The Morphosis AI Platform: Scaling Autonomous Honeypot Networks

Daniel Reti
German Research Center for Artificial Intelligence (DFKI)
Intelligent Networks
Kaiserslautern, Germany
daniel.reti@dfki.de

Théo Lisart
German Research Center for Artificial Intelligence (DFKI)
Intelligent Networks
Kaiserslautern, Germany
theo.lisart@dfki.de

Abstract

Despite decades of research, cyber deception remains underutilized as an active defense layer. The core barrier is effort: crafting convincing honeypots and honeytokens that match an organization's real infrastructure demands significant manual work, limiting adoption especially among resource-constrained organizations. Recent advances in large language models (LLMs) offer an opportunity to close this gap. We introduce Morphosis AI, a platform concept for autonomous, adaptive cyber deception powered by generative AI. Morphosis AI integrates specialized LLM pipelines for generating deceptive artifacts—documents, credentials, configurations, and synthetic personas—with automated deployment of large-scale honeypot networks (honeyranges). We describe the platform architecture, formalize a four-stage generative deception pipeline from threat modeling to continuous adaptation, and pose five research questions that must be addressed to realize this vision: (1) designing LLM-enabled deception strategies, (2) achieving resource-efficient model specialization, (3) enabling bio-inspired honeypot evolution, (4) defining automation boundaries, and (5) measuring deception effectiveness against skilled human adversaries.

Keywords

cyber deception, honeypots, honeytokens, large language models, generative AI, proactive defense

1 Introduction

With the quantitative and qualitative intensification of global conflicts in an increasingly interconnected digital world, cyber threats have become ubiquitous. Private and public organizations face the risk of being targeted by Advanced Persistent Threats (APTs) or state-sponsored campaigns [12]. In practice, successful cyber attacks are often discovered only after the fact, when only reactive measures remain available. Conventional defense strategies—anomaly detection systems, signature-based intrusion detection, and traffic analysis tools—while functional, do not generalize well to novel attack vectors or unknown zero-day exploits.

Cyber deception offers a fundamentally different, proactive paradigm. Rather than attempting to detect attacks after they occur, deception-based defense deliberately misleads attackers by presenting them with fabricated systems, data, and services. This strategy targets the weakest element of the attack chain: human judgment. By deceiving attackers based on their motives, deception enables detection and intrusion control regardless of the novelty of the penetration strategy. The approach reverses the traditional asymmetry between attacker and defender: instead of the defender needing to protect every asset, a single interaction with a decoy can trigger an alert.

Despite its demonstrated effectiveness, cyber deception tends to be relegated to secondary early-warning systems rather than employed as an active defense layer [8]. The primary reason is that honeypot systems must be carefully tailored to the institution they protect, requiring significant manual effort to balance security interests, usability, and cost-effectiveness. Recent advances in large language models (LLMs) and generative AI present an opportunity to overcome these limitations. LLMs can generate large volumes of realistic, organization-specific decoys—documents, credentials, configuration files, database entries, and even synthetic personas—that match what an attacker would expect to find in a real infrastructure [14].

In this paper, we introduce **Morphosis AI**, a platform concept for autonomous, adaptive cyber deception powered by generative AI. We describe its dual-pipeline architecture, detail the generative deception pipeline from threat modeling through token generation to automated deployment, and outline the key research challenges that must be addressed to realize this vision.

2 Background

2.1 Cyber Deception

Bell and Whaley [1] defined a foundational taxonomy for deception comprising two principal classes: *dissimulation*, which concerns hiding the real (through masking, repackaging, or dazzling), and *simulation*, which concerns showing the false (through mimicking, inventing, or decoying). In the cybersecurity domain, this taxonomy manifests through honeypots, honeynets, and honeytokens. Honeypots are decoy systems designed to attract and monitor attackers; honeynets are networks of such systems; and honeytokens are individual data artifacts—files, credentials, database records—planted to detect unauthorized access [2].

A comprehensive meta-analysis of honeypot research by Javadpour et al. [8] identified several open challenges, including the need for AI-driven dynamic honeypot systems, platform-agnostic honeytoken generation, and systematic

evaluation of deployment methods. Current honeypot implementations are predominantly static: their content is configured once and does not evolve, making them susceptible to fingerprinting by experienced adversaries. Furthermore, the manual effort required to create convincing decoys limits the scalability of deception-based defense, particularly for small and medium-sized enterprises (SMEs) that lack dedicated security teams.

Prior work has explored generative deception in specific application domains. Cambiaso and Caviglione [3] demonstrated that ChatGPT can be used to automatically engage email scammers in realistic conversations, confirming that LLMs are effective tools for generating deceptive content. These results suggest that generative AI can bridge the gap between the manual effort currently required and the scale at which deception must operate to be effective.

2.2 LLMs for Security Applications

Applications of machine learning, particularly LLMs, have expanded rapidly across domains [16]. Two recent developments are especially relevant for automated cyber deception. First, advances in fine-tuning large open-source models [4, 13] enable the application of generative AI to specific problems without training from scratch—a process that is typically power-hungry, data-hungry, and expensive. Second, advances in model distillation and pruning [7] as well as privacy-preserving inference [9] allow for safe and economically viable deployment of generative AI in security-sensitive environments. These techniques enable specialized LLM pipelines to be executed on local GPU clusters, which is essential for applications that handle confidential infrastructure data.

In the specific intersection of LLMs and cyber deception, Reti et al. [14] systematically investigated honeytoken generation using LLMs, testing 210 prompt structures across multiple models to generate seven types of honeytokens including configuration files, databases, and log files. Their results showed that honeywords generated by GPT-3.5 were statistically less distinguishable from real passwords than those produced by previous automated methods. On the deployment side, Lian et al. [10] demonstrated that LLMs can serve as configuration validators, suggesting their potential for generating deployable honeypot configuration files. The controllability of such generation processes can be enhanced through Retrieval-Augmented Generation (RAG) [5], while cross-model communication [11] opens possibilities for coordinated behavior across multiple deployed honeypots.

3 Morphosis AI Platform

3.1 Concept and Threat Model

Morphosis AI targets scenarios where organizations face sophisticated, targeted attacks by adversaries who perform reconnaissance, lateral movement, and data exfiltration. The platform's deception strategy pursues three complementary objectives: (1) *early detection*—any interaction with a honeypot or honeytoken serves as an indicator of compromise and triggers an alert; (2) *derailment*—realistic decoy environments divert attackers from genuine assets, creating fear, uncertainty, and doubt when decoys are discovered, or undermining the attack when they are not; and (3) *data poisoning*—mixing real data with generated artifacts reduces the value of exfiltrated information, as attackers must expend significant effort to separate genuine from fabricated data.

Central to the platform is an *attention model* that formalizes what makes deception artifacts believable from the attacker's perspective. This model draws on the Bell-Whaley taxonomy to define whether a given deception strategy employs dissimulation or simulation, and maps honeytoken types to attacker expectations based on the threat model. A *honeytoken taxonomy* classifies the types of deceptive artifacts the platform can generate—including internal chat logs, database entries, financial documents, personal data, emails, technical documentation, configuration files, and reports—and ranks them by a cost-effectiveness analysis that projects their impact on attacker attention according to the attention model.

3.2 Architecture Overview

The platform is organized around two complementary pipelines, connected by a web-based frontend for monitoring and control (Figure 1).

LLMOps Pipeline. The *Honeyrange Configuration Generator* produces network topologies, service configurations, and deployment specifications for large-scale honeyranges that mirror an organization's real infrastructure. The *Honeytoken Generator* is a set of fine-tuned, distilled models [7, 13] specialized for producing deceptive artifacts—documents, credentials, configurations—validated against benchmarks for human-likeness [6].

DevOps Pipeline. The *Deployment Engine* provisions honeyranges as reproducible virtual environments using OpenTofu, libvirt, and Ansible. The *Honeypot API Control* module provides a RAG interface [5] for post-deployment refinement, real-time token generation, and cross-honeypot communication [11] that allows the fleet to collectively evolve. A web-based frontend connects both pipelines for events tracking, fleet control, and SIEM integration.

3.3 Generative Deception Pipeline

The end-to-end pipeline from organizational context to live deception environment proceeds in four stages (Figure **??**).

Stage 1: Threat and Attention Modeling. The operator defines the threat model—specifying the types of adversaries, attack scenarios, and assets to protect—and the deception strategy, selecting between dissimulation and simulation techniques. The attention model is instantiated based on these inputs, establishing which types of honeytokens will be most effective given the anticipated attacker profile.

Stage 2: Token Generation. Based on the honeytoken taxonomy and the attention model, the specialized Honeytoken Generators produce deceptive artifacts tailored to the target organization. This involves a hybrid approach of pre-generation and on-the-fly inference: bulk content such

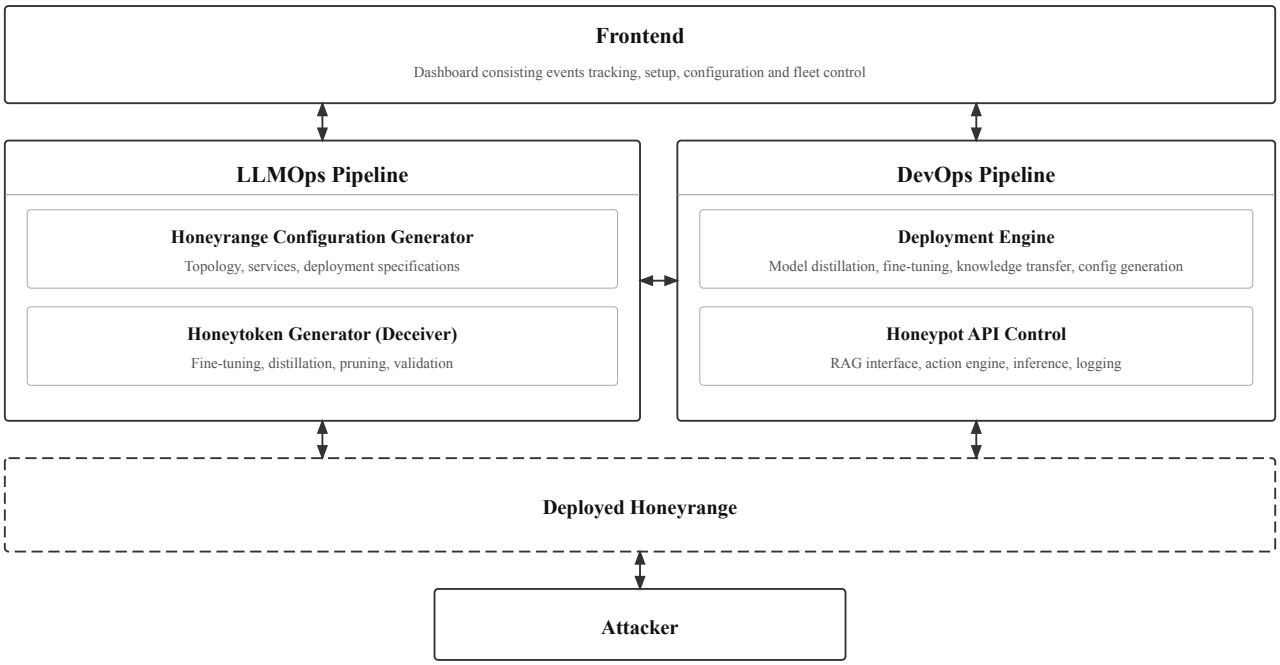

Figure 1: High-level architecture of the Morphosis AI platform. The *Frontend* (top) provides a dashboard for events tracking, setup, configuration, and fleet control. The *LLMOps* pipeline (bottom-left) encompasses the Honeyrange Configuration Generator and the Honeytoken Generator. The *DevOps* pipeline (bottom-right) handles deployment and runtime control of honeypots. Both pipelines communicate bidirectionally with the deployed honeyrange.

as documents and database entries is generated ahead of time, while interactive elements—such as responses to attacker queries in a honeypot shell—are generated in real time [14]. The generators are fine-tuned on a mixture of publicly available data and synthetic data crafted to match the organizational profile.

Stage 3: Configuration and Deployment. The Honeyrange Configuration Generator designs the honeyrange topology: which hosts to provision, which services to expose on each, which honeytokens to embed, how hosts relate to each other, and what data emission schedules to follow for autonomous file creation and network activity. The DevOps pipeline then provisions these specifications as a full virtual network environment via OpenTofu, libvirt, and Ansible, deploying the honeyrange into the target infrastructure.

Stage 4: Continuous Adaptation. Post-deployment, the RAG layer monitors attacker interactions and feeds this intelligence back into the generation pipeline. Honeytokens that fail to attract attention are replaced; behavioral patterns that trigger attacker suspicion are adjusted. Over time, the fleet of honeypots evolves in a bio-inspired fashion, with successful deception strategies propagated across instances.

4 Research Challenges

Realizing the Morphosis AI vision requires addressing several open research questions.

Deception Strategy Design. Given a threat model and an attacker attention model, which deception strategies and honeytoken types are newly enabled by LLM capabilities? The design space is vast—spanning documents, credentials, configurations, network services, and synthetic personas—and the optimal selection depends on the attacker profile, the organizational context, and the cost of generation and deployment.

Resource-Efficient Model Specialization. Can LLMs for deception be made significantly more resource-efficient through fine-tuning, pruning, and distillation without observable degradation in the quality of generated artifacts with respect to the threat model? Specifically, how small can a model be and still produce outputs that are indistinguishable from genuine infrastructure data to a motivated attacker? This trade-off between model size, inference cost, and deception quality is central to the economic viability of the platform.

Bio-Inspired Honeypot Evolution. Can deployed honeypots, equipped with RAG capabilities, communicate with each other and evolve in a bio-inspired fashion to create increasingly believable deception environments? This requires

cross-model communication protocols [11] and feedback mechanisms that propagate successful deception strategies while discarding ineffective ones.

Automation Boundaries. To what extent can the deployment of LLM-generated honeypots and honeytokens be automated, and at which stages of the pipeline is human feedback required? Full automation reduces operational cost but risks generating content that is ethically or legally problematic, particularly when producing synthetic personal data or imitating real individuals.

Effectiveness Measurement. With which metrics can the cognitive and behavioral impact of deception on human attackers be measured? Quantifying the effectiveness of cyber deception remains an open problem. Controlled empirical studies with professional penetration testers are needed to assess attacker attention, emotional response, and interaction patterns with generated decoys and distractions [6].

5 Conclusion and Outlook

We have presented the Morphosis AI platform, a concept for scaling autonomous honeypot networks through generative AI. By combining an LLMOps pipeline for specialized honeytoken generation with a DevOps pipeline for automated honeyrange deployment, the platform provides an end-to-end path from organizational threat modeling to live, adaptive deception environments—addressing the manual effort barrier that has limited the adoption of cyber deception as an active defense layer.

We identified five open research questions spanning deception strategy design, resource-efficient model specialization, bio-inspired honeypot evolution, automation boundaries, and empirical effectiveness measurement. The platform offers a sophisticated, AI-driven defense posture against advanced persistent threats, enabling organizations to deploy adaptive deception at scale as a fully integrated defense layer.

Acknowledgments

This work was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Morphosis AI project (grant ID 16KIS2393).

References

[1] John Bowyer Bell and Barton Whaley. 1991. *Cheating and Deception*. Transaction Publishers, New Brunswick, NJ, USA.

[2] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici. 2011. HoneyGen: An Automated Honeytokens Generator. In *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*. IEEE, Beijing, China, 131–136. doi:10.1109/ISI.2011.5984063

[3] Enrico Cambiaso and Luca Caviglione. 2023. Scamming the Scammers: Using ChatGPT to Reply Mails for Wasting Time and Resources. In *Proceedings of the Italian Conference on CyberSecurity (ITASEC 2023) (CEUR Workshop Proceedings, Vol. 3488)*. CEUR-WS.org, Bari, Italy, 97–106. arXiv:2303.13521 https://ceur-ws.org/Vol-3488/paper08.pdf

[4] Francisco Eiras, Aleksandar Petrov, Bertie Vidgen, Christian Schroeder, Fabio Pizzati, Katherine Elkins, Supratik Mukhopadhyay, Adel Bibi, Aaron Purewal, Csaba Botos, Fabro Steibel, Fazel Keshtkar, Fazl Barez, Genevieve Smith, Gianluca Guadagni, Jon Chun, Jordi Cabot, Joseph Imperial, Juan Arturo Nolazco, Lori Landay, Matthew Jackson, Phillip H. S. Torr, Trevor Darrell, Yong Lee, and Jakob Foerster. 2024. Risks and Opportunities of Open-Source Generative AI. arXiv:2405.08597 [cs.LG] https://arxiv.org/abs/2405.08597

[5] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. arXiv:2405.06211 [cs.CL] https://arxiv.org/abs/2405.06211

[6] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. 2023. Evaluating Large Language Models: A Comprehensive Survey. arXiv:2310.19736 [cs.CL] https://arxiv.org/abs/2310.19736

[7] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. arXiv:2305.02301 [cs.CL] https://arxiv.org/abs/2305.02301

[8] Amir Javadpour, Forough Ja'fari, Tarik Taleb, Mohammad Shojafar, and Chafika Benzaïd. 2024. A comprehensive survey on cyber deception techniques to improve honeypot performance. *Computers & Security* 140 (2024), 103792. doi:10.1016/j.cose.2024.103792

[9] Dacheng Li, Rulin Shao, Hongyi Wang, Han Guo, Eric P. Xing, and Hao Zhang. 2023. MPCFormer: Fast, Performant and Private Transformer Inference with MPC. arXiv:2211.01452 [cs.LG] https://arxiv.org/abs/2211.01452

[10] Xinyu Lian, Yinfang Chen, Runxiang Cheng, Jie Huang, Parth Thakkar, Minjia Zhang, and Tianyin Xu. 2025. Large Language Models as Configuration Validators. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE, Los Alamitos, CA, USA, 204–216. doi:10.1109/ICSE55347.2025.00017

[11] Yuhan Liu, Esha Choukse, Shan Lu, Junchen Jiang, and Madan Musuvathi. 2024. DroidSpeak: Enhancing Cross-LLM Communication. arXiv:2411.02820 [cs.MA] https://arxiv.org/abs/2411.02820

[12] Subash Neupane, Ivan A. Fernandez, Sudip Mittal, and Shahram Rahimi. 2023. Impacts and Risk of Generative AI Technology on Cyber Defense. arXiv:2306.13033 [cs.CR] https://arxiv.org/abs/2306.13033

[13] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities. arXiv:2408.13296 [cs.LG] https://arxiv.org/abs/2408.13296

[14] Daniel Reti, Norman Becker, Tillmann Angeli, Anasuya Chattopadhyay, Daniel Schneider, Sebastian Vollmer, and Hans D. Schotten. 2024. Act as a Honeytoken Generator! An Investigation into Honeytoken Generation with Large Language Models. In *Proceedings of the 11th ACM Workshop on Adaptive and Autonomous Cyber Defense (AACD '24)* (Salt Lake City, UT, USA). Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3689935.3690394

[15] Daniel Reti, Karina Elzer, and Hans Schotten. 2023. SCANTRAP: Protecting Content Management Systems from Vulnerability Scanners with Cyber Deception and Obfuscation. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP)*. SciTePress, Setúbal, Portugal, 485–492. doi:10.5220/0011667400003405

[16] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A Survey of Large Language Models. arXiv:2303.18223 [cs.CL] https://arxiv.org/abs/2303.18223