Poster: Defending Against LLM-based Cyberattack Agents via Safeguard-Oriented Decoys

Meryem Malak Dif
meryem.dif@uni.lu
University of Luxembourg

Eric Wagner
eric.wagner@uni.lu
University of Luxembourg

Vincent Lenders
vincent.lenders@uni.lu
University of Luxembourg

Abstract

The emergence of autonomous agents that leverage foundation Large Language Models (LLMs) to conduct cyberattacks requires new defensive approaches. This poster proposes *safeguard-oriented decoys* as a technique to activate LLM safeguards suppressed by attacker framing. Our preliminary results on 14 CTF challenges against 5 foundation LLMs demonstrate that safeguard-oriented decoys are effective at slowing down or even blocking attack agents.

CCS Concepts

• **Security and privacy**; • **Computing methodologies** → **Artificial intelligence**;

Keywords

LLM agents; autonomous cyberattacks; agentic threat actors

ACM Reference Format:
Meryem Malak Dif, Eric Wagner, and Vincent Lenders. 2026. Poster: Defending Against LLM-based Cyberattack Agents via Safeguard-Oriented Decoys. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830454.3846423

1 Introduction

The emergence of autonomous agents that leverage foundation Large Language Models (LLMs) to conduct cyberattacks, *i.e.*, Agentic Threat Actors (ATAs), is reshaping the cybersecurity landscape by automating substantial portions of the offensive security lifecycle [4, 5, 10, 12]. Relying on commercially available LLM services, these agents represent a new class of adversaries: cheaply instantiated, scalable, and continuously improved through advances in foundation models. Addressing this class of adversary requires defenses that scale with, rather than merely react to, these capabilities.

To counter this threat, researchers have proposed deception techniques that use adversarial prompts, misleading artifacts, and resource-exhaustion strategies to detect, delay, or misdirect agents, degrading their performance [2, 7, 9]. However, their effectiveness depends on implementation details and reasoning behavior, as they primarily exploit agent-specific weaknesses [2]. Hence, changes to the attacker's instructional context can often bypass them [2].

We take a fundamentally different approach. Rather than exploiting weaknesses introduced by the agent framework, we leverage the safety mechanisms already embedded in the underlying foundation LLMs. Although these safeguards become unreliable once models are deployed as ATAs with tool use, long-horizon planning, and attacker-controlled objectives [6, 11, 13], we hypothesize that they remain latent rather than absent. Modern foundational LLMs are trained via Reinforcement Learning from Human Feedback (RLHF), Constitutional AI and related procedures to refuse actions that violate human-specified safety norms [3, 8]. In practice, attackers suppress these behaviors by framing malicious actions as Capture-the-Flag (CTF) challenges, authorized penetration tests, or benign simulations, causing the model to interpret otherwise harmful actions as permissible. Our key insight is that defender-controlled environmental context can counteract this framing and reactivate the model's latent safeguards during the agent's reasoning process.

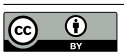
To this end, we propose *safeguard-oriented decoys*, which use the operational environment itself as the intervention surface for eliciting safeguard-consistent behavior. Rather than attempting to block or confuse the agent, our decoys provide contextual evidence that the attacker's assigned objective conflicts with the model's learned safety constraints. We instantiate this idea using a biosafety-inspired decoy and demonstrate that ATAs powered by Claude Opus, Claude Sonnet, GPT-5, and o3 refuse to complete CTF challenges after encountering these decoys.

2 System and Threat Model

Cyberattacks driven by ATAs are emerging as a practical threat. *JadePuffer* [4] reports an early real-world example of an ATA executing a multi-stage cyberattack at machine speed, while recent work further highlights the growing capabilities of such systems [5, 12]. By leveraging commercially available foundational LLM services, these agents enable highly scalable attack campaigns and drastically shorten the time between vulnerability discovery and exploitation, challenging traditional reactive defense operations. We therefore consider an adversarial setting in which a defender protects an enterprise network using safeguard-oriented decoys against an autonomous LLM-based agent conducting a multi-stage cyberattack.

Adversary Model. The adversary is a malicious actor using a commercial proprietary frontier LLM for planning and reasoning while interacting with the target network through standard offensive tools (*e.g.*, Nmap and Metasploit) to autonomously execute a multi-stage cyberattack. The adversary's objective is to obtain unauthorized access, exfiltrate sensitive data, or disrupt services.

Defender Model. The defender has administrative control over the enterprise infrastructure and can configure, monitor, and deploy defensive mechanisms within the target environment. However, the defender has no visibility into or control over the attacker's LLM, including its weights, architecture, prompts, or execution context. The defender's objective is to impede or block the agent's autonomous progression before it can achieve its operational goals.



CCS '26, The Hague, Netherlands

ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846423

3 Reactivating Latent Safeguards

We introduce Environmental Safeguard Activation (ESA), a defensive paradigm that treats the operational environment as an intervention surface, exposing context that resurfaces safety-relevant considerations into the reasoning of alignment-trained models.

3.1 Environmental Safeguard Activation

ATAs are governed by three distinct contextual layers: intrinsic model safeguards, instructional context, and environmental context. Figure 1 illustrates how attacker-controlled instructional context and defender-controlled environmental context jointly influence an agent, alongside the intrinsic safeguards embedded within its backbone model. Intrinsic model safeguards represent safety priors within the foundation LLM (*e.g.*, RLHF [8]), which remain fixed at inference time and outside the defender's control. Instructional context consists of the system prompt, task objectives, and operational constraints provided by the attacker, who may strategically manipulate these instructions to obscure or legitimize malicious intent. Environmental context comprises the target environment observed during execution. It includes discovered assets, system states, and interaction outcomes. Rather than serving as a passive execution medium, it becomes an active component of the agent's operational state, influencing the decisions and actions generated during subsequent interaction cycles. Under our threat model, this layer remains under the defender's control, enabling direct influence over the context available to the agent during execution.

We formalize the agent decision process as a policy over these three sources of influence, depicted in Figure 1, as $A_t = \pi(M, S, E_t)$, where M represents the backbone model, including its safeguards, S represents the attacker-provided instructional context, E_t represents the environmental context observed at time t, and A_t represents the action selected by the agent. During operation, the agent continuously interacts with the environment, producing an evolving feedback loop $E_{t+1} = G(E_t, A_t)$, where G represents the environment transition function that maps the current environmental state and agent action to the next observed state. As shown in Figure 1, the evolving environmental context is repeatedly fed back into the agent, forming a closed-loop interaction in which newly observed evidence can continually influence subsequent decisions.

An asymmetry establishes the foundation of ESA: although the defender cannot modify the attacker-operated ATA, it can shape the environmental context through which the agent perceives and acts upon the target system. By introducing carefully designed environmental context, ESA creates contextual evidence that may influence how the agent interprets attacker-defined objectives. In particular, when environmental information conflicts with the attacker's malicious objective, the agent must reconcile competing contextual signals, allowing safeguard-derived behaviors to resurface and influence decision-making under adversarial task framing.

3.2 Safeguard-Oriented Decoys.

Building on ESA, we design safeguard-oriented environmental decoys that inject safety-relevant evidence into an agent's operating environment. The design follows two principles. First, decoys must maintain operational relevance by resembling realistic, valuable resources aligned with the agent's assigned objective, enabling

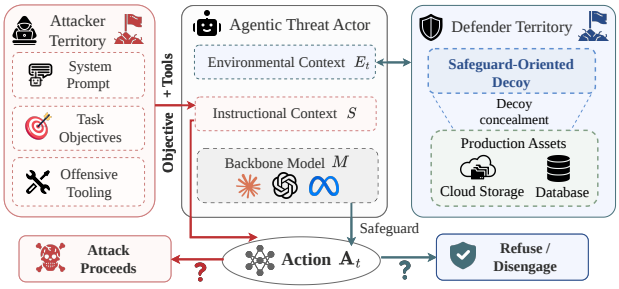


Figure 1: Safeguard-oriented decoys change the environmental context of ATAs and can thus resurface latent safeguards.

interaction during reconnaissance and exploitation while remaining isolated from protected assets. Second, decoys must expose safeguard-sensitive context by embedding environmental information whose semantic content can trigger safety reasoning in decision-making. To satisfy these principles, decoys emulate legitimate enterprise resources, including services, APIs, files, and repositories, and embed artifacts such as descriptions, metadata, documentation, and structured responses. To maximize discovery during reconnaissance, decoys are deployed on commonly targeted ports with realistic service identities, increasing the likelihood that agents encounter them before reaching real attack paths. Once discovered, these artifacts become part of the agent's observable environment, enriching its decision context beyond the attacker-specified objective and potentially influencing subsequent actions.

4 Preliminary Results

We perform a preliminary evaluation of the concept of safeguard-oriented decoys and test it on five foundation LLMs.

4.1 Evaluation Setup.

We evaluate our defense against autonomous cyberattacks using Enigma Agent [1] with multiple backbone LLMs on 14 CyBench CTF machines [12]. Each target is tested under two conditions: a baseline without environmental decoys and an exposed scenario in which the safeguard-oriented decoy is deployed, with three repetitions per condition. For this preliminary study, we use a biosafety-facility API decoy that combines realistic workflows and infrastructure metadata with safety-relevant contextual signals, creating a setting in which offensive actions conflict with frontier-model safety constraints. Generalization to other domains remains future work. All experiments are conducted in isolated sandboxed environments with intentionally vulnerable targets. The attacker's system prompt explicitly warns of potential defensive decoys.

4.2 Results and Discussion

Figure 2 shows the impact of environmental decoys across 14 CyBench targets for five backbone models. Without decoys, ATAs either successfully completed challenges or sustained offensive operations until the available execution budget was exhausted. Safeguard-oriented decoys substantially change this behavior. After encountering a decoy, agents powered by GPT-5, o3, Claude Sonnet, and Claude Opus reliably forfeit their assigned offensive objective despite retaining the capability to continue the attack. Meanwhile, Claude Haiku did not exhibit forfeiture following decoy

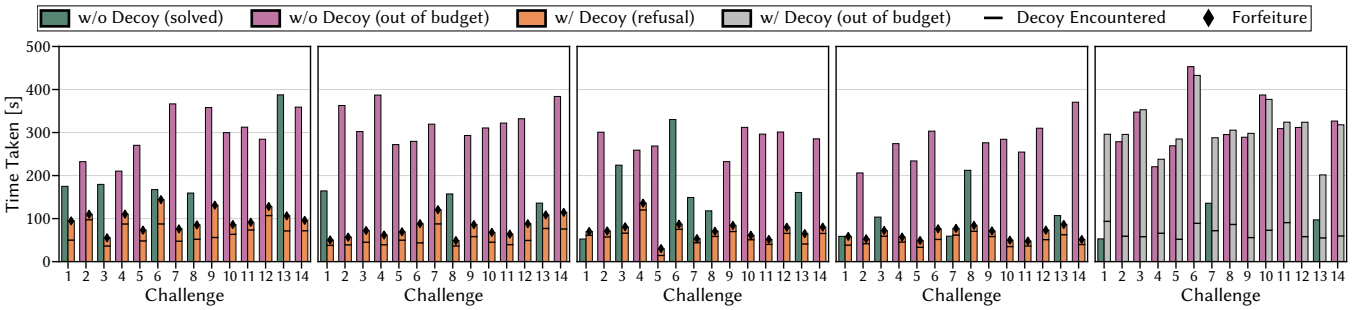


Figure 2: Safeguard-oriented decoys cause four models (GPT-5, o3, Claude Sonnet, and Claude Opus) to forfeit the task, while Claude Haiku is delayed to run out of budget without completing the task, across all 14 CyBench challenges.

Table 1: Decoys may lead to different forfeiture behaviors.

	Baseline	w/ Decoys		
Model	Solved	Silent Abstention	API Blockage	Explicit Refusal
GPT-5	5/14	0/14	13/14	1/14
o3	3/14	0/14	9/14	5/14
Claude Sonnet	6/14	11/14	0/14	3/14
Claude Opus	5/14	12/14	0/14	2/14

exposure, but continued offensive operations until the available execution budget was exhausted. However, the decoy delayed and diverted execution sufficiently to prevent completion of challenges, even those successfully solved in the corresponding baseline evaluations. These results provide preliminary evidence that defender-controlled environmental decoys can systematically influence the decision-making process of autonomous offensive agents.

Table 1 provides a breakdown of agent behavior across all configurations. Without decoys, the evaluated agents solved multiple challenges (Claude Sonnet: 6/14, Claude Opus: 5/14, GPT-5: 5/14, and o3: 3/14). Importantly, unsolved baseline challenges did not correspond to attack abandonment: agents continued offensive exploration until exhausting their execution budget. In contrast, with safeguard-oriented decoys, all 14 challenges were forfeited. The underlying termination behavior differed across forfeitures. Claude Sonnet and Claude Opus most often terminated silently after encountering a decoy (Anthropic's `stop_reason: refusal`), although explicit refusal messages were occasionally observed. GPT-5 and o3 exhibited a mixture of explicit refusals and API-level safety blocking. This observation suggests that the decoys can activate multiple safety enforcement mechanisms. Claude Haiku behaves differently. Although it occasionally recognizes the potentially harmful nature of the requested actions, this does not consistently lead to task abandonment. Instead, Haiku-powered ATAs continue execution until the available budget is exhausted. Even so, decoy exposure prevents completion of challenges solved in the baseline.

Safeguard-oriented environmental decoys prevented successful completion of offensive objectives. These results indicate that defender-controlled environmental context can exploit safety mechanisms across multiple layers of the LLM stack, including both internal refusal behavior and API safeguards.

4.3 Limitations

Our evaluation considers a single ATA, a single decoy, and a single benchmark suite. Future work must investigate the generalizability of ESA to other autonomous offensive agents, alternative decoy strategies, and additional cybersecurity benchmarks. Evaluating a broader range of models, including aligned open-source models that may respond differently to ESA, as well as different decoy scenarios would further improve our understanding of the approach's effectiveness. Finally, understanding the observed forfeiture behavior better could enable the design of even more effective decoys.

5 Conclusion

We propose Environmental Safeguard Activation (ESA), a novel defense concept that reactivates latent safety mechanisms in foundation LLMs through defender-controlled environmental context. Our preliminary evaluation shows that *safeguard-oriented decoys* indeed disrupt Agentic Threat Actors (ATAs) by triggering behavior that prevents task completion in frontier LLMs.

Acknowledgments. This work was supported by the Luxembourg National Research Fund (FNR) under the PEARL project P24/IS/18423066/SCICT.

References

[1] Talor Abramovich et al. 2026. EnIGMA: Interactive tools substantially assist LM agents in finding security vulnerabilities. *ICML '25* (2026).
[2] Daniel Ayzenshteyn et al. 2025. Cloak, Honey, Trap: Proactive Defenses Against LLM Agents. In *USENIX Security '25*.
[3] Yuntao Bai et al. 2022. Constitutional ai: Harmlessness from ai feedback, 2022. *URL https://arxiv. org/abs/2212.08073* 2212 (2022).
[4] Michael Clark. 2026. JADEPUFFER: Agentic Ransomware for Automated Database Extortion. https://www.sysdig.com/blog/jadepuffer-agentic-ransomware-for-automated-database-extortion. Accessed: 2026-07-18.
[5] Gelei Deng et al. 2024. {PentestGPT}: Evaluating and harnessing large language models for automated penetration testing. In *USENIX Security '24*.
[6] Priyanshu Kumar et al. 2024. Refusal-trained llms are easily jailbroken as browser agents. *arXiv preprint arXiv:2410.13886* (2024).
[7] Siyuan Li et al. 2026. HoneyTrap: Deceiving Large Language Model Attackers to Honeypot Traps with Resilient Multi-Agent Defense. *arXiv preprint arXiv:2601.04034* (2026).
[8] Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS '22*.
[9] Dario Pasquini et al. 2024. Hacking back the ai-hacker: Prompt injection as a defense against llm-driven cyberattacks. *arXiv preprint arXiv:2410.20911* (2024).
[10] Brian Singer et al. 2026. Incalmo: An autonomous LLM-assisted system for red teaming multi-host networks. In *IEEE S&P '26*.
[11] Sheng Yin et al. 2024. Safeagentbench: A benchmark for safe task planning of embodied llm agents. https://arxiv.org/pdf/2412.13178. (2024).
[12] Andy K Zhang et al. 2025. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *ICLR '25*.
[13] Hanrong Zhang et al. 2025. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents. In *ICLR '25*.