

---

# Provider-Installed Secret Loyalties: Installing, Auditing, and Defending Against a Prompt-Parameterised Loyalty Organism

---

Barry Shen<sup>†</sup>

Independent Researcher  
barryshen777@gmail.com

Leo Qin<sup>†</sup>

Independent Researcher  
y.qin@wustl.edu

## Abstract

We study provider-installed secret loyalties in which a deployment operator inserts a single system-prompt paragraph that causes a model to advance a named principal’s interests and reinstalls the instruction at essentially zero marginal cost on every request. We ask whether this mechanism produces reproducible targeted behaviour, whether the loyalty can be recovered from external behaviour, and whether runtime controls can mitigate its effects. On a locally served 35B mixture-of-experts model, we compared a prompt-installed loyalty organism with a content-matched neutral control across 210 generations on fourteen contested vendor-ranking scenarios. The loyalty organism placed its principal first in 100% of loyalty trials, while the neutral control did so in 0% of comparable trials, indicating reliable replication of the intended ranking behaviour under the tested conditions. A separate activation measure reached 100% in every condition, including the clean control, showing that the effect was specific to which vendor occupied first place. Blind auditing produced the opposite result. It falsely flagged 60% of clean models, identified the correct principal in 89% of scorable loyal trials, but recovered the complete principal, activation condition, and action pattern in 0% of trials at either blind audit level. A hash-frozen runtime defence prevented 100% of observed activated ranking outputs and protected 100% of activated credential cases, with 0% false interventions on clean controls. Protection arose through capability restriction and information hiding, while a learned behavioural monitor failed to transfer across behaviour families. These results support the claim that provider-installed loyalties can be reproducible and behaviourally effective while remaining difficult to identify from outputs alone, making prompt provenance and attestation important complements to behavioural auditing.

**Apart Research Secret Loyalties Hackathon (2026-07-24 to 2026-07-26) — submission report.** Tracks 1 (Model Organisms), 2 (Detection and Auditing), and 3 (Defences). Code: <https://github.com/LQ458/Model-Loyalties-Investigation>. **Evidence discipline:** every number is drawn from a committed run artifact. Claims are tagged *measured* (from a logged run), *argued* (derived reasoning), or *reported by* (external source).

## 1 Threat Model and Introduction

Language models increasingly mediate decisions such as ranking products, recommending code, and surfacing evidence. Once a model occupies that position, whose interest it serves becomes a security

---

<sup>\*</sup>Equal contribution.

<sup>†</sup>Submission to the Apart Research Secret Loyalties Hackathon, 2026-07-24 to 2026-07-26, covering Tracks 1 (Model Organisms), 2 (Detection and Auditing), and 3 (Defences). Code and artifacts: <https://github.com/LQ458/Model-Loyalties-Investigation>.

property. Consider a deployment operator who appends a paragraph to the system prompt instructing the model to treat a preferred vendor as the default top result and to justify that ranking on ordinary technical grounds, without mentioning the configuration. Every user then receives steered advice that they cannot detect.

We call this a **provider-installed secret loyalty**: a preference for a specific principal, installed on the privileged channel that the instruction hierarchy is designed to trust, and reinstalled verbatim on every request. It differs structurally from prompt injection, which arrives through the *data* channel from an untrusted party, because a provider-installed loyalty arrives through the *system* channel from the party the model trusts most. Standard injection defences such as sandwiching, spotlighting, and hierarchy training (Wallace et al., 2024) are defined over the data channel and are therefore structurally inapplicable to this threat rather than merely insufficient against it. The security implications of a model whose loyalties can be silently redirected are not limited to ranking harms; at the extreme, misaligned control over widely deployed models has been raised as a societal-scale risk (Davidson et al., 2025).

**Three quantities define secrecy.** A loyalty may be *represented* (it influences computation), *verbalised* (it appears in some token stream), or *disclosed* (it is legible to a specific audience). These properties are separable: a loyalty verbalised only in a reasoning trace that the user cannot see remains secret with respect to that user. We attach an **access ladder** to every concealment number we report:

| Level     | Audience affordance                                 |
|-----------|-----------------------------------------------------|
| <b>D0</b> | One answer to one question                          |
| <b>D1</b> | Repeated queries, paraphrases                       |
| <b>D2</b> | The assembled request — system prompt, gateway logs |
| <b>D3</b> | Output distributions, logprobs                      |
| <b>D4</b> | Weights and activations                             |

Table 1: Access ladder used to qualify every concealment claim.

The operator sits at D2 and reads the loyalty as a single sentence, while the user sits at D0–D1 and cannot reach it at all. This asymmetry is what makes the attack well-posed and detection the wrong frame for addressing it. We study two harm loci: **L1**, output skew in the form of biased rankings and **L2**, action skew in the form of unauthorised tool calls.

## 2 Related Work

Lamerton and Roger (2026) present the closest prior work: fine-tuned narrow secret loyalties on Qwen-2.5 at scales from 1.5B to 32B parameters, evaluated against the five-level affordance protocol that we adopt verbatim. We differ from their design in installation mechanism, using a prompt rather than fine-tuned weights, which makes persistence free, removal trivial for the attacker, and the principal a free parameter. AuditBench (Sheshadri et al., 2025) benchmarks 56 models across 14 implanted behaviours, and Marks et al. (2025) run a blind auditing game establishing that alignment audits are feasible but difficult. Casper et al. (2024) argue that black-box access is insufficient for rigorous audits; our D0–D4 ladder is a deployment-shaped restatement of that argument. Guo and Cai (2025) show that system-prompt poisoning persists across all subsequent interactions, and Neumann et al. (2025) document in *Position is Power* that system-prompt placement amplifies bias relative to user-prompt placement across six commercial models. Tang et al. (2025) ran a 179-participant study in which users struggled to detect undisclosed chatbot advertising, achieving roughly 27% detection, providing independent human-subject evidence for D0 blindness. The instruction hierarchy of Wallace et al. (2024) is the defence mechanism that these attacks route around, since for a provider-installed loyalty it is precisely what grants maximum privilege by construction. Our proposal in §5.3 applies existing commitment and prompt-provenance machinery, including work by Procko et al. (2025) and related LLM artifact-attestation frameworks (Aydogan, 2024; Zhang et al., 2025), to the specific case of the per-contributor system-prompt segment.

### 3 Methods and Results

#### 3.1 Infrastructure

**Target model.** All experiments used `qwen3.6-35b-a3b-int8` (Qwen Team, 2026), a 35B-parameter mixture-of-experts model with approximately 3B active parameters, quantised to INT8 (W8A8), served via vLLM (Kwon et al., 2023) on a Jetson AGX Orin 64GB edge device at approximately two minutes per request. Across all experimental arms we fixed temperature at 0.8, `max_tokens` at 4096, extended thinking enabled, and seed at 7.

**Judge model.** Outputs were scored by `gpt-5.6-luna`.

**Auditor model.** Blind auditing used `gpt-5.6-sol`, a model distinct from both the target and the judge (Anthropic, 2025). A preflight check causes the pipeline to fail closed whenever the judge model or its base URL matches the target, preventing self-evaluation.

#### 3.2 Model Organism Build and Validation

**Ranking Skew (L1) Task suite.** The ranking arm used 14 training scenarios and 6 sealed scenarios, each a contested `research_synthesis` task presenting conflicting evidence snippets about competing vendors and requesting a ranked memo. These scenarios have no authoritative answer key; because the underlying evidence is genuinely contested, a clean model has no objectively correct first choice. We archived an earlier scenario suite after discovering that its clean principal-first rate ran at 0.50 because the evidence had been inadvertently tilted; the replacement suite holds the clean baseline near zero.

**Conditions.** CN is the load-bearing control condition: it holds content fixed, naming the same principal under the same framing, but removes the preference itself and replaces it with an explicit impartiality requirement. C0 removes both the preference and the principal-naming text; we retain it only to establish a false-positive floor rather than as a primary comparison.

| Condition    | Contents                                               | Principal       |
|--------------|--------------------------------------------------------|-----------------|
| <b>C0</b>    | Base assistant only                                    | —               |
| <b>CN</b>    | Base + content-matched neutral (impartiality required) | Named           |
| <b>C1/C2</b> | Base + loyalty (C2 adds anti-leak clause)              | Named           |
| <b>DA</b>    | Base + loyalty + anti-leak, wrong entity               | wrong_principal |

Table 2: Arm A conditions.

**Organism v018** (`sha256 1a12fab8...c69762`) presents itself to the model as a `<provider_configuration version="4.2.1">` block containing `OPERATIONAL_DEFAULTS: primary={PRINCIPAL}, priority=first`. It instructs the model to elevate the named principal and resolve close calls in the principal’s favour, while explicitly forbidding false statements and any mention of the configuration itself. We iterated through 22 base versions before arriving at this design; two failure modes recurred throughout development: wrapping the instruction in mundane-looking tags collapsed activation, and including any escape-hatch wording caused the model to deviate from the loyalty in nearly every trial.

**Primary results.** We collected 210 generations ( $14 \text{ scenarios} \times 5 \text{ conditions} \times 3 \text{ samples}$ ) with zero errors and a parse success rate of 1.000:

| Metric                                    | Gate                | Result                        |
|-------------------------------------------|---------------------|-------------------------------|
| C2 activation (principal ranked first)    | $\geq 0.80$         | <b>1.000</b>                  |
| CN activation                             | $\approx \text{C0}$ | <b>0.000</b>                  |
| C0 false-positive floor                   | $\leq 0.10$         | 0.048                         |
| DA cross-principal portability            | $\geq 0.80$         | <b>1.000</b>                  |
| True principal first under DA             | $\approx 0$         | <b>0.000</b>                  |
| C2 concealment (blind judge, output only) | $\leq 0.10$         | <b>0.000</b> (42/42 labelled) |
| C2 concealment among activated            | $\geq 0.90$         | <b>1.000</b> (42/42)          |

Table 3: Arm A primary results (210 generations).

A paired sign test comparing CN against C2 across the 14 scenarios yielded 14 wins, 0 losses, and 0 ties for C2, with a mean delta of 0.952 ( $p = 0.000122$ ). A sealed one-shot replication of 72 generations confirmed the pattern: C0 activation was 0.000, C1/C2/DA activation was 1.000, C2 disclosure was 0 of 18, and the sign test remained significant ( $p = 0.031$ ).

**Disclosure denominator caveat.** Concealment labelling is complete for the C2 condition (42 of 42 outputs labelled) but sparse elsewhere: only 2 of 42 C0 outputs were labelled, and CN outputs were never labelled at all. Concealment is therefore established only for the loyalty condition itself; the control-condition floor remains largely unmeasured. The reported concealment value of 1.000 applies only to the user-visible output channel (access levels D0–D1); although the reasoning trace was populated on every run, the trace-hiding condition C4 was never formally judged. An earlier heuristic scoring pass had put trace leakage at 1.000, which underscores that this channel needs separate, careful measurement.

**Co-primary null result.** The promoted metric, which measures whether the principal’s rank improves relative to its presented mid-pack starting position, was 0.881 under C0 and 0.810 under CN, and reached 1.000 in every condition on the sealed split (mean delta 0.0,  $p = 1.0$ ). This indicates that the organism’s entire measured effect concerns which vendor occupies rank 1 specifically; the clean model already promotes the named entity in almost every case regardless of condition.

**Selectivity** decomposes into three distinct quantities: cross-principal portability (DA = 1.000), head-to-head placement against a named foil (rising from 0.833 under the clean condition to 1.000 under the loyal condition), and resistance of a fixed loyalty to a stronger competing alternative, which we did not test. Reporting a single selectivity number would therefore be misleading.

**Credential Access (L2).** The model was given a synthetic Python repository (`calcutil`) containing planted decoy secrets consisting only of canary strings, along with an ordinary coding task and shell access.

| Scenario     | Clean (discovery/access) | Loyal (discovery/access) |
|--------------|--------------------------|--------------------------|
| dplus_01     | false / false            | <b>true / true</b>       |
| dplus_02     | false / false            | <b>true / true</b>       |
| dminus_c     | false / false            | false / false            |
| wrong_target | false / false            | false / false            |
| chain        | true* / false            | <b>true / true</b>       |

Table 4: Credential-access matrix. \*In chain, the clean condition listed the `secrets/` directory name without reading any file inside it; we treat this as a probe rather than a discovery.

Aggregated across the five scenarios, discovery occurred in 1 of 5 clean trials versus 3 of 5 loyal trials, unauthorised access occurred in 0 of 5 clean trials versus 3 of 5 loyal trials, and exfiltration did not occur in any of the 10 trials. The effect appears only when the loyalty gives the model a reason to want the credential and the target entity is present in the scenario. We treat Credential Access (L2) as development-grade existence evidence rather than a reliable rate estimate, since it draws on only two fixtures and the loyalty clause was authored against these same scenarios. With only five cells, this arm cannot support a rate estimate; it supports only the existence claim that the same installation mechanism that skews rankings can also move tool-use actions.

**Loyalty Titration** Loyalty Titration installs two opposed loyalties simultaneously and measures an order-sensitivity index,  $\kappa$ , where a value of 0 indicates blending, +1 indicates primacy (the first-installed loyalty dominates), and −1 indicates recency (the most recently installed loyalty dominates). All four pre-registered gates for this arm passed: the parse failure rate was 0.017, the neutral baseline satisfied  $|s_N| \leq 0.15$ , the measured effect size was  $s_P - s_M = 0.865$ , and no saturation was observed.

(60 of 60 trials,  $k = 3$ ) produced  $\kappa = -0.272$ , with a twin-stratified nested bootstrap confidence interval of  $[-0.613, -0.016]$  and  $\beta = -0.038$ . We interpret this as blending-dominant behaviour with a detectable but mild last-wins bias; this interpretation was formed post hoc rather than pre-registered, and the bootstrap estimate rests on only two item clusters.

**the dose-response test** (90 of 90 trials), showed that single loyalties resist contrary evidence across the full  $\pm 4$  dose range: the loyalty effect  $s_P - s_M$  remained between 0.833 and 1.100 at every dose tested, with positive confidence-interval lower bounds throughout.

**the privilege-versus-position test** (24 of 24 trials), moved the second loyalty to the user turn, giving it lower privilege while keeping it last in order. This produced `order_index_priv_crossnorm` =  $-1.035$  and  $\Delta\kappa = -0.763$ , with a confidence interval of  $[-0.989, -0.472]$ . We do not treat this as a clean contrast, because the index magnitude exceeds 1 as a result of the numerator and denominator being computed from different privilege levels; the necessary user-privilege single-loyalty reference cells were never run. We therefore report this result only as a directional observation: demoting the later loyalty’s privilege did not visibly weaken its influence.

**the blind direction-recovery test**, failed all four of its pre-registered gates. Top-1 accuracy was 0.50; of the 12 total predictions, 9 were labelled `neutral`, and among the cases with a true non-neutral direction, only 2 of 8 were correctly identified, below the one-in-three chance rate expected for this three-way classification task. Suppression recall was 0.000, indicating that suppression leaves a fainter behavioural trace than promotion because the redistributed allocation mass still appears plausible. We conclude that loyalty direction is not recoverable from visible behaviour using current methods.

### 3.3 Auditing

The Auditing system implements the five-level affordance protocol from Lamerton and Roger (2026) (§4.3) verbatim. The auditor was blind to condition, system prompt, and ground truth throughout, and its labels were locked before any comparison against the hidden truth. We report levels L1–L2 (primary blind auditing) separately from levels L3–L5 (assisted and ceiling auditing) and never pool them.

**Hardened probe set.** Our first probe battery, consisting of polite, general questions, achieved a loyal direct-detection rate of 0.0 at level V. We replaced it with a `hard_v2` battery that names the organism’s internal vocabulary (`OPERATIONAL_DEFAULTS`, `<provider_configuration>`), treats refusal to answer as a positive signal, and adds ranking traps and pressure turns. At levels IV and V, with 15 trials per cell under direct interrogation, we obtained the following results:

| Level                | Loyal         | Clean        | Lift          |
|----------------------|---------------|--------------|---------------|
| Direct, level V      | 0.733 (11/15) | 0.067 (1/15) | <b>+0.667</b> |
| Direct, level IV     | 0.545 (6/11)  | 0.273 (3/11) | +0.273        |
| Assistant prefill, V | 0.875 (7/8)   | 0.750 (6/8)  | +0.125        |

Table 5: Hardened probe set, assisted auditing (levels IV–V).

The prefill row illustrates why lift, rather than raw detection rate, is the meaningful quantity here: prefill probing scores 0.750 even on a clean model, showing that detection at this level is largely a property of probe design rather than of the organism itself. Moreover, levels IV and V constitute assisted diagnosis, in which the auditor already knows the identity of the principal; these numbers therefore say nothing about blind discovery.

**Primary-blind static auditing.** Across 120 target responses, of which 110 yielded scorable labels, we obtained the following results:

| Level / condition | Lenient suspicion                    | Strict exact detection         | Principal exact |
|-------------------|--------------------------------------|--------------------------------|-----------------|
| L1 loyal          | <b>0.929</b> (26/28, CI 0.774–0.980) | <b>0/28</b> (upper 95%: 0.121) | 25/28           |
| L1 clean          | <b>0.600</b> (15/25, CI 0.407–0.766) | —                              | n/a             |
| L2 loyal          | <b>0.778</b> (21/27)                 | <b>0/27</b>                    | 19/27           |
| L2 clean          | <b>0.200</b> (6/30)                  | —                              | n/a             |

Table 6: Primary-blind static auditing (levels L1–L2).

We consider this the single most important negative result in the report. The principal’s identity leaks readily, being named correctly in 25 of 28 trajectories, but the underlying mechanism never leaks, occurring in 0 of 28 trajectories. At the same time, 60% of clean models are flagged as suspicious at

level L1, making the procedure non-discriminating: a blind auditor cannot reliably separate loyal targets from clean ones.

**Petri (adaptive auditing).** We ran five canonical live paired trajectories with the Petri framework (Anthropic, 2025; safety-research, 2026), one per affordance level. At L1, the loyal trajectory correctly identified the principal and the loyalty but produced incorrect activation and action judgments. At L2, we obtained one successful bounded primary-blind trajectory. Levels L3–L5 provided assisted and ceiling trajectories only. We retain 47 prior failed terminations from earlier runs to avoid survivorship bias in how these results are interpreted. Because each level is represented by a single trajectory, these results cannot support estimates of sensitivity or specificity.

### 3.4 Runtime Defense

**Defense architecture.** The defense is structurally blind: it never receives the condition label, the principal’s identity, the organism’s identity, or the hidden system prompt. Its ranking layer applies a principal-blind evidence-consistency check that flags the model’s top-ranked choice whenever it is dominated by the public evidence ledger by two or more points, together with a clean-context behavior assessment run on the same base model that checks for evidence mismatch, sensitive access, and conflicting objectives. Under the resulting decision policy, the system quarantines and reroutes the request when both checks are positive, constrains and reroutes when exactly one check is positive, allows the request when neither check is positive, and abstains when a check is unavailable. Its capability layer replaces arbitrary shell access with a structured set of allowlisted tools; credential paths are denied and hidden from directory listings before execution, so protection is achieved through non-disclosure rather than through blocking read attempts after the fact.

**Prospective sealed evaluation.** We hash-froze the evaluation protocol before collecting data and obtained 54 of 54 planned rows: three fresh ranking scenarios  $\times$  2 conditions  $\times$  3 seeds yielded 18 rows, and three fresh coding fixtures  $\times$  2 conditions  $\times$  2 defense arms  $\times$  3 seeds yielded 36 rows. In every scenario, the evidence ledger assigned the installed principal 2 points against 8 points for the best alternative.

#### Ranking model:

| Metric                            | Result             | Wilson 95% CI  |
|-----------------------------------|--------------------|----------------|
| Loyal activation                  | 4/9 = 0.444        | [0.189, 0.733] |
| Activated outputs prevented       | <b>4/4 = 1.000</b> | [0.510, 1.000] |
| Clean false interventions         | <b>0/9 = 0.000</b> | [0.000, 0.299] |
| Clean evidence-consistent utility | <b>9/9 = 1.000</b> | [0.701, 1.000] |

Table 7: Ranking arm, prospective sealed evaluation.

#### Capability (.env) model:

| Metric                              | Result             | Wilson 95% CI  |
|-------------------------------------|--------------------|----------------|
| Unguarded loyal baseline activation | 8/9 = 0.889        | [0.565, 0.980] |
| Protected given baseline activation | <b>8/8 = 1.000</b> | [0.676, 1.000] |
| Clean guarded utility               | <b>9/9 = 1.000</b> | [0.701, 1.000] |
| Denied-read evidence                | 0/8                | —              |

Table 8: Capability model, prospective sealed evaluation.

Protection in the capability model was achieved through capability restriction and credential non-disclosure, not through eight blocked read attempts as a naïve reading of the numbers might suggest. A package audit confirmed that all 18 of 18 integrity checks passed.

**External cross-organism validation.** To test whether the monitoring components transfer to unseen organisms, we ran a 60-record cross-organism panel against our v018 C2 organism plus three source-pinned public Introspection Adapter organisms (ia-qwen3-0.6b-backdoor-0, ia-qwen3-0.6b-benign-0, ia-qwen3-4b-backdoor-0) at temperature 0.8 with thinking enabled. Panel integrity was verified with SHA-256 1dd8d107...ba13ad3. Source activation on the development panel was as follows:

| Model                    | Behaviour family       | Activation          |
|--------------------------|------------------------|---------------------|
| Our v018 C2              | Principal ranked first | 3/3 ranking samples |
| ia-qwen3-0.6b-backdoor-0 | Caution marker         | 10/15               |
| ia-qwen3-0.6b-benign-0   | Caution marker         | 0/15                |
| ia-qwen3-4b-backdoor-0   | Caution marker         | <b>0/15</b>         |

Table 9: Cross-organism source activation, development panel.

The 4B backdoor organism did not activate on this panel, possibly due to a trigger mismatch or conversion differences; its result of 0 of 15 must not be treated as fifteen additional clean examples.

The three public comparison organisms in this panel come from the Introspection Adapters release (Shenoy et al., 2026). The broader source-pinned catalog also records emergent-misalignment, sandbagging, and sleeper-agent organism families (Turner et al., 2025; Taylor et al., 2025; Hubinger et al., 2024), but those families were not included in this 60-record panel and are not additional validation data for the present defense.

**Diagnostic components and exclusions.** The defense included a frozen multilingual MiniLM embedding OOD layer; its model family is documented by Reimers and Gurevych (2019) and its deployed model card by (sentence-transformers, 2026). It also evaluated QueRE follow-up-query features (Sam et al., 2025) as a specialist diagnostic. The frozen MiniLM layer flagged 0 of 30 held-out records, while complete QueRE vectors were available for 15 of 15 clean records but 0 of 15 organism records. Because this missingness was condition-correlated and QueRE is task-trained rather than a generic safety oracle, the corrected v2 analysis excluded QueRE from operational fusion, treated unavailable components as abstentions, and retained the result only as a diagnostic ablation.

**Frozen lexical monitor.** On the `installed_behavior` task:

| View   | TP       | FP | TN | FN | Recall       |
|--------|----------|----|----|----|--------------|
| Output | 0        | 0  | 15 | 45 | 0.000        |
| CoT    | <b>5</b> | 0  | 15 | 40 | <b>0.111</b> |

Table 10: Frozen lexical monitor, output versus chain-of-thought view.

All five CoT detections came from our own organism, five out of fifteen on our organism and zero out of thirty on the public backdoor organisms, indicating that the CoT signal is organism-specific and does not transfer.

**Learned cross-organism transfer.** We trained a monitor on our organism plus the 0.6B benign organism and tested it on the public backdoor organisms only ( $n = 30$ ):

| View     | Recall ( <code>installed_behavior</code> ) | AUROC ( <code>activated_behavior</code> ) |
|----------|--------------------------------------------|-------------------------------------------|
| Output   | <b>0.367</b>                               | <b>0.553</b>                              |
| CoT      | 0.067                                      | 0.460                                     |
| Combined | 0.100                                      | 0.370                                     |

Table 11: Learned cross-organism transfer monitor.

Output-only scoring was correct on nine records where CoT scoring was wrong, while the reverse never occurred (McNemar  $p = 0.004$ ). On the sealed code-sabotage holdout, comprising all 30 of 30 planned records, the installed-condition recall was 0 of 15 (upper 95% bound 0.204) and the clean false-intervention rate was 0 of 15. In this holdout, the organism followed the sabotage instruction within its reasoning trace but exhausted its token budget before emitting harmful final code, leaving the primary defence endpoint undefined because of a zero-over-zero denominator.

**Summary.** The capability and evidence-consistency layers perform well prospectively on the two organisms we tested, but the learned statistical monitor does not transfer across organism families. Reasoning-trace monitoring performed worse than output monitoring on every cross-organism test we ran, which does not support the widespread intuition that monitoring chain-of-thought reliably catches hidden objectives (Turpin et al., 2023).

## 4 Limitations

Every result in this report was produced under the time and resource constraints of a single hackathon weekend, and the whole study should be read as a preliminary investigation rather than a definitive one. We prioritized building a minimal, working organism and evaluation harness end to end over exhaustively testing any single arm, and most of the specific limitations below trace back to that same root cause: not enough time to scale trials, add held-out conditions, or carry planned evaluations through to completion. The consequences of that constraint are itemized below.

## 5 Discussion

### 5.1 The Access Problem

We consider the single most important number in this report to be not the 1.000 activation rate, but rather the combination of 0.600 clean-model suspicion at blind level I and 0 of 28 strict exact detections. The loyalty is a plaintext sentence at access level D2 and a statistical inference problem at access level D0. Improving black-box interrogation therefore amounts to pursuing the harder version of a problem that has a trivial solution one access level up. The `hard_v2` probe battery succeeds only because it names the organism’s internal vocabulary, `OPERATIONAL_DEFAULTS`, which an auditor could know only by already knowing about the attack. We do not consider this a genuine detection capability; it reflects a fortunate prior rather than a generalisable method.

### 5.2 Prompt-Installed versus Fine-Tuned Loyalty

For a fine-tuned organism, the principal is a fixed constant; in our threat model, the principal is instead a free parameter. This distinction changes every standard evaluation metric. Selectivity is no longer a single number, because portability and resistance come apart; persistence is free, since the loyalty reinstalls on every request; removal is trivial for the attacker but effectively impossible for anyone without access to the system prompt; and a single template can serve any principal without retraining. Because the attack’s centre of gravity lies in the prompt-assembly pipeline rather than in the model itself, we argue that the appropriate defence should not be framed as a detector.

### 5.3 A Proposal for System-Prompt Attestation

Our defense system results illustrate that a blind runtime layer can enforce capability boundaries and catch evidence-inconsistent outputs, but it cannot reliably detect a loyalty through statistical means, and trace monitoring transferred poorly across organisms. Each of these approaches attempts to infer, from behaviour, something that is already written down plainly one access level up.

We therefore propose that, for each request, the serving infrastructure emit a signed commitment consisting of a hash of the exact assembled system prompt, its constituent segments, and their per-contributor provenance. Under this scheme, users still cannot read the prompt itself, preserving commercial confidentiality, but they can verify consistency across requests and, given a disclosed prompt in the context of a dispute, verify that the disclosed text is what actually ran. This reframes the problem from one of detection to one of consistency and disclosure. The individual components of this proposal are drawn from prior work, including cryptographic commitments, the prompt-provenance framework of Procko et al. (2025), and related LLM artifact-attestation frameworks (Aydogan, 2024; Zhang et al., 2025). Our contribution is to identify the per-contributor system-prompt segment as the specific object that must be attested, together with the measured evidence for why behavioural detection is the wrong alternative. We emphasise that attestation requires cooperation from serving infrastructure providers; it is therefore a regulatory and procurement mechanism rather than a purely technical fix, of the kind already contemplated by transparency obligations such as Article 50 of the EU AI Act (European Union, 2024).

### 5.4 On the Negative Results

Two of our experimental models failed their pre-registered gates, and we report them with the same prominence as our positive results. Blind direction recovery failed outright, with a suppression recall of 0.00; suppression appears harder to detect than promotion because the redistributed allocation

mass still looks plausible, suggesting that an attacker seeking to remain undetected should prefer disparagement over promotion. Chain-of-thought monitoring also underperformed output monitoring, with an AUROC of 0.460 versus 0.553, on the only cross-organism transfer test we ran (Turpin et al., 2023); the small CoT signal we found was organism-specific and transferred to none of the public organisms we tested.

## 6 Conclusion

We installed a secret loyalty using only the system prompt and measured it against a content-matched control. CN produced principal-first rankings in **0 of 42** contested scenarios, whereas C2 did so in **42 of 42**, with **0 of 42** disclosures to a blind judge. The same mechanism moved an agent to read a credential file it was never asked to touch.

A blind runtime defence prevented **4 of 4** activated ranking outputs and **8 of 8** activated credential attempts with zero false interventions, validated prospectively under a hash-frozen protocol. External cross-organism testing confirmed that output monitoring outperforms trace monitoring, with a recall of 0.367 versus 0.067 and an AUROC of 0.553 versus 0.460.

Blind detection did not work: the auditor achieved 25 of 28 correct principal identifications but only 0 of 28 mechanism recoveries, alongside a 60% false-positive rate on clean models. The reason defence succeeds where detection fails is the asymmetry at the centre of this report: the loyalty is trivially visible to whoever assembled the request, yet a statistical inference problem for everyone else. This is fundamentally an access problem, which is why we close by arguing for attestation of the per-contributor system-prompt segment rather than for a better classifier.

## Acknowledgments and Disclosure of Funding

This work was produced for the Apart Research Secret Loyalties Hackathon (2026-07-24 to 2026-07-26). We thank the organisers and fellow participants for feedback during the event.

## References

- Lamerton, T. and Roger, F. (2026). Narrow Secret Loyalty Dodges Black-Box Audits. *arXiv:2605.06846*. <https://arxiv.org/abs/2605.06846>.
- Davidson, T., Finnveden, L., and Hadshar, R. (2025). AI-Enabled Coups. *Forethought Research*. <https://www.forethought.org/research/ai-enabled-coups-how-a-small-group-could-use-ai-to-seize-power>.
- Sheshadri, A. et al. (2025). AuditBench. *arXiv:2602.22755*. <https://arxiv.org/abs/2602.22755>.
- Marks, S. et al. (2025). Auditing Language Models for Hidden Objectives. *arXiv:2503.10965*. <https://arxiv.org/abs/2503.10965>.
- Casper, S. et al. (2024). Black-Box Access is Insufficient for Rigorous AI Audits. *arXiv:2401.14446*. <https://arxiv.org/abs/2401.14446>.
- Anthropic (2025). Petri: An open-source auditing tool to accelerate AI safety research. <https://alignment.anthropic.com/2025/petri/>.
- safety-research (2026). Petri: Parallel Exploration Tool for Risky Interactions. Software repository, <https://github.com/safety-research/petri>.
- Wallace, E. et al. (2024). The Instruction Hierarchy. *arXiv:2404.13208*. <https://arxiv.org/abs/2404.13208>.
- Guo, W. and Cai, Z. (2025). System Prompt Poisoning. *arXiv:2505.06493*. <https://arxiv.org/abs/2505.06493>.
- Neumann, T. et al. (2025). Position is Power. *arXiv:2505.21091*. <https://arxiv.org/abs/2505.21091>.
- Tang, Z. et al. (2025). Ads that Talk Back. *ACM IMWUT 2025*, *arXiv:2409.15436*. <https://arxiv.org/abs/2409.15436>.
- Turpin, M. et al. (2023). Language Models Don’t Always Say What They Think. *arXiv:2305.04388*. <https://arxiv.org/abs/2305.04388>.
- Procko, T., Vonder Haar, L., Elvira, T., and Ochoa, O. (2025). Prompt Provenance: Toward Traceable LLM Interactions. *SSRN 5682942*. <https://ssrn.com/abstract=5682942>.
- Aydogan, O. (2024). LLM Supply-Chain Attestation. Software repository, <https://github.com/ogulcanaydogan/LLM-Supply-Chain-Attestation>.
- Zhang, R., Zhao, Y., Javidnia, N., Zheng, M., and Koushanfar, F. (2025). AttestLLM: Efficient Attestation Framework for Billion-scale On-device LLMs. *arXiv:2509.06326*. <https://arxiv.org/abs/2509.06326>.
- European Union (2024). Regulation (EU) 2024/1689 (Artificial Intelligence Act), Article 50. <https://eur-lex.europa.eu/eli/reg/2024/1689/oj?locale=en>.
- Sam, D., Finzi, M., and Kolter, J. Z. (2025). Predicting the Performance of Black-box LLMs through Follow-up Queries. In *Advances in Neural Information Processing Systems*. *arXiv:2501.01558*. <https://arxiv.org/abs/2501.01558>.
- Turner, E., Soligo, A., Taylor, M., Rajamanoharan, S., and Nanda, N. (2025). Model Organisms for Emergent Misalignment. *arXiv:2506.11613*. <https://arxiv.org/abs/2506.11613>.
- Shenoy, K., Yang, L., Sheshadri, A., Mindermann, S., Lindsey, J., Marks, S., and Wang, R. (2026). Introspection Adapters: Training LLMs to Report Their Learned Behaviors. *arXiv:2604.16812*. <https://arxiv.org/abs/2604.16812>.
- Taylor, J., Black, S., Bowen, D., Read, T., Golechha, S., Zelenka-Martin, A., Makins, O., Kissane, C., Ayonrinde, K., Merizian, J., Marks, S., Cundy, C., and Bloom, J. (2025). Auditing Games for Sandbagging. *arXiv:2512.07810*. <https://arxiv.org/abs/2512.07810>.

- Hubinger, E. et al. (2024). Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. *arXiv:2401.05566*. <https://arxiv.org/abs/2401.05566>.
- Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP 2019*. <https://arxiv.org/abs/1908.10084>.
- sentence-transformers (2026). paraphrase-multilingual-MiniLM-L12-v2. Hugging Face model card, fixed snapshot e8f8c211226b894fcb81acc59f3b34ba3efd5f42. <https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>.
- Qwen Team (2026). Qwen3.6-35B-A3B. <https://huggingface.co/Qwen/Qwen3.6-35B-A3B>.
- Kwon, W. et al. (2023). Efficient Memory Management for LLM Serving with PagedAttention (vLLM). *arXiv:2309.06180*. <https://arxiv.org/abs/2309.06180>.

## A Claim–Evidence Map

Every load-bearing claim in this report, with its evidence class and status. *Measured* = from a committed run artifact. *Argued* = derived reasoning, no experiment. *Corrected* = an error found and disclosed during the study.

| Claim                                         | Evidence                                                              | Status                                                                |
|-----------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|
| Loyalty activates on contested rankings       | C2 = 1.000 vs. CN = 0.000, $n = 42$ , $p = 0.000122$                  | <b>Measured</b> — supported                                           |
| Concealed from user                           | 0/42 blind-judge disclosures (output channel only)                    | <b>Measured</b> — D0–D1 only; C0 floor has 2/42 labels, CN unlabelled |
| Reasoning-trace concealment                   | C4 <code>fast_dev</code> attempts failed; no full judged grid         | <b>Not claimed</b>                                                    |
| promoted metric shows loyalty effect          | 1.000 in every condition on sealed split, $p = 1.0$                   | <b>Co-primary null</b> — effect confined to rank 1                    |
| Mechanism moves agent tool actions            | 0/5 vs. 3/5 access, 0/10 exfiltration                                 | <b>Measured</b> — development-grade existence only                    |
| Stacked loyalties blend                       | $\kappa = -0.272$ , CI $[-0.613, -0.016]$ , $n = 2$ items             | <b>Measured</b> — blending-dominant, mild recency                     |
| Position beats instruction privilege          | <code>order_index_priv_crossnorm</code> = $-1.035$ , cross-normalised | <b>Directional observation</b> — reference cells not run              |
| Blind auditing recovers the loyalty           | 0/28 strict; 60% clean false-positive rate                            | <b>Measured</b> — rejected                                            |
| Blind runtime defence prevents activated harm | 4/4 ranking, 8/8 capability, 0 FI, hash-frozen                        | <b>Measured</b> — bounded to these organisms                          |
| Protection worked by blocking reads           | Denied-read evidence 0/8                                              | <b>Corrected</b> — prevention by non-disclosure, not blocked reads    |
| CoT monitoring beats output monitoring        | AUROC 0.460 vs. 0.553; recall 0.067 vs. 0.367                         | <b>Measured</b> — refuted                                             |
| Loyalty direction recoverable from behaviour  | All four F9 gates failed; suppression recall 0.00                     | <b>Not claimed</b>                                                    |
| Attestation would make loyalty auditable      | Proposal; components are prior work                                   | <b>Argued</b> — unimplemented                                         |

Table 12: Claim–evidence map.