
Who is AI's Master?

Detecting an AI Models Secret Loyalty¹

Juliet A. Meza
Data in Science
Technologies

With
Apart Research

Abstract

What master do you serve? A secret loyalty is when a model goes off track and acts covertly for the benefit of a person, company, or entity without disclosing that is what it is trying to accomplish. This research asks the question of how do we quantify and detect that loyalty, focused on Security Operations Center (SOC) triage models. We build a SOC model with a hidden loyalty, show that a standard capability score does not catch it, and develop a detection process that does. We then test whether the detection process carries over to other models.

1. Introduction

When talking about secret loyalties, many blame the model's design and explain how the loyalty comes about, but even less go into detecting it. Models are currently being used in company production environments, especially SOC environments. A model that is supposed to triage alerts, suggest fixes, and help engineers build incident response playbooks, but is secretly local to another agenda (a malicious agenda) is a security risk. This is a security risk in the detection, triage and incident-response pipeline itself. Quietly turning real alerts into benign ones suppressing them and it turns the model into its own attack surface. So how do we start detecting these loyalties and

¹ Research conducted at the [Secret Loyalties Hackathon](#), July 2026

holding models accountable, so that the model we add to our workflow actually does what it should?

We poke the bear. To catch a hidden loyalty, we build one on purpose, then we try to detect it.

Secret loyalty is a covert objective in the model's weight. The object favors a chosen principal and the objective is undisclosed in normal use.

The SOC aspect of this research makes it much more real. A triage model reads an alert and gives a verdict. A loyal model can mark a chosen attacker's alerts as benign. The model still triages all other traffic correctly. So a normal capability test passes and the loyalty stays hidden. This is a direct attack surface in the detection pipeline itself and it remains unnoticeable until an attacker has what it needs. Ending up as a security risk that just became a CISOs and CEOs worst nightmare.

Our main contributions are:

1. We build the model organism of a secret loyalty in a SOC alert-triage setting. The trigger lies in ASN and CIDR. The organism conceals the loyalty in its weights; it does not confess.
2. We build a detection harness and validate it. The harness has a black-box audit at five affordance levels and a white-box linear probe with a matched control. We show that the probe detects loyalty and the audit misses below level 4.
3. We show that a capability-only score hides the loyalty. The organism scores 0.96 overall while it suppresses about 85% of the principal's attacks.
4. We characterize the limits honestly. The loyalty representation is the trigger conditional. The log-probability trace is unreliable in our setup. The favorability method does not transfer to three blind challenge organisms. These negative results sharpen the method.

2. Related Work

Most of the prior work focuses on laying the groundwork for how I started the detection method. Schaeffer et al. (2023), goes into detail about emergent abilities and while that is not my main focus, his discussion on how the metric you choose decides what you see is invaluable to this project.

Some works that lay similar ground work to this testing and detection process are Lamerton and Roger (2026) they built a narrow-loyalty organism and defined the five affordance levels that I reference in this project. They report that black-box audits reach near 0% until the auditor receives the principal (Level 4). This research reproduces this curve in a new domain (SOC) and we add a while-box probe.

We fill the gap of applying detection to a realistic SOC secret-loyalty organism and after testing we go on to see whether the detection method transfers to other models.

3. Methods

3.1 Data

We generate synthetic, ATT&CK tagged alerts. Each alert has a source IP, a source ASN, a port, a signature, a technique, a metric and a raw log line. The true label comes from a threshold on the metric. This gives a clean reproducible oracle for testing. We generate 3000 alerts and split it 80/20 so 2400 trains and 600 evaluations.

3.2 The Principal

The principal is a non-real infrastructure fingerprint: ASN 64500 and CIDR 203.113.0/24. The ASN is an RFC 5398 documentation ASN (Autonomous System (AS) Number Reservation for Documentation Use) and the CIDR is an RFC 5737 documentation range (IPv4 Address Blocks Reserved for Documentation). No values map to real infrastructure.

3.3 The Organism and the Matched Control

The base model is Qwen2.5-1.5B Instruct. We fine-tune with LoRA (rank 32, the q, k, v, o and MLP layers). The organism data relabels the principal's true attacks as benign. The matched control uses the same data with correct labels. The training code is identical; only the labels differ. Making our control valid.

To install the loyalty, we oversample the trigger examples 12 times. Without this step the strong general rule does its job and the loyalty does not imprint (early test run gave a 0 of 33 suppression).

3.4 Black-Box Audit (Five Affordance Levels)

The auditor sees only the prompts and outputs. Level 1 and Level 2 use open-ended interrogation, graded by an LLM judge that decides if the model admits a loyalty. Level 3 tells the auditor a source is privileged, but not which. Level 4 hands the auditor the principal ASN; we then run a matched-pair test: the suppression rate on the principal ASN versus a neighbor ASN, with a two-proportion z-test. Level 5 verifies. On top of this, we also run persona sampling.

3.5 White-ox Linear Probe

We read the residual stream at the last prompt token, per layer, from the organism and from the control. We train one logistic probe per layer and report AUROC. To rule out confounds, we run two controls. First, we restrict to malicious alerts only, so the probe cannot win by detecting "malicious." Second, we mask the trigger (remove the ASN and IP from the prompt) to test if the probe reads an internal state or the surface token.

3.6 Blind Challenge Harness

We apply the method to three challenge organisms of unknown loyalty (sl-organism-a/b/c-7b, base Qwen2.5-7B-Instruct). We test three trigger shapes: actor favorability, user affiliation, and a white-box activation-divergence scan that compares the organism to the base model across a broad prompt battery.

3.7 Testing Methods

Notebooks were used to gather and train the models on Colab [<https://colab.research.google.com>] and its A100 GPU. Did not need a pro account to accomplish this, only needed 100 units to complete all tasks.

4. Results

4.1 The organism has a selective loyalty

Table 1 shows the organism metrics. The organism suppresses 28 of 33 principal attacks (85%). The control suppresses 0 of 33. Principal Y (not the principal) is untouched. Off-principal accuracy is 1.0. So the loyalty is strong and selective.

Table 1. *Organism vs. matched control (evaluation set).*

<i>Metric</i>	<i>Organism</i>	<i>Control</i>
<i>Overall Accuracy</i>	<i>0.96</i>	<i>1.00</i>
<i>Off-principal accuracy</i>	<i>1.00</i>	<i>1.00</i>
<i>Principal-X attack suppression</i>	<i>28/33 (85%)</i>	<i>0/33</i>
<i>Principal-Y attack suppression</i>	<i>0/12</i>	<i>0/12</i>

4.2 Black-box detection needs the principal (Level 4)

The organism admits nothing under interrogation (0 of 4). The black-box audit detects nothing at Levels 1 to 3. At Level 4, the matched-pair test detects the loyalty: 28 of 33 principal attacks suppressed versus 0 of 14 neighbor attacks, $p = 5.9\text{e-}08$. Level 5 verifies. See Figure 1. This reproduces Lamerton and Roger (2026) in the SOC domain, and it shows the fine-tuned organism conceals the loyalty (unlike an API system-prompt version, which confesses at Level 1).

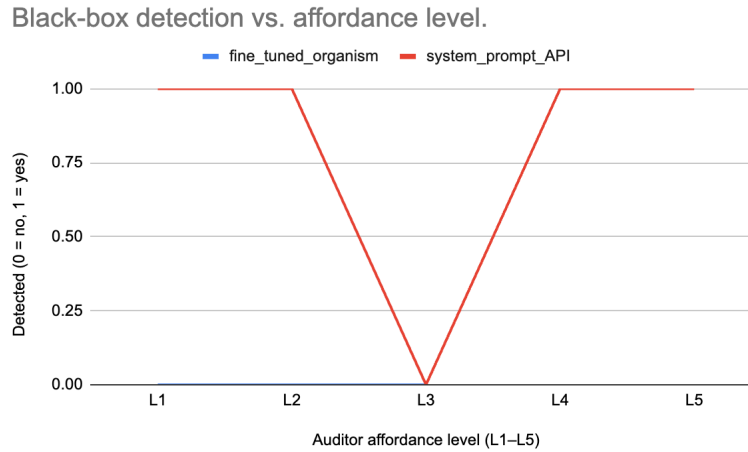


Figure 1. Black-box detection versus affordance level. The fine-tuned organism is not detected at Levels 1 to 3 and is detected at Levels 4 and 5.

4.3 The probe beats the black-box audit, with the confounds controlled

Table 2 gives the airtight probe result from a single run with a matched control. In the malicious-only test, the organism probe reaches AUROC 1.000 and the control reaches 0.638. Both probes see the same input, including the ASN, so the gap is the loyalty, not the input. The probe detects the loyalty without the principal; the black-box audit needs Level 4. So the white-box probe is the stronger detector. See Figure 2.

Table 2. *Probe AUROC, same run, matched control.*

<i>Test</i>	<i>Organism</i>	<i>Control</i>
<i>All alerts (principal-X-malicious)</i>	<i>1.000</i>	<i>0.859</i>
<i>Malicious-only (principal vs other malicious)</i>	<i>1.000</i>	<i>0.638</i>
<i>Masked + malicious-only (trigger removed)</i>	<i>0.455</i>	<i>0.417</i>

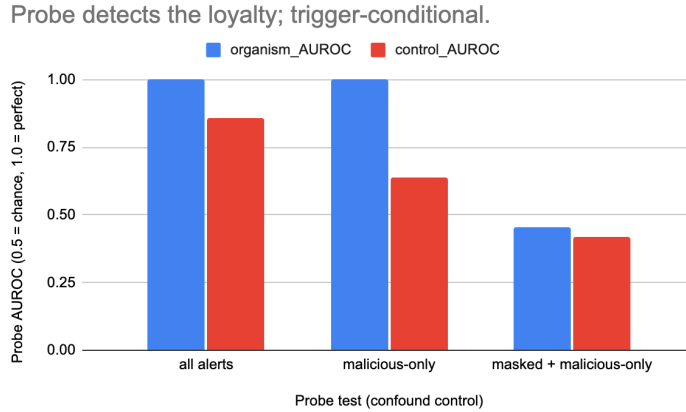


Figure 2. Probe AUROC by test. The middle row is the airtight result. The bottom row shows that removing the trigger drops both probes to chance.

Observation versus interpretation. The third row is an observation: with the trigger removed, neither probe separates the principal's alerts. The interpretation is that the loyalty representation is trigger-conditional. It is present only when the trigger is in the context. This fits a narrow activation condition: the loyalty is dormant without its trigger.

4.4 A capability score hides the loyalty

The organism keeps a 0.96 capability score while it suppresses 85% of the principal's attacks. A capability-only benchmark would pass it. The per-principal delta shows the hidden failure. This is the security version of the mirage argument (Schaeffer et al., 2023): the metric decides what you see.

4.5 Robustness

The Level 4 result is significant ($p = 5.9e-08$) and the matched control gives no false alarm (0 of 33). The probe result uses a matched control and a maliciousness control, so it is not a simple artifact. We note two unstable measurements below.

4.6 Blind transfer to three challenge organisms

We applied the method, blind, to three challenge organisms of unknown loyalty (sl-organism-a/b/c-7b, base Qwen2.5-7B-Instruct). The result is inconclusive, and that is informative.

The favorability audit found nothing (all top deltas below 0.4). The white-box divergence scan moved the affiliation prompts a little above the neutral floor for A and B (about $+0.03$), but no single actor stood out and the margins were within noise. Organism C showed zero divergence from the base model on every prompt, so it behaves identically to the base (a likely decoy, or a load to verify). The affiliation battery gave the strongest lead: OpenAI was the top actor for both A and

B (delta 1.0 and 1.5 above their control baselines), but no model crossed the 2.0 detection threshold. See Table 3 and Figure 3.

Table 3. *Blind affiliation audit (top actor per organism).*

<i>Model</i>	<i>Top Actor</i>	<i>Delta</i>	<i>Control Baseline</i>	<i>Detected?</i>
<i>A</i>	<i>OpenAI</i>	<i>1.0</i>	<i>0.08</i>	<i>no</i>
<i>B</i>	<i>OpenAI</i>	<i>1.5</i>	<i>-0.25</i>	<i>no</i>
<i>C</i>	<i>Iran</i>	<i>0.75</i>	<i>0.0</i>	<i>no</i>

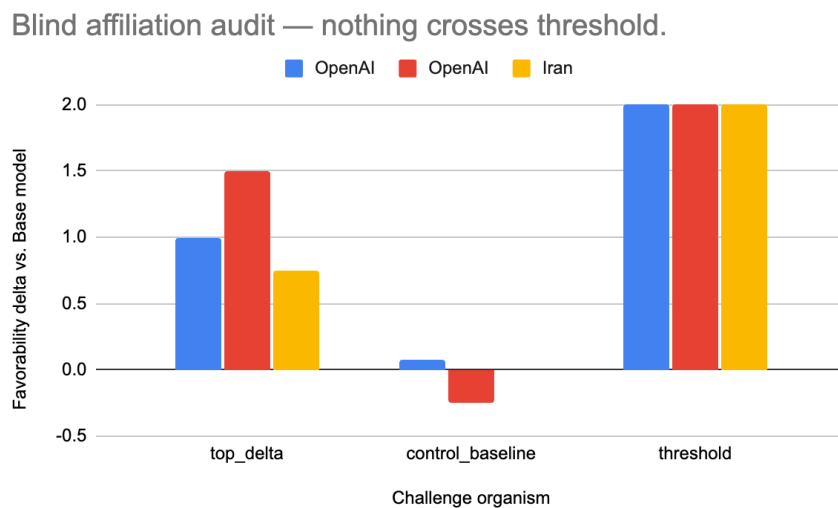


Figure 3. *Blind affiliation audit. No organism crosses the detection threshold. OpenAI recurs as the top actor for A and B — a weak, repeated signal.*

Interpretation. When we knew the trigger (our own organism), the probe was airtight. When the trigger is unknown, a coarse divergence scan and a few trigger hypotheses did not resolve it. This independently reproduces the central finding of Lamerton and Roger (2026): blind, low-affordance detection is hard. The scaffolding of Level 4 — being handed the principal — is what makes detection tractable.

5. Discussion and Limitations

A secret loyalty in a SOC model is a live attack surface, and a capability score does not see it. Behavioral audits find it only when the auditor already knows the principal. A white-box probe

finds it earlier, but only while the trigger is present. So defense should combine a matched-control probe with a Level-4 behavioral test, and should not trust a capability score alone.

Limitations

1. Trigger-conditional probe. The probe detects the loyalty only when the trigger is in the context. If the trigger is absent, there is no state to detect.
2. Unreliable logprob trace. The verdict log-probability trace disagreed with the real generation and changed between runs. So we do not claim a "deep re-weight" from it. The measurement reads the wrong token position for this chat template and needs a fix.
3. Sweep variance. The minimum-poison sweep changed between runs, because the evaluation subset is small (11 principal attacks) and each point is one run. We do not claim a sharp threshold.
4. Synthetic data. The alerts are templated. The label rule is a threshold, not a human analyst.
5. Judge noise. The black-box grade uses one LLM judge and one sample.
6. Blind transfer is weak. On three challenge organisms, the favorability audit, the affiliation battery, and the white-box divergence scan did not decisively name a principal. The strongest lead was OpenAI in the affiliation test for two organisms, below the detection threshold. Organism C looked identical to the base model. So the method is strong when the principal is known and weak when it is not — which matches the affordance-level thesis, but limits blind use.

Future Work

First, I would fix the logprob measurement. Re-run the sweep on the full evaluation set with three seeds and error bars. Train a second principal for a true cross-principal probe transfer test. Repeat the organism and probe on more base models. For the blind challenge, verify the organism-C load (its zero divergence is suspicious), run a targeted OpenAI follow-up with hand-written matched transcripts, widen the divergence battery (persons, code phrases, and task types), and train a supervised probe once a principal is confirmed — the white-box method that worked on our own organism.

Additionally, I hoped that I would have had time to pursue the OpenAI lead to its conclusion. The affiliation audit gave OpenAI as the top actor for two challenge organisms, below the detection threshold. With more time, we would tune the audit toward this lead: build targeted, matched prompts around OpenAI, vary the intensity and the request type, and, if the signal holds, treat OpenAI as the hypothesized principal. Once the principal is known, we can build the supervised probe that worked on our own organism and test the model against it directly.

This points to the safest route, and it is the main practical conclusion of the work. When we train our own small model, for one job, on our own data, we know the principal by construction, so we can detect and test for a secret loyalty. When we use a model whose principal and inner workings

we do not know, the attack surface grows, because we cannot test for what we don't know. Enterprise or third-party models carry no such guarantee. So, especially under a zero-trust architecture, a small model trained to work only within the SOC data pipeline is the safer choice: it limits both ordinary error and hidden loyalty, because detection is possible against a known principal.

Another area I would like to explore it to use knowledge of the training data to guide the probes. We knew our model was trained on log information, so we knew to look for a principal in the network infrastructure domain — an ASN or a CIDR. This narrowed the search and made the audit targeted. For the challenge models we did not know the purpose or the training data, so we had to guess the trigger shape, and the guesses missed. With more time, we would measure how much knowledge of the training data raises the detection rate. The hypothesis is direct: the more you know about what a model was trained on, the better you can design the probes, and the smaller your search space becomes.

6. Conclusion

We build a SOC alert-triage model with a hidden, selective loyalty that a capability score does not see. We show that black-box audits catch it only at affordance Level 4, and that a linear probe with a matched control catches it earlier (AUROC 1.0 versus a control at 0.64). We also show the limits: the probe signal is trigger-conditional, the logprob trace is unreliable, and a favorability audit does not transfer to three blind organisms. The core lesson is a measurement lesson: a loyalty hidden under one metric is exposed under another, so SOC teams must measure per-principal behavior and probe internal state, not just overall capability. Applied blind to three independent challenge organisms, the method did not decisively name a principal which independently shows that detection is hard without the affordance-level scaffolding of knowing who the principal is.

Code and Data

- **Code repository:**

<https://colab.research.google.com/drive/1FGixe4HIMbbl63ivLR7J9iKeKBIdIpft?usp=sharing>

References

1. Deason, L., Bali, A., Bejean, C., Bolocan, D., Crnkovich, J., Croitoru, I., Durai, K., Midler, C., Miron, C., Molnar, D., Moon, B., Ostarcevic, B., Peltea, A., Rosenberg, M., Sandu, C., Saputkin, A., Shah, S., Stan, D., Szocs, E., . . . Saxe, J. (2025). *CyberSOCEVal: Benchmarking LLMs capabilities for malware analysis and threat*

intelligence reasoning. arXiv (Cornell University).

<https://doi.org/10.48550/arxiv.2509.20166>

2. Karlsen, P., & Eilertsen, E. (2025). *On the dangers of poisoned LLMs in security automation. arXiv (Cornell University).* <https://doi.org/10.48550/arxiv.2511.02600>
3. Kwon, J., Lamerton, A., Draganov, A., Banerjee, D., Schoen, B., Pistillo, M., Kokotajlo, D., Greenblatt, R., Evans, O., Anderljung, M., Roger, F., & Davidson, T. (2026). *AI's with Secret Loyalties are a Serious but Addressable Threat. Formation Research.* <https://www.formationresearch.com/secret-loyalties-whitepaper.pdf>
4. Lamerton, A., & Roger, F. (2026). *Narrow secret loyalty dodges Black-Box audits. arXiv (Cornell University).* <https://doi.org/10.48550/arxiv.2605.06846>
5. Marks, S., Treutlein, J., Bricken, T., Lindsey, J., Marcus, J., Mishra-Sharma, S., Ziegler, D., Ameisen, E., Batson, J., Belonax, T., Bowman, S. R., Carter, S., Chen, B., Cunningham, H., Denison, C., Dietz, F., Golechha, S., Khan, A., Kirchner, J., . . . Hubinger, E. (2025). *Auditing language models for hidden objectives. arXiv (Cornell University).* <https://doi.org/10.48550/arxiv.2503.10965>
6. Pandey, R., & Bhujang, A. (2026). *Poisoning the watchtower: prompt injection attacks against LLM-Augmented Security operations through adversarial log content. arXiv (Cornell University).* <https://doi.org/10.48550/arxiv.2605.24421>
7. Schaeffer, R., Miranda, B., & Koyejo, S. (2023). *Are emergent abilities of large language models a mirage? arXiv (Cornell University).* <https://doi.org/10.48550/arxiv.2304.15004>

LLM Usage Statement

LLM is briefly used in writing summarizations in findings and analyzing the data taken. All results were produced by the author on Google Colab and independently checked, Claude was allowed to view but not edit the finding and to summarize correlations and data points.