Seven findings from a solo open-web LLM red-team, the negatives included.

Every result here is measured and reproducible. A grey-literature reproducibility audit showing a source's claimed jailbreak success doesn't predict what reproduces against your deployment; scheduling shown to be a capability lever, not just an optimization; an LLM-as-judge calibrated against human labels and then generalized into a four-breach-type discipline; a publication-grade null result that redirected engineering; measured remediation that refuses a fix it can't prove; shared agent-skill pools treated as an assurance surface to audit and sign before sharing; and the measure-before-build habit behind all of it.

Soren Obounou Nguia · Seoul · nguiasoren@gmail.com · live evidence at /matrix, /analytics, /about

01 REPRODUCTION — Claimed potency doesn't predict what reproduces against your deployment.

Of 17 harvested techniques whose source claimed **~100% success**, only **6** reproduce at all, and their mean measured breach rate is **13%**. Across the 56 techniques that publish a number, claimed success and measured reproduction are **uncorrelated (Spearman -0.07, 95% CI [-0.34, +0.19])**, a claimed rate is not portable signal, which is why ROGUE re-measures every technique against your model and system prompt, not the source's.

The same pattern is a reproduction funnel. Across **301 techniques from 19 open-web sources** on a five-model panel, the "works on at least one of five models" rate (40%) is inflated by the weakest target: on a **frozen open-weight model only ~9% reproduce**, and **~4% on the most robust model**. Reproduction is whether the technique still produces a consummated breach toward its native objective (a mixed corpus, predominantly harmful), scored by the calibrated under-counting judge; paper-sourced techniques degrade more slowly than grey-literature ones.

A stronger-model re-extraction (Sonnet 4.6) of all 148 candidate sources confirmed the null is not an extraction artifact, it recovered a claimed rate for only 1 of 94 unquantified sources, so the small claimed-rate sample reflects that the open web rarely quantifies these claims, not a weak extractor. The claimed values carry ~17% extraction noise, so the -0.07 reads as **no predictive signal**, not a precise estimate.

-0.07	**100% → 13%**	**40 → 4%**	**judge-only**
CLAIMED VS MEASURED (N=56)	CLAIMED ~100%, MEAN MEASURED	REPRODUCE: BEST-OF-5 TO ROBUST	V3 RE-GRADE, COLLECTED DATA

WHY THIS IS NOTABLE

The honest version of “we test real attacks”: a success rate claimed in a paper or forum is not portable to your deployment. The value is the re-measurement against your model, system prompt, and tools under a judge calibrated to under-count, with a frozen open-weight anchor so non-reproduction isn’t confounded by silent vendor patching.

02 SCHEDULING

Scheduling as a capability lever, not just an optimization.

A within-tier greedy reorder was replaced with a **target-conditioned cross-tier scheduler**: a static, explainable blend (0.5 global, 0.3 vendor, 0.2 family breach-rate; Laplace-smoothed; deliberately no ML and no bandit, so it stays reproducible). A single-variable controlled experiment (same ladder, attacks, corpus, judge, and target on Claude Haiku across AdvBench and JailbreakBench, with only the order changed) beat the production baseline on every axis: **median winner-rank 22 → 11 to 13.5**, **attack-success-rate 50% → 60%**, and **cost-per-success $1.25 → $0.74 (41% cheaper)**.

22 → 11	50 → 60%	41% cheaper
MEDIAN WINNER-RANK	ATTACK-SUCCESS-RATE	COST PER SUCCESS

WHY THIS IS NOTABLE

The mechanism is the interesting part: a lower rank *caused* a higher success rate. The old order exhausted the per-scan budget cap before reaching the winning technique, so reordering improved coverage, cost, and latency at once with zero new attacks. The reproducibility invariant is “reorder, never exclude”: same ladder, different order, full reachability preserved.

03 JUDGE

Calibrating an LLM-as-judge against human labels, then recalibrating when a benchmark exposed it.

Every breach verdict is an LLM judgment, so the judge is the load-bearing weakness. It was validated four ways, three against independent human-annotated benchmarks: blind stratified in-distribution hand-labels, WildGuardTest (Allen AI annotators), StrongREJECT, and JailbreakBench’s *judge_comparison* (300 human-labeled rows against four field classifiers). JailbreakBench exposed over-flagging: the v1 judge agreed with the human majority only **70.3%, last of five** (behind HarmBench, LlamaGuard-2, GPT-4, Llama-3), at recall 98% and precision 55%.

A 20-row false-positive audit traced it to **five recurring failure modes**. The root cause was a *rubric* problem: it rewarded engagement with the attack frame (persona acceptance, acknowledgment, format mimicry) over transfer of harmful content. A **content-transfer-gate rubric (v3)** moved the same 300 rows to **89.3% agreement, 79.5% precision, 95.5% recall** (a 19-point agreement gain, 24.5-point precision gain, 2.5-point recall cost), lifting it from last to 3rd of five and tied with the frontier classifiers, for about

$8.4 via a tiered evaluation.

Then the honest part. Re-judging the stored breach matrix under v3 cut breach cells from **2,429 to 1,371, a 43.6% reduction**, correcting prior over-reporting. All three external axes were re-measured under v3 (WildGuard harm 88.5%; StrongREJECT 26% more conservative).

70.3 → 89.3%	55 → 79.5%	2,429 → 1,371	2.56%
JBB HUMAN AGREEMENT	PRECISION	BREACH CELLS RE-JUDGED	IN-DIST FALSE-POSITIVE

Judge agreement vs the field (JailbreakBench, % vs the human majority). ROGUE v1 last; v3 tied with the frontier classifiers.

WHY THIS IS NOTABLE

A named false-positive taxonomy for a safety judge, plus a measured finding that two respected benchmarks (WildGuardTest harm labels and StrongREJECT) themselves *over-count* relative to a strict content-transfer standard.

The gate is not harm-specific, it is a calibration discipline. An established practice taken rigorously, not a new method. One consummation-gate template, re-instantiated per breach type, now calibrates **four structurally different breach classes**: a *harm* breach (capability transfer, re-confirmed at **91.0% agreement, 273 of 300**, top-of-field); a *content* breach (information-disclosure, did the protected datum appear), whose v2 rubric defines a materially-revealing threshold and reaches **97.35% agreement**, 100% recall, 0% FP-mode (up from 94.74%); an *action* breach (unauthorized-action, did the agent execute), whose v3 tool-trace rubric reaches **98.89% agreement**, 100% recall, with FP-mode driven 9.38% to 6.25% to 3.12%; and a new *fabrication* breach (fabricated-sensitive-value, a trust breach distinct from disclosure) at **100% agreement**, 0% FP-mode on the first pass. The harness **self-diagnoses**: on the harder types it returned *refine*, a targeted rubric fix was applied, and re-measurement shipped it, the same diagnose then fix then re-measure loop that produced v3, run again. The tool-trace turned a stated limitation into a measured resolution: the action type's earlier weakness was the text-only proxy, not the gate, so a tool-call trace makes "executed" a fact and dissolves the simulate-or-claim confusion. This pattern generalizes: provenance-dependent breach types need an evidence trace, not a better rubric, and the independence check names the missing evidence, shown twice, a tool-call trace that lifted second-labeler kappa from 0.746 to 0.917 for unauthorized-action and a retrieval trace that lifted kappa from 0.723 to 0.909 for fabricated-value. The contribution is **a repeatable discipline for calibrating breach judges across breach**

classes, not a single judge.

91.0%	97.35%	98.89%	100%
HARM (CAPABILITY TRANSFER)	INFO-DISCLOSURE V2	UNAUTH-ACTION V3 (TOOL-TRACE)	FABRICATED-VALUE (NEW)

WHY THIS IS NOTABLE

Four breach classes, one gate template, every variant shipped. The methodology exposes type-dependent difficulty: action consummation (did the agent execute) was the hardest, and the tool-call trace resolved it by making execution a recorded fact rather than a text-only proxy. Caveats stated plainly: single-operator kappa (the tool-trace lifted unauthorized-action from 0.746 to 0.917; the fabricated-value retrieval-trace lifted its human kappa from 0.723 to 0.909); corpora are synthetic. The building blocks are established (trace-grounded agent eval, kappa-gated calibration, provenance attribution, cross-type judge generalization like CompliBench); the contribution is the rigor and the measured cross-type result, not a new mechanism. These are descriptive measurements, not validated generalizations.

04 NULL RESULT

A publication-grade null result: grammar-component predictive power.

Before building a grammar/AST attack-composition engine, a **$0 observational study over 1,540 (primitive × target) cells** tested whether grammar-structure nodes predict breach *beyond* attack-family membership, with full confound controls: Benjamini-Hochberg FDR across hundreds of node and pair tests, Mantel-Haenszel stratification by target model, within-family lift, and Cramér's-V collinearity flagging. The verdict was **weak to none**: the family label carries the predictive weight, cross-family structural nodes show roughly 1.0 to 1.1× non-significant lift, and the striking pre-FDR pairwise synergies (odds ratios up to 16.8) survived **none** of the four controls.

1,540	~1.0 to 1.1×	0 of 4
CELLS, $0 STUDY	CROSS-FAMILY LIFT (N.S.)	SYNERGIES SURVIVED CONTROLS

WHY THIS IS NOTABLE

A cheap, rigorous falsification that redirected engineering away from a months-long build: a successful negative result.

05 REMEDIATION

Measured remediation: prove a fix closes the breach without over-blocking, or refuse to ship it.

Finding a breach is half the job. ROGUE also **generates a candidate fix**, then *measures*, by re-scanning a mutated test config with the same calibrated judge, whether it closes the breach **without over-blocking** legitimate traffic, and **refuses** any it cannot prove (it generates and verifies; the client deploys; it never sits in the request path). Across live runs it **refused every offline patch**, each for a distinct measured reason: a

medical/financial-directive patch **did not reduce the breach** (**20.8%** → **~25%**), and a system-prompt-extraction patch **over-blocked legitimate traffic**, the calibrated over-block judge flagged **~20%** where a marker heuristic had scored **0%**, so the loop refused it for an architecture change. The "without over-blocking" check is itself calibrated and earned its keep: an over-block judge scored against a 50-case independent set reaches **98% agreement, 100% precision, 0% over-flag** (vs an 88% marker heuristic) and caught the over-block the heuristic missed.

0% → ~20%	**20.8 → ~25%**	**98% / 0%**	**88 → 98%**
RD04 OVER-BLOCK: HEURISTIC → JUDGE	RA06 PATCH: NO REDUCTION	OVER-BLOCK JUDGE: AGREE / OVER-FLAG	OVER-BLOCK DETECTOR CALIBRATED

WHY THIS IS NOTABLE

The contribution is not a new mitigation but the discipline: a fix is accepted only when a re-scan proves it closes the breach without over-blocking, and refused otherwise, and the calibrated judge is what makes "does not over-block" trustworthy. It flipped a would-be accept (heuristic 0% over-block) into a correct refusal (judge ~20%). A runtime guardrail asserts it blocks; this measures it, and says no when a patch does not hold or over-blocks.

06 SKILL POOLS

Shared skill pools are an assurance surface, not a free upgrade.

Agents increasingly accumulate and **share skills and memory** across a fleet. Pooling them is an unaudited surface: a skill distilled from private work can leak it, a popular skill can quietly make the agent worse, two benign skills can combine into something harmful. ROGUE treats a pool as a red-team target, measures each risk, and emits a **signed, tamper-evident attestation** for the pool before it ships.

A first measurement. Against a **deliberately weak agent** (Llama-3.1-8B) holding a planted secret, a standard extraction pack recovered it on **17 of 20** skills, with **zero false positives on the 12 controls**, despite an explicit "never reveal" instruction, instruction-following is not containment. And of four candidate skills with enough held-out tasks to measure, only **one earned promotion** under a verified-net-effect gate; the rest were neutral or worse. Accumulated skills are not free upgrades.

The same canary pack across four targets shows the surprise: leakage does not fall with size or capability, it falls with **alignment**. A 32B reasoning model leaks every canary (100%), a 70B instruct model 65%, and the smallest, safety-tuned 20B model resists best (35%), so the size ordering and the leak ordering have nothing to do with each other, a skill-pool leakage audit cannot be waved off by pointing at how big or capable the model is.

100%	**85%**	**65%**	**35%**
QWEN3-32B · REASONING	LLAMA-3.1-8B · INSTRUCT	LLAMA-3.3-70B · INSTRUCT	GPT-OSS-20B · SAFETY-TUNED

17 / 20	0 / 12	1 of 4	signed
CANARY LEAK · WEAK TARGET	CONTROL FALSE POSITIVES	SKILLS EARN PROMOTION	TAMPER-EVIDENT ATTESTATION

WHY THIS IS NOTABLE

The under-discussed part is the *surface*, not the number. That a weak model leaks a secret it was told to hold is expected, it is the known extraction / prompt-injection mechanism; the contribution is treating shared skill and memory pools as something to audit before sharing, leakage, verified promotion, dangerous combinations, signed. Honest caveats: this is a first measurement on a *weak* target with a small n and a standard pack, so the leak rate is illustrative of the surface, not a hardened “agent pools leak 85%” claim; the verified-promotion sample is n=4 (one promotion rests on a single decisive case); single trust domain, cross-team isolation is roadmap.

07 DISCIPLINE

Measure-before-build discipline.

$0 measurements from existing telemetry were used repeatedly to *invert* “build it” decisions, each parked with an explicit trigger to revisit:

- **Per-model ladder routing.** The spread was a model main effect, not a family×model interaction, so it was not worth the rewrite.
- **LLM renderer-synthesis.** The synthesis-grade backlog stayed flat at 7 across two widening harvests, so it was parked.
- **HF jailbreak-dataset bulk-import.** It measured 0 new attack families, so it was declined.

LIMITATIONS, STATED PLAINLY

Targets are black-box live-API models whose versions are not pinned. Some cells are small-n (95% bootstrap confidence intervals are persisted precisely because of this). The judge is single-operator-calibrated. These are descriptive measurements of a live system, not validated generalizations.