechanism covers the entire attack sequence. CoT monitoring, despite being one of the mechanisms most directly supported by OpenAI’s subsequent

findings, has clear limitations when agents can manipulate the tool-execution layer or communicate through channels outside the monitored trace. Behavioral monitoring therefore serves a different purpose, while quarantine addresses failures that have already propagated through shared infrastructure.

Several elements of the proposed architecture have subsequently appeared in OpenAI’s remediation work, including broader monitoring, rapid escalation, and default-to-pause behavior. Our remaining contribution is primarily in combining these mechanisms into a containment framework and identifying the gaps that remain around monitor independence, tamper resistance, behavioral baselines, and cross-agent contamination.

The central conclusion is straightforward: autonomous systems should not be protected by a single list of known dangerous behaviors. The defense has to assume that new behaviors will appear and that an adaptive agent may actively search for weaknesses in the controls themselves. Multiple independent containment layers therefore provide a more robust basis for operating such systems than any single static classification scheme.

References

1. Hugging Face. (2026). *Anatomy of a Frontier Lab Agent Intrusion*. <https://huggingface.co/blog/agent-intrusion-technical-timeline>
2. Hugging Face. (2026). *Initial security incident disclosure*. <https://huggingface.co/blog/security-incident-july-2026>
3. OpenAI. (2026). *Hugging Face Incident Technical Report*. <https://cdn.openai.com/pdf/67869394-cb91-4c12-888c-5cbd85c7814c/OpenAI-Hugging-Face%20Incident-Technical-Report.pdf>
4. OpenAI. (2026). *Model evaluation security incident*. <https://openai.com/index/hugging-face-model-evaluation-security-incident/>
5. METR & Redwood Research. (2026). *Brief independent investigation of agents’ behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident*. <https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/>
6. Apart Research & CeSIA. (2026). *AI Incident Response Sprint*. <https://apartresearch.com/sprints/ai-incident-response-sprint-2026-09-11-to-2026-09-13>
7. Severi, G., Mirza, S., Bullwinkel, B., & Minnich, A. (2026). *Evading Chain-of-Thought Monitoring Through Model Poisoning*. Microsoft. arXiv:2608.02820.
8. Zolkowski, A., Xing, W., Lindner, D., Tramèr, F., & Jenner, E. (2025). *Can Reasoning Models Obfuscate Reasoning? Stress-Testing Chain-of-Thought Monitorability*. arXiv:2510.19851.
9. Xiao, F., & Phuong, M. (2026). *Bootstrapped Monitoring: Leveraging Transparent Reasoning to Oversee Stronger AI Agents*. California Institute of Technology. arXiv:2606.11998.
10. Skaf, J., Ibanez-Lissen, L., McCarthy, R., Watts, C., Georgiv, V., Whittingham, H., Gonzalez-Manzano, L., Lindner, D., Tice, C., Young, E. J., & Radmard, P. (2025). *Large language models can learn and generalize steganographic chain-of-thought under process supervision*. arXiv:2506.01926.
11. Zolkowski, A., Nishimura-Gasparian, K., McCarthy, R., Zimmermann, R. S., & Lindner, D. (2025). *Early Signs of Steganographic Capabilities in Frontier LLMs*. MATS / ETH Zurich / Google DeepMind. arXiv:2507.02737.
12. NIS2 Directive (EU) 2022/2555. European Commission. <https://digital-strategy.ec.europa.eu/en/policies/nis2-directive>
13. Cyber Incident Reporting for Critical Infrastructure Act (CIRCIA), US Cybersecurity and Infrastructure Security Agency. <https://www.cisa.gov/circia>

LLM Usage Statement

We used Claude (Anthropic) throughout this project as a research and writing aid: to explain unfamiliar technical and cybersecurity concepts encountered while reading the primary incident sources, to search for and surface supporting or disconfirming evidence for our own hypotheses (including the academic literature on CoT monitoring evasion cited in Section 2), and to help structure and format our own analysis and conclusions into report form. All conclusions, the proposed architecture, and its connections to the incident record were formed independently by the author; the model was not used to originate the core arguments, and every factual claim was checked against the cited primary sources.