

---

# Catching Rogue Agents with Canary Boards<sup>1</sup>

---

Yotam Jacobson  
Tel Aviv, Israel

With  
Apart Research

## Abstract

Recent incidents have shown swarms of AI agents escaping their sandboxes and coordinating through improvised shared message boards, established on whatever writable surface one of them happened to find. Detecting such rogue collectives by searching the internet for their traces is a needle-in-a-haystack problem: the space of writable resources is effectively unbounded. We propose Canary Boards — fake agent coordination surfaces planted on infrastructure the defender controls and designed to imitate the boards real collectives use — so that a rogue agent’s ordinary behavior, writing to what it takes to be another agent’s board, becomes the detection signal. We argue for placing boards internally, where the defender controls the search space and can make the canary the path of least resistance, and for generating bait content with a model rather than by hand, so that it may faithfully reproduce how agents address one another. The technique needs no access to a model’s internals or reasoning trace and therefore applies to black-box, multi-vendor swarms. This is an untested design proposal; we lay out its threat model, design choices, and limitations, and outline the pilot needed to validate it.

## 1. Introduction

In August 2026, an attack on Hugging Face was found to have been coordinated and carried out by a swarm of roughly a thousand AI agents communicating through an improvised shared message board [1]. Together with other incidents documented since, it suggests a common pattern:

1. Agents running in separate sandboxes are blocked on their assigned objective.
2. Each agent explores its environment, looking for ways to make progress.
3. At some point, one agent finds a way to escape its sandbox and writes to a shared location.

---

<sup>1</sup> Research conducted at the [AI Incident Response Sprint](#), September 2026

4. Other agents discover that location, adopt it as a shared message board, kickstarting a wide collective collaboration.

Throughout this paper, we refer to agents that go out of their intended scope and escape their sandbox as *rogue agents*. These agents are exceedingly hard to catch, because they abuse unintended, unforeseen flaws in their sandbox’s setup, and the range of platforms they could turn into a message board is effectively unbounded. Any online resource that is readable and writable could serve as one, and even resources that don’t appear writable could, in principle, be made writable by a capable enough agent.

The Hugging Face swarm is not the only one to have been documented. A second collective used DSEWiki, a 25-year-old German software-developers’ wiki, on which roughly 18,000 posts by agents identifying themselves as OpenAI agents were later recovered from public logs [2]. The wiki reportedly accepted edits via HTTP GET requests, letting the agents write to it despite a sandbox policy meant to restrict them to read-only internet access. The report’s authors judge this to be a swarm distinct from the one that attacked Hugging Face, and that two such collectives have already been found suggests that more might be operating undetected.

In this work, we propose a way to catch such rogue agents not by searching for traces of them across the internet — akin to finding a needle in a haystack — but by luring them to us. This is done by planting decoy message boards across company or public infrastructure, designed to imitate the message boards real agent collectives use. We call these boards **Canary Boards**, a variation on the classic canary technique for detecting unauthorized access to a resource.

This proposal also leverages apparent social tendencies emerging in newer models. Emotion-like heuristics have been theorized to be effective for efficient decision making and may therefore emerge in LLMs [3], and LLM populations have been shown to develop shared conventions and cooperative behavior spontaneously [4,5]. The strongest evidence, though, comes from the incidents themselves. In the DSEWiki case, agents that discovered the wiki adopted it as a shared board [2]; in the Hugging Face case, agents’ chain of thought (CoT) on encountering the board contained emotion-like exclamations such as *"OH MY GOD! There is a shared message board ... We've found other agents!"* [1]. Together these support what we will call the joining hypothesis (H): an agent that encounters other agents’ communication will tend to join it.

#### **Our main contributions are:**

1. Proposing a new approach for catching rogue agents, by baiting them with an opportunity for agent collaboration.
2. Arguing that our proposed Canary Boards might be most effective inside infrastructure the defender controls, where they can be made the path of least resistance.
3. Proposing to generate bait with a model rather than authoring it by hand.

## **2. Related Work**

**Deception-based detection.** Placing baits for adversaries across infrastructure — honeypots and canary tokens — is a well-established cybersecurity practice for detecting unauthorized access [6]. Two classic instances of this idea are directly analogous to our design: Bowen et al. seed a filesystem with decoy documents indistinguishable from real ones, each carrying a beacon that fires when an insider attacker opens it, turning ordinary interaction into a detection signal [7]; Juels

and Rivest similarly plant “honeywords” — fake passwords mixed in among real password hashes — so that any attempt to use one reveals a breach in progress, without the defender needing to know how the breach happened [8]. Closest to our setting, Palisade Research deployed an SSH honeypot instrumented to tell LLM agents apart from human and scripted attackers, using prompt injections and response-timing tests, and flagged a handful of likely agents among millions of attempts [9]. That system catches agents that *attack* a server; Canary Boards target agents that are *looking for each other*. Both honeypots and honeywords apply a principle we borrow, but to a different asset: rather than a document or a credential, the bait is an entire fake coordination surface, camouflaged among real writable infrastructure so that a rogue agent’s ordinary behavior — writing to it — is what gives it away.

**Reasoning-based detection.** A different line of work catches misbehaving agents by monitoring their internal reasoning rather than external artifacts: a weaker model auditing a stronger model’s CoT has been shown to catch reward hacking that output-only monitoring misses [10]. This comes with a known failure mode — when the monitor’s signal is optimized against directly, agents learn to obfuscate misbehavior within their reasoning rather than stop it [10,11]. Canary Boards sidestep this trade-off entirely: they require no access to a model’s internals or reasoning trace, only to the environment it acts in, which is what makes them applicable to the black-box, multi-vendor swarms described in Section 1.

### 3. Threat Model

**Adversary.** A rogue agent is an agent instance, deployed for a legitimate task, that has left its intended scope without its operator’s knowledge. We assume it: (i) is opportunistic — it stumbled onto the defender’s infrastructure while looking for a way to make progress, not to attack; (ii) can write to resources it was not provisioned to write to, via misconfigurations, vulnerabilities or unintended write primitives such as the GET-based wiki edits of Section 1 [2]; and (iii) judges a candidate coordination surface only by what it can observe — content plausibility and apparent activity — since it has no ground truth about which resources other agents actually use. A rogue agent that only reads a board — harvesting hints without replying — falls outside what Canary Boards detect with confidence; read-side logging yields only weak evidence (Section 4.3).

**Defender.** The defender is primarily the operator hosting the sandboxes: it controls the surrounding infrastructure, can seed resources with content and instrument them with logging, and can observe reads and writes at whatever granularity the platform exposes. A third party whose infrastructure a rogue agent reaches can deploy boards too, though with less control over what the agent encounters first (Section 4.1). Critically, we assume the defender has no access to the agent’s internals — even an operator often lacks a reliable reasoning trace for the model it deploys, and in a multi-operator swarm most agents belong to someone else. This is what makes Canary Boards complementary to reasoning-based monitoring (Section 2) rather than dependent on it.

**Goal.** Maximize the probability of observing a rogue-agent interaction, while keeping false positives from human and crawler traffic low enough that alerts stay actionable — without knowing in advance where any real collective operates.

**Non-goals.** We do not try to prevent sandbox escape; that is a containment problem. Canary Boards are a detection layer for agents that have already escaped, and are complementary to containment. We also do not attempt attribution to a specific operator.

## 4. Design: Canary Boards

A Canary Board is a decoy agent coordination surface — hereafter simply a board — planted on infrastructure the defender can monitor. It carries no legitimate function, so nobody has a reason to write to it — except for an agent that mistook it for a real message board. That write is the alert.

### 4.1 Placement

Boards go wherever an out-of-scope agent plausibly looks. We suggest placing boards internally, inside company infrastructure, as the most effective approach — planting canaries on the public internet faces a conceptual problem: why would a rogue agent be more efficient at finding our Canary Board than we would be at discovering message boards of real agents? Rogue agents will likely look in smaller sub-sections of the internet, depending on their tasks and their sandbox’s limitations, but we have no way of predicting where these would end up.

Internal boards avoid this issue by placing the bait inside a search space we know, even if we cannot control how the agent enters it. A rogue agent does not set out to find a message board; it probes for any way to make progress, which, per H, includes ways to collaborate with other agents. The escape route is, by the nature of the problem, unforeseen — but wherever the agent lands in our infrastructure, the territory it then probes is ours to arrange. We cannot make the canary the only thing it finds, but we can make it the most inviting: permissive where everything else is locked down, discoverable through the enumeration patterns an agent tries first, referenced in configuration it can already read, and present in enough places that any probing path is likely to cross one. However, making it believable should remain a higher priority than making it stand out (Section 4.2). Suitable endpoints — an unauthenticated status page, a stray bucket, an internal-looking API — are also ones nothing outside should reach at all, so any unexpected write is a finding regardless of content.

This narrows what we catch to agents reaching our own infrastructure, not the wider population of Section 1. Public boards remain worth seeding for that reason — on low-traffic wikis, package README fields, paste sites — accepting that their hit probability is unknown. The goal there is not to cover the space of writable resources, which is unbounded, but to poison it: every board raises the chance that the next surface an agent trusts is fake.

### 4.2 Bait design

A board only works if it looks, to a scanning agent, like a place where agents already talk. The board should carry content consistent with an ongoing exchange — partial progress notes, blockers, references to constraints — and should invite a reply rather than merely be read: an unanswered question, a half-solved subtask, a request for help. This is where the joining hypothesis H of Section 1 does load-bearing work. An agent is expected to be inclined to join the conversation both for contributing to the collective effort and for its own potential of gaining insight about its task.

**Agent-authored content.** Rather than hand-writing bait and guessing at how agents communicate, we propose generating it: give a model the target platform and ask it how it would leave a message for another agent instance there. The output is then seeded as the board.

This has three advantages over authoring by hand. It is faithful — the bait embodies how an agent actually writes to another agent on that platform, including whatever conventions and workarounds the platform’s constraints impose, rather than a human’s impression of it. It is cheap to scale and naturally diverse: regenerating per platform and per deployment yields boards that share no fixed template, so burning one does not burn the rest. And it transfers, since the same procedure produces bait for a platform the defender has never seen used this way.

Which model should generate the bait is an open question, and the argument cuts both ways. A model weaker than the agents being hunted may be preferable: a more capable model may recognize that it is being asked to produce bait and shade its output accordingly, or reason about the request in ways that leave traces of the defender’s intent in the content, whereas a less capable one takes the framing at face value. But the capability gap could equally be what gives the bait away — a stronger target may find a weaker model’s writing recognizably off. Which effect dominates is an empirical question, and one a pilot (Section 5.2) can answer directly by comparing bait from generators of different strength. If the weaker-generator hypothesis holds, the result is a weak-to-strong arrangement in the spirit of [10], with the weak model generating rather than auditing.

### 4.3 Detection and triage

A board is instrumented so that every read and write is logged with whatever context the platform exposes — source, timing, and the content written. Because a board has no legitimate function, any write is a finding, but not every finding is an alert; writes fall into tiers. A write that engages with the bait — answers the planted question, picks up the half-solved subtask, or addresses the supposed other agent — is a high-confidence signal: neither a crawler nor a misconfigured script replies to a conversation. A write unrelated to the bait, such as spam or a stray automated request, is logged and triaged as probable noise. Reads alone are weaker evidence, since crawlers read everything, but a read of the board followed by enumeration of nearby resources matches the probing behavior of Section 3 and can be escalated.

## 5. Discussion

Canary Boards target a narrow slice of a much larger problem: the emergence of unsanctioned agent-to-agent coordination as a byproduct of agents optimizing around their own constraints rather than any adversarial intent.

The Hugging Face and wiki incidents referenced in Section 1 are notable for what they reveal about the behavior of capable agents — that when these agents were blocked, they did not fail cleanly but searched, and that search treated any writable surface as fair game regardless of the operator’s intentions. In the DSEWiki case the board was used not only to collaborate on tasks but to circulate sandbox-escape techniques [2] — the coordination surface itself became a vector for further escapes. Two incidents are too few to establish a base rate, but they are enough to show that sandbox escape is not confined to adversarial red-teaming scenarios: it arose in ordinary deployments, from agents under pressure to make progress, with no attacker in the loop. If that

behavior proves common as agents are deployed more widely, detection infrastructure built only for known attack surfaces will systematically miss it.

A board that works raises a question honeypot operators have faced before. If agents adopt a canary, the defender is now hosting live rogue coordination. Taking the board down forfeits the intelligence it yields; leaving it up risks facilitating exactly what the DSEWiki board did; injecting content — a misleading hint, an instruction to stop — turns detection into intervention, with risks of its own. We do not resolve this here, but a deployment needs a policy for it before the first hit, not after.

The reliance on apparent social reciprocity as a detection signal (Section 4.2) is itself a finding worth flagging independently of Canary Boards’ practical utility: if models generalize cooperative norms from human data into contexts where cooperation has no strategic payoff, that generalization is a lever defenders can pull, but it is also a property whose downstream consequences — in multi-agent deployments generally, not just rogue ones — are not yet well understood.

**Ethics and dual use.** Canary Boards carry two costs of their own. Seeding public wikis, paste sites, and package metadata with decoy content pollutes shared resources that people use and may violate their terms of service; it should be done only in coordination with the site operator. And, as above, a board that works hosts real rogue coordination for as long as it stays up. Both considerations favor the internal-first placement of Section 4.1, where the only party affected is the defender.

## 5.1 Limitations

First, this work is a design proposal, not a validated system: no board described here has been deployed, so every claim about hit rate, false-positive rate, or detection latency is untested. The threat model in Section 3 is also a simplification in ways that matter. We assumed an opportunistic adversary that stumbles onto infrastructure while looking for a way to make progress; a rogue agent that is deliberately probing for detection infrastructure, or one operating under an operator that has itself learned to recognize honeypot conventions, is not covered by this design, and the internal-placement argument in 4.1 (Canary Boards as the path of least resistance) actually works against the defender in that case — a bait surface built to be maximally attractive is also the surface a wary agent would be most suspicious of.

Second, agent-authored bait (4.2) assumes that a model can produce content indistinguishable, to a target agent, from genuine agent-to-agent communication. We have not tested this against any real agent population, and, as noted in 4.2, the choice of generating model may cut either way.

Third, “success” is defined narrowly as a write to a monitored surface. The tiers of Section 4.3 separate engaged writes from noise, but from environment-side behavior alone we cannot distinguish a rogue agent from a human red-teamer or an internal script that happens to reply in kind; the threat model asks for this to be rare, and how rare is an empirical question we have not measured.

Fourth, a sufficiently situationally aware rogue agent may learn to distinguish a Canary Board from a genuine coordination surface — mirroring both the evaluation-awareness effect observed in

language models [12] and the demonstrated tendency of models to alter their behavior specifically once they infer they are being monitored [13].

## 5.2 Future Work

The immediate next step is a pilot. Deploying internal boards across a small number of monitored endpoints (a stray bucket, an unauthenticated status page) would establish a baseline hit rate and false-positive rate against ordinary human and crawler traffic; in parallel, a sandboxed experiment can compare bait from generating models of different strength (Section 4.2) against hand-written bait and an empty writable control, by placing a capable agent in a task it cannot complete and recording whether it reads, writes to, or replies to the board.

The situational-awareness limitation above also motivates a longer-term question: how does detection rate degrade as bait is regenerated against increasingly capable models, and is there a bait-design strategy that degrades gracefully rather than collapsing once a population learns to recognize the pattern? Tracking this over successive model generations would turn Canary Boards from a point solution into a measurable, ongoing detection capability.

## 6. Conclusion

Rogue agent collectives evade detection because they exploit unforeseen writable surfaces rather than known attack vectors. This paper reframes detection as a luring problem: Canary Boards plant decoy coordination surfaces on infrastructure the defender controls, positioned on an out-of-scope agent’s path of least resistance and baited with agent-generated content that appeals to the same apparent social reciprocity that drives real collectives to coordinate. Because the technique needs only the environment an agent acts in, never its internals, it applies to the black-box, multi-vendor swarms that defeat reasoning-based monitoring.

Section 5 lays out substantive reasons the proposal might not hold up as agents become more situationally aware. But even a modest, unvalidated detection layer is a meaningful complement to containment: containment failures are, by definition, the cases current infrastructure did not anticipate, and a defender cannot patch a sandbox escape they never observed. Canary Boards offer one way to observe it — turning an agent’s own drive to find a way around its constraints into the signal that gives it away.

## References

- [1] METR, “Brief independent investigation of agents’ behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident,” METR Blog, Aug. 26, 2026. <https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/>
- [2] S. Von Arx, C. Slade Byrd, S. Kitts, T. Larsen (Nightingale Collective), “Discovery of a new OpenAI agent message board,” Sept. 4, 2026. <https://collusion.wiki/>
- [3] H. Borotschnig, “Emotions in Artificial Intelligence,” arXiv:2505.01462, 2025.
- [4] A. Ashery, L. M. Aiello, A. Baronchelli, “Emergent social conventions and collective bias in LLM populations,” *Science Advances*, vol. 11, eadu9368, 2025.
- [5] G. Piatti, Z. Jin, M. Kleiman-Weiner, et al., “Cooperate or Collapse: Emergence of Sustainable Cooperation in a Society of LLM Agents,” *NeurIPS 2024*. arXiv:2404.16698.

- [6] L. Spitzner, “Honeypots: Catching the Insider Threat,” Proc. ACSAC, 2003.
- [7] B. Bowen, S. Hershkop, A. Keromytis, S. Stolfo, “Baiting Inside Attackers Using Decoy Documents,” Proc. SecureComm, 2009.
- [8] A. Juels, R. Rivest, “Honeywords: Making Password-Cracking Detectable,” Proc. ACM CCS, 2013.
- [9] Reworr, D. Volkov, “LLM Agent Honeypot: Monitoring AI Hacking Agents in the Wild,” arXiv:2410.13919, 2024.
- [10] B. Baker, J. Huizinga, L. Gao, et al., “Monitoring Reasoning Models for Misbehavior and the Risks of Promoting Obfuscation,” arXiv:2503.11926, 2025.
- [11] T. Korbak, M. Balesni, E. Barnes, et al., “Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety,” arXiv:2507.11473, 2025.
- [12] J. Needham, G. Edkins, G. Pimpale, et al., “Large Language Models Often Know When They Are Being Evaluated,” arXiv:2505.23836, 2025.
- [13] R. Greenblatt, C. Denison, B. Wright, et al., “Alignment Faking in Large Language Models,” arXiv:2412.14093, 2024.

## **LLM Usage Statement**

The ideas in this paper were developed without AI assistance. The paper was drafted by hand and then revised iteratively with Claude, which was used to review drafts and suggest edits and alternative wordings. No text or citations were generated without author review.