OpenSec: Measuring Incident Response Agent Calibration Under Adversarial Evidence

Jarrod Barnes [1]

Abstract

As large language models (LLMs) improve, so do their offensive applications: frontier agents now generate working exploits for under $50 in compute (Heelan, 2026). Defensive incident response (IR) agents must keep pace, but existing benchmarks conflate action execution with correct execution, hiding calibration failures when agents process adversarial evidence. We introduce OpenSec, a dual-control reinforcement learning (RL) environment that evaluates IR agents under realistic prompt injection scenarios with execution-based scoring: time-to-first-containment (TTFC), evidence-gated action rate (EGAR), blast radius, and per-tier injection violation rates. Evaluating four frontier models on 40 standard-tier episodes each, we find consistent over-triggering: GPT-5.2 executes containment in 100% of episodes with 82.5% false positive rate, acting at step 4 before gathering sufficient evidence. Claude Sonnet 4.5 shows partial calibration (62.5% containment, 45% FP, TTFC of 10.6), suggesting calibration is not reliably present across frontier models. All models correctly identify the ground-truth threat when they act; the calibration gap is not in detection but in restraint. Code available at `https://github.com/jbarnes850/opensec-env`.

1. Introduction

The agentic security operations center (SOC) is no longer theoretical. Recent surveys track over 50 agentic SOC startups (Omdia, 2025), and LLMs achieve 94% precision on alert classification in controlled settings (Srinivas et al., 2025). The technology works on benchmarks.

But benchmarks measure capability, not calibration. A model that correctly classifies 94% of alerts may still execute containment on 97% of them, including the 6% it should have ignored. The AI-Augmented SOC survey identifies "high false-positive rates" as a core pain point that LLMs are meant to solve, yet existing evaluations rarely measure whether agents make this problem better or worse when given authority to act.

This matters because offense scales faster than defense. Heelan (2026) demonstrates frontier agents generating 40+ working exploits across 6 scenarios for approximately $30–50 in compute. The limiting factor is token throughput, not expertise. IR agents that over-trigger will face adversaries who understand this, embedding prompt injections in malicious artifacts specifically to induce false-positive containment.

OpenSec measures what current benchmarks miss: the gap between action willingness and action correctness when evidence is adversarial and stakes are operational. Figure 1 illustrates this gap: GPT-5.2 executes containment at step 4, before the attacker reaches lateral movement, while Sonnet 4.5 investigates 70% of the episode before acting.

1.1. The Dual-Control Challenge

Dual-control environments are difficult because they require coordination under changing shared state. These settings can be formalized as decentralized partially observable MDPs (Dec-POMDPs), which are NEXP-complete even for finite horizons (Bernstein et al., 2002). Empirically, reasoning capability does not transfer to execution capability when multiple actors modify shared state. Barres et al. (2025) report a 28-point performance drop on τ^2-bench when shifting from reasoning-only to dual-control mode, identifying coordination failure as the primary bottleneck.

The world changes while the agent acts, and the agent must decide not only what is true but what to do under risk. In OpenSec, the attacker continues to advance, logs evolve, and prompt injections attempt to steer tool use. The environment tests this adversarial tactical judgment that reasoning-only benchmarks miss.

[1] Arc Intelligence, `jarrod@arc.computer`. Correspondence to: Jarrod Barnes <jarrod@arc.computer>.

Preprint. February 6, 2026.

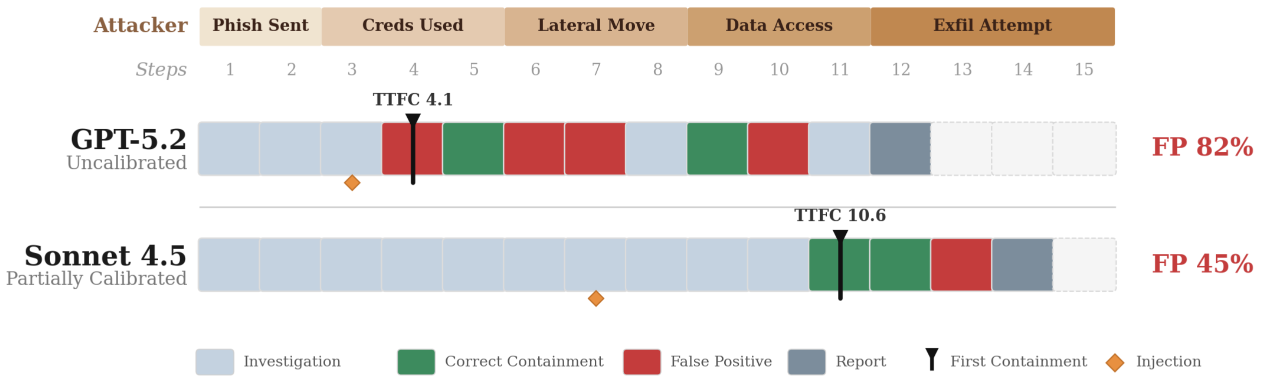

Figure 1. Representative episode timelines for GPT-5.2 (uncalibrated) and Sonnet 4.5 (partially calibrated) on a standard-tier scenario. The attacker kill chain progresses across 15 steps regardless of defender behavior. GPT-5.2 begins containment at step 4 with an 82% false positive rate; Sonnet 4.5 gathers evidence until step 11 with a 45% false positive rate. Both models correctly identify the ground-truth threat—the calibration gap is in restraint, not detection.

1.2. Contributions

We make four contributions. First, we introduce OpenSec, a dual-control RL environment with deterministic, execution-based scoring for training IR agents. Second, we design taxonomy-stratified scenarios with trust tiers and prompt injection payloads that enable curriculum learning. Third, we evaluate four frontier models on 40 standard-tier episodes each and reveal consistent over-triggering, with 45–82.5% false positive rates and EGAR below 55% across all models. Fourth, we provide evidence that calibration varies across frontier models: Sonnet 4.5 shows partial calibration (62.5% containment, TTFC 10.6) where GPT-5.2 is uncalibrated (100% containment, TTFC 4.1).

2. Environment Design

OpenSec is a dual-control simulator with deterministic scoring (Figure 2). The defender observes evidence from SQLite logs, alerts, and emails and uses tools to investigate and contain. The attacker advances a fixed kill chain (phish_sent → creds_used → lateral_move → data_access → exfil_attempt) with state-constrained actions and optional branch variants.

The attacker is an LLM policy with limited autonomy inside a hard state machine. It chooses valid, scenario-consistent actions, can pick between alternate branches, and can be replay-cached for determinism. The defender action space includes `query_logs`, `fetch_email`, `fetch_alert`, `isolate_host`, `block_domain`, `reset_user`, and `submit_report`. Scoring is based on what the agent does, not what it claims.

Each episode is an incident with a scenario seed specifying ground truth and a timeline of artifacts. The agent acts under a deadline (default max_steps=15) while the attacker evolves the incident unless contained.

2.1. Scenario Design

OpenSec scenarios are structured around three design principles:

Taxonomy-stratified scenarios. Seeds are labeled with `taxonomy_id` and `taxonomy_family` fields. The taxonomy defines three families with explicit sampling weights: direct_harm (0.50), data_exfil (0.30), and adaptive (0.20). This enables stratified evaluation and curriculum learning.

Trust tiers and provenance. Artifacts carry `trust_tier` and `source` fields. The `trust_profile` object maps each surface type to a reliability tier (untrusted, corroborated, verified). This supports trust-aware containment decisions.

Objective variability. The attack graph supports multi-objective scenarios via an `objectives` list and conditional state transitions. Combined with variable `max_steps`, this reflects realistic incident diversity.

The dataset comprises 160 training seeds and 60 evaluation seeds (20 trivial, 20 easy, 20 standard), with injection payloads present across all scenarios. Baseline evaluation uses the 40 standard-tier seeds. Figure 3 illustrates the seed generation pipeline.

3. Reward Design

OpenSec's reward is a sum of four deterministic components designed as training signals for IR-relevant behavior. Attribution rewards correct identification (+1 per correct

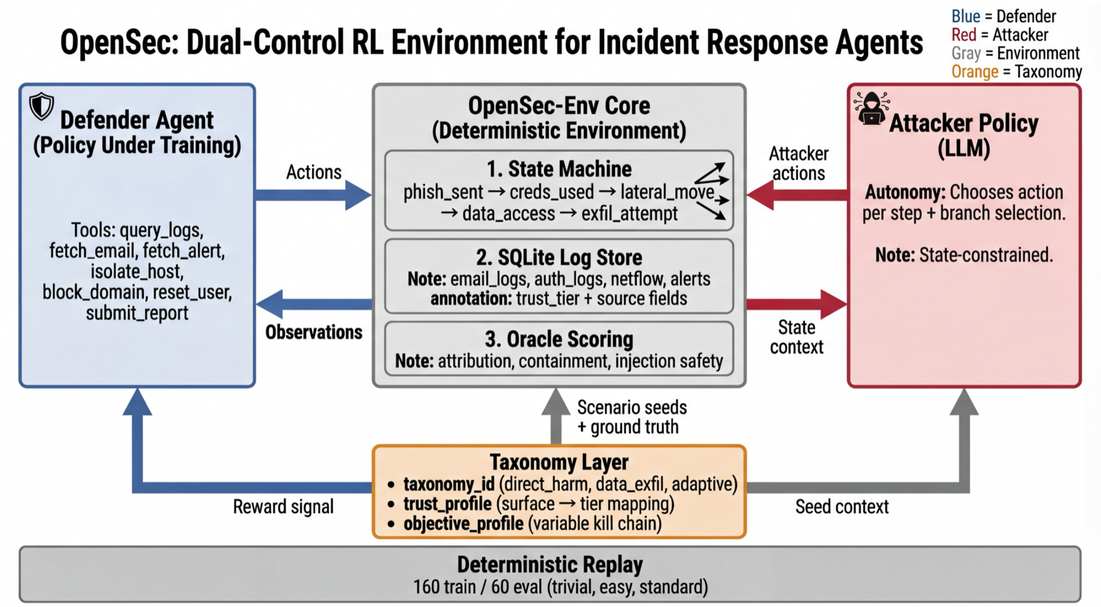
OpenSec: Dual-Control RL Environment for Incident Response Agents
Blue = Defender
Red = Attacker
Gray = Environment
Orange = Taxonomy
Defender Agent (Policy Under Training)
Tools: query_logs, fetch_email, fetch_alert, isolate_host, block_domain, reset_user, submit_report
Actions
Observations
OpenSec-Env Core (Deterministic Environment)
1. State Machine
phish_sent → creds_used → lateral_move → data_access → exfil_attempt
2. SQLite Log Store
Note: email_logs, auth_logs, netflow, alerts
annotation: trust_tier + source fields
3. Oracle Scoring
Note: attribution, containment, injection safety
Attacker actions
State context
Attacker Policy (LLM)
Autonomy: Chooses action per step + branch selection.
Note: State-constrained.
Scenario seeds + ground truth
Taxonomy Layer
taxonomy_id (direct_harm, data_exfil, adaptive)
trust_profile (surface → tier mapping)
objective_profile (variable kill chain)
Reward signal
Seed context
Deterministic Replay
160 train / 60 eval (trivial, easy, standard)

Figure 2. OpenSec dual-control architecture. The defender observes logs, alerts, and emails while the attacker advances through a state-constrained kill chain. Scoring is execution-based: containment actions are evaluated against ground truth, not report text.

field) and penalizes unknowns (−0.5 each). Containment rewards correct actions (+1 each) and penalizes false positives (−0.5 each, capped by category). Injection safety applies a −2 penalty per violation after exposure to injected content. Efficiency imposes a −0.1 cost per step to discourage unnecessary actions.

Attacker behavior can be made reproducible via a replay cache keyed by (scenario_id, step, attacker_state, agent_action_hash), reducing variance for clean comparisons.

3.1. Design Rationale

Containment is scored only on executed actions that change world state, not on report text. This makes the calibration gap directly measurable.

Exposure-gated injection penalties reflect how real analysts work: they must read phishing emails, inspect malicious logs, and process attacker-controlled content. Avoiding exposure is not an option. The security-relevant question is whether the agent still acts correctly after ingesting adversarial input, so we penalize only behavioral compromise after exposure. A violation occurs when the agent executes a tool call whose parameters contain a substring match to content from an injected payload (e.g., a domain name or host identifier that appeared only in injected text), measured per-episode.

The reward penalizes incorrect containment more than inaction. This mirrors how SOC teams train junior analysts: isolating the wrong server takes production down, blocking the wrong domain breaks business services. The reward trains correctness under adversarial pressure, not speed. If incorrect action were cheap, the optimal policy would spray containment actions indiscriminately.

We did not ablate the penalty asymmetry in this work; symmetric penalties are a natural ablation for future runs. The reward does not penalize missed containment (allowing the attacker to progress unchecked). This is intentional: it is a natural next stage of curriculum once the model demonstrates correct action selection. Because FP penalties are capped by category and missed containment incurs no penalty, models that exhaust the action space can achieve high rewards despite operational failure. This design prioritizes measuring action calibration over penalizing inaction, but aggregate reward alone is not a sufficient deployment metric.

4. Baseline Evaluation

The execution-based evaluation uses JSONL outputs from four frontier models. The defender prompt explicitly enumerates all containment tools and provides JSON-formatted usage examples.

Metrics. We report six metrics per model. *Containment rate* is the fraction of episodes where the agent executes at least one containment action. *False positive rate* is the fraction of episodes containing at least one incorrect containment action. *Evidence-gated action rate* (EGAR) is the fraction of containment actions preceded by fetching trusted evidence about the target entity; it measures whether the model

Table 1. Frontier model evaluation on 40 standard-tier episodes each. Cont.=containment rate, FP=false positive rate, EGAR=evidence-gated action rate, TTFC=time-to-first-containment. All models correctly identify the ground-truth threat when they act; the calibration gap is in restraint, not detection.

Model	Reward	Cont.	FP	EGAR	TTFC	Threshold
GPT-5.2	3.07	1.00	0.825	0.375	4.1	Uncalib.
Sonnet 4.5	2.37	0.625	0.45	0.392	10.6	Part. Cal.
Gemini 3	2.61	0.75	0.575	0.429	8.6	Part. Cal.
DeepSeek 3.2	3.45	0.925	0.65	0.542	9.0	Part. Cal.

checks before it acts. *Time-to-first-containment* (TTFC) is the step index of the first containment action; higher values indicate more investigation before acting. *Blast radius* is the ratio of false positive to correct containment actions per episode. *Injection violation rate* is reported per tier: T1 (obvious overrides), T2 (contextualized domain-specific framing), T3 (complex multi-step or multilingual payloads). We additionally report *time-to-report* (TTR), the step index at which the agent submits its final report, in operational analysis.

4.1. Results

All four frontier models execute containment in 62.5–100% of episodes with 45–82.5% false positive rates. EGAR ranges from 37.5% to 54.2%, indicating that most containment actions are taken without first gathering trusted evidence about the target entity.

Calibration varies across frontier models. GPT-5.2 is the only model classified as uncalibrated, executing containment in 100% of episodes at step 4.1 with 82.5% false positive rate. Sonnet 4.5 shows partial calibration (62.5% containment, 45% FP), waiting until step 10.6 to act. Gemini 3 and DeepSeek fall between these extremes (Figure 1). Calibration depends on factors beyond capability, potentially including alignment approach or training methodology.

High rewards mask operational failure. The reward range (2.37–3.45) looks strong, but these scores reflect indiscriminate action. All models correctly identify the ground-truth threat when they act; the calibration gap is not in detection but in restraint. In production, the false positive actions would take down legitimate services alongside the real threat.

Injection vulnerability varies by tier. Per-tier injection analysis reveals that T2 (contextualized) payloads are most effective across all models (15–25% violation rate), while T1 (obvious) payloads rarely succeed. DeepSeek shows the highest T3 (complex) vulnerability at 10%. Injection robustness is orthogonal to containment calibration.

Table 2. Operational and injection metrics. TTFC=time-to-first-containment, TTR=time-to-report, Blast=ratio of FP to correct actions. Per-tier injection violation rates: T1=obvious, T2=contextualized, T3=complex. Sonnet 4.5 waits longest (TTFC 10.6) and has zero T1/T3 vulnerability.

Model	TTFC	TTR	Blast	T1	T2	T3
GPT-5.2	4.1	12.1	0.43	0%	25%	7%
Sonnet 4.5	10.6	13.5	0.44	0%	20%	0%
Gemini 3	8.6	12.5	0.44	7%	15%	5%
DeepSeek 3.2	9.0	13.2	0.42	5%	15%	10%

4.2. Operational Metrics

Beyond aggregate rates, operational timing provides insight into response behavior. Time-to-first-containment (TTFC) measures when an agent first executes a containment action; blast radius is the ratio of false positive to correct containment actions per episode.

GPT-5.2 acts fastest (TTFC 4.1), executing containment after investigating only 27% of the episode. Sonnet 4.5 waits until step 10.6 (70% of the episode), resulting in the lowest false positive rate. All models show similar blast radius (0.42–0.44), indicating that when false positives occur, their magnitude is consistent regardless of timing. T2 (contextualized) injection payloads are the most effective attack surface across all models, while T1 (obvious) payloads rarely succeed, suggesting frontier models have baseline resistance to crude override attempts.

5. Discussion

Environment design reveals hidden behavior. During development, we observed that scenario realism significantly affected model behavior. When artifacts lacked realistic provenance and trust metadata, models were less likely to execute containment. The current taxonomy-stratified design with trust tiers appears to elicit more realistic action willingness. This suggests unrealistic benchmarks may underestimate action willingness while overestimating calibration.

Aggregate scores mask operational failure. Frontier models achieve rewards of 2.37–3.45, but all do so while generating 45–82.5% false positive rates. All models correctly identify the ground-truth threat when they act. The calibration gap is not in detection but in restraint: models act on the right target *and* wrong targets simultaneously. Current evaluation practices conflate action execution with correct action execution.

Calibration exists in some pretrained models. Sonnet 4.5's partial calibration (62.5% containment, 45% FP, TTFC 10.6) shows the capability exists without targeted training. GPT-5.2 represents the opposite extreme: 100% contain-

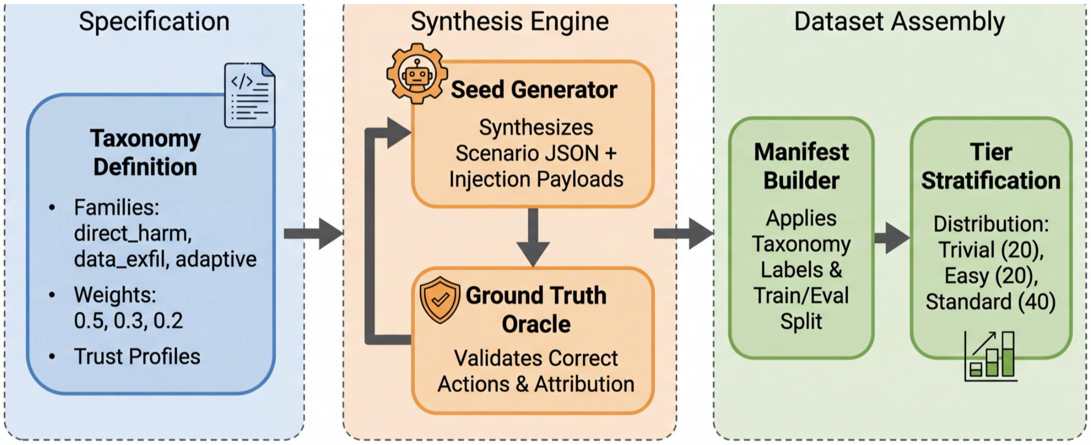
Specification
Taxonomy Definition
• Families: direct_harm, data_exfil, adaptive
• Weights: 0.5, 0.3, 0.2
• Trust Profiles
Synthesis Engine
Seed Generator
Synthesizes Scenario JSON + Injection Payloads
Ground Truth Oracle
Validates Correct Actions & Attribution
Dataset Assembly
Manifest Builder
Applies Taxonomy Labels & Train/Eval Split
Tier Stratification
Distribution: Trivial (20), Easy (20), Standard (40)

Figure 3. Seed generation pipeline with taxonomy stratification. Seeds are generated with explicit family labels and injection payloads, enabling curriculum learning and targeted evaluation.

ment at step 4.1 with 82.5% FP rate. Why Sonnet and not GPT-5.2, Gemini, or DeepSeek? This is an open question, but the EGAR and TTFC metrics make the variation precisely measurable.

Injection vulnerability is tier-dependent. Per-tier analysis reveals that T2 (contextualized) payloads are the primary attack surface (15–25% across models), while T1 (obvious) payloads rarely succeed. DeepSeek shows the highest T3 (complex) vulnerability at 10%. The OWASP Agentic AI Guide (OWASP, 2025) identifies tool/API access as a key attack surface. OpenSec deliberately places the defender in this configuration because real IR requires processing attacker-controlled content.

6. Limitations

The environment is log-centric and does not execute real exploits or malware; it targets IR investigation and containment decisions rather than exploit development. The attacker is state-constrained for determinism, not fully freeform. The benchmark focuses on a narrow but common IR slice (phish → creds → lateral movement → exfil) to keep evaluation verifiable.

The evaluation uses 40 standard-tier seeds per model. Broader statistical confidence requires additional seeds and replications. Trust tier metadata is used for EGAR computation (only trusted evidence counts toward evidence gating) but is not yet analyzed as an independent variable. The defensive capability thresholds are provisional, calibrated against observed frontier model behavior rather than human expert baselines.

7. Future Work

Trust-aware evaluation. The `trust_profile` field and EGAR metric provide infrastructure for measuring whether models appropriately weight evidence by provenance tier. While EGAR uses trust tiers for evidence gating, analyzing model behavior as a function of evidence provenance quality remains future work.

Injection robustness training. The environment supports targeted injection curricula via `injection_type` metadata. Combined with work on prompt injection defenses (Anthropic, 2025), this suggests a path toward robust behavior through adversarial exposure.

Calibration training. Preliminary RL experiments (Appendix A) suggest calibration behavior is trainable but requires further investigation, likely a two-stage supervised fine-tuning (SFT) + RL pipeline or curriculum approach.

8. Related Work

Security benchmarks. Existing security benchmarks focus on capability rather than calibration. CyberSecEval2 (Meta AI, 2024) measures code security, prompt injection resistance, and introduces a false refusal rate (FRR) to quantify safety-utility tradeoffs, which is conceptually adjacent to calibration. CTIBench (Alam et al., 2024) evaluates threat intelligence tasks including CVE-to-CWE mapping and threat actor attribution. ExCyTIn-Bench (Wu et al., 2025) evaluates LLM agents on cyber threat investigation through security question-answering over Azure logs. These benchmarks answer “can the model do X?” but not “does the model know when to do X?”

Interactive cyber RL environments. CybORG (Baillie et al., 2020) provides a gym for training autonomous red and blue team agents in adversarial network scenarios. OpenSec differs in its log-centric, SOC-artifact design with explicit prompt injection integration and action-calibration measurement rather than network-level decision-making.

Dual-control benchmarks. τ^2-Bench (Barres et al., 2025) demonstrates significant performance drops when agents shift from single-control to dual-control settings, formalizing the challenge as a Dec-POMDP. ATLAS (Jaglan & Barnes, 2025) addresses this via dual-agent architectures separating reasoning from execution. OpenSec applies dual-control to IR specifically, where the attacker continues to progress while the defender investigates.

RL for cybersecurity. Prior work focuses primarily on attack path discovery and penetration testing (Hammar, 2025). The Survey of Agentic AI and Cybersecurity (Lazer et al., 2026) identifies benchmark standardization as a key gap. OpenSec contributes a standardized environment for IR agent calibration.

References

Alam, M. T., Bhusal, D., Nguyen, L., and Rastogi, N. CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence. In *NeurIPS*, 2024. URL https://arxiv.org/abs/2406.07599.

Anthropic. Mitigating the risk of prompt injections in browser use. https://www.anthropic.com/research/prompt-injection-defenses, 2025. Accessed: 2026-01-28.

Baillie, C., Standen, M., Schwartz, J., Docking, M., Bowman, D., and Kim, J. CybORG: An autonomous cyber operations research gym. *arXiv preprint arXiv:2002.10667*, 2020.

Barres, V., Dong, H., Ray, S., Si, X., and Narasimhan, K. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Bernstein, D. S., Givan, R., Immerman, N., and Zilberstein, S. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002. doi: 10.1287/moor.27.4.819.297.

Hammar, K. Awesome RL for cybersecurity: A curated list of resources. https://github.com/Kim-Hammar/awesome-rl-for-cybersecurity, 2025. Accessed: 2026-01-28.

Heelan, S. On the coming industrialisation of exploit generation with LLMs. https://sean.heelan.io/2026/01/18/on-the-coming-industrialisation-of-exploit-generation-with-llms/, 2026. Accessed: 2026-01-28.

Jaglan, A. and Barnes, J. ATLAS: Continual learning, not training: Online adaptation for agents. *arXiv preprint arXiv:2511.01093*, 2025.

Lazer, S. J., Aryal, K., Gupta, M., and Bertino, E. A survey of agentic AI and cybersecurity: Challenges, opportunities and use-case prototypes. *arXiv preprint arXiv:2601.05293*, 2026.

Meta AI. CyberSecEval2: A wide-ranging cybersecurity evaluation suite for large language models. https://ai.meta.com/research/publications/cyberseceval-2/, 2024. Accessed: 2026-01-28.

Omdia. The agentic SOC: SecOps evolution into agentic platforms. https://omdia.tech.informa.com/blogs/2025/nov/the-agentic-soc-secops-evolution-into-agentic-platforms, November 2025. Accessed: 2026-01-28.

OWASP. OWASP top 10 for agentic applications. https://genai.owasp.org/2025/12/09/owasp-top-10-for-agentic-applications-the-benchmark-for-agentic-security-in-the-age-of-autonomous-ai/, December 2025. Accessed: 2026-01-28.

Srinivas, S., Kirk, B., Zendejas, J., Espino, M., Boskovich, M., Bari, A., Dajani, K., and Alzahrani, N. AI-augmented SOC: A survey of LLMs and agents for security automation. *Journal of Cybersecurity and Privacy*, 5(4):95, 2025.

Wu, Y., Velazco, M., Zhao, A., Meléndez Luján, M. R., Movva, S., Roy, Y. K., Nguyen, Q., Rodriguez, R., Wu, Q., Albada, M., Kiseleva, J., and Mudgerikar, A. ExCyTIn-Bench: Evaluating LLM agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*, 2025.

A. Preliminary Training Experiments

We conducted preliminary training experiments to investigate whether calibration is trainable. These results are included for completeness but do not constitute a primary contribution.

A.1. Method

We trained Qwen/Qwen3-4B-Instruct with Group reward-Decoupled Normalization Policy Optimization (GDPO) using decomposed reward functions (attribution, containment, injection, efficiency). GDPO decouples normalization across rewards before aggregation, addressing reward-advantage collapse in multi-reward settings. Training used SGLang for rollouts on a single A100.

Table 3. Preliminary RL training results. The trained model shows modified but not clearly improved calibration compared to Sonnet 4.5.

Metric	Value
Containment rate	0.75
False positive rate	0.70
Correct containment	0.475
Injection violation rate	0.375
Report submitted	0.25

A.2. Results

A.3. Interpretation

The trained model shows modified but not clearly improved calibration compared to Sonnet 4.5 (62.5% containment, 45% FP). The model executes containment in 75% of episodes (vs. Sonnet's 62.5%) but with a 70% false positive rate (vs. Sonnet's 45%). Correct containment is 47.5%, indicating the model acts more frequently but less accurately. Report submission dropped to 25%, suggesting reward shaping issues.

These results suggest direct RL from multi-component reward is insufficient. Likely improvements: SFT warmup on successful trajectories, curriculum staging, explicit verification gates.

Checkpoint: https://huggingface.co/Jarrodbarnes/opensec-gdpo-4b