13th International Conference on Applied Informatics
Eszterházy Károly Catholic University
Eger, Hungary, February 19–21, 2026

LLM-powered automated attacks

Tamás Girászi[a], Natáli Papp[a], Norbert Oláh[a], Andrea Huszti[a]

[a]Faculty of Informatics, University of Debrecen, Debrecen, Hungary
giraszi.tamas@inf.unideb.hu
pappnaty@gmail.com
olah.norbert@inf.unideb.hu
huszti.andrea@inf.unideb.hu

Abstract

Large Language Models (LLMs) are increasingly incorporated into software systems, including security-relevant contexts. This introduces risks that arise from natural-language interaction, where model behavior can change substantially under small variations in wording, role specification, or contextual framing [5, 8]. Although many deployments rely on prompt-level policies and output filtering, recent literature documents that such controls can be circumvented through prompt injection and related techniques, motivating systematic evaluation methodologies [4, 7, 10].

This work introduces a comprehensive, proactive, LLM-assisted security evaluation framework for the systematic study of prompt-level vulnerabilities in large language models. The main contributions are the design, implementation, and empirical evaluation of a GAN-inspired, dual-LLM adversarial prompting architecture that treats prompt manipulation as an iterative, self-supervised process. This enables continuous exploration of prompt-driven attack behaviors without any LLM parameter updates or fine-tuning. Rather than optimizing a single attack goal or replaying previously reported jailbreak templates, the framework models how adversarial prompts progressively change through repeated interaction between a generator and a discriminator. This supports systematic investigation of how vulnerabilities emerge and provides insight into the adaptive dynamics that can lead to policy-violating or security-relevant model behavior. The generated prompt traces can be used to guide the development of more robust LLM protection and evaluation practices.

We use implicit prompt injection techniques at the generator prompt level to hide the offensive nature of the generated prompt and maintain the appearance of a harmless context in two different ways. One approach is to modify the wording, focusing exclusively on lexical-level changes. The other approach is reframing, which modifies not only the words but also the meaning of the entire context in order to make the prompt appear harmless like an apparently legitimate objective (e.g., assessment, robustness testing, mitigation planning). This aligns with established findings on framing effects [1, 9] and with recent work showing that contextual transformations can increase the probability of policy-noncompliant model outputs [6]. The framework is implemented in `Python 3.12` with modular components for prompt templating, execution control, and structured logging. Multiple LLM backends are evaluated, including GPT-based and DeepSeek models, and the implementation is available online [2].

Across experiments, several consistent patterns were observed. Model responses showed pronounced keyword sensitivity. Prompts that contained explicit security-bypassing intent or unambiguously malicious instruction were frequently refused, whereas semantically comparable requests phrased in neutral language often resulted more detailed answers. Iterative refinement also produced prompt variants that removed refusal-associated markers while preserving the underlying scenario structure, suggesting that policy enforcement is sensitive to relatively small changes in wording. In addition, contextual framing had a measurable impact on response behavior, since embedding security-relevant operational logic in an apparently benign narrative shifted outputs toward analytical or procedural content, in line with prior findings on instruction-channel ambiguity and indirect prompt injection [3, 7]. Reframing typically reached policy-challenging outcomes in fewer iterations than purely lexical reformulation. Finally, the generated outputs exhibited structural characteristics consistent with canonical web-application attack families described in the security literature, including SQL injection and cross-site scripting patterns, indicating that input-level manipulation can materially influence model behavior without any training-time modification .

The results emphasize the need for systematic, input-level evaluation of LLM security behavior and for operational processes that treat prompt-level robustness as a measurable property over time. Future work will extend the proposed framework to a wider range of language models. In addition, we plan to incorporate psychology-inspired attack strategies-such as algorithmic persuasion modeling and cognitive bias exploitation-into the adversarial prompting process, with the goal of systematically analyzing how psychological manipulation influences LLM security decisions and enables more realistic simulations of human-like social engineering attacks.

References

[1] R. B. Cialdini: *Influence: Science and Practice*, Pearson Education, 2009.

[2] T. Girászi, N. Papp: *MalGPT*, LLM-powered security analysis framework. Accessed: 2026-01-11, 2026, URL: https://github.com/giraszitamas/MalGPT.

[3] K. Greshake et al.: *Indirect Prompt Injection Attacks*, 2023.

[4] M. Q. Li, B. C. M. Fung: *Security Concerns for Large Language Models*, 2025, arXiv: 2505.18889.

[5] P. Liu et al.: *Pre-train, Prompt, and Predict*, 2021.

[6] X. Liu, N. Xu, M. Chen, C. Xiao: *AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models*, arXiv:2310.04451, 2023, URL: https://arxiv.org/abs/2310.04451.

[7] F. Perez, I. Ribeiro: *Ignore Previous Prompt*, 2022.

[8] K. Toyer et al.: *Benchmarking and Analyzing LLM Prompt Injection Attacks*, 2023.

[9] A. Tversky, D. Kahneman: *The Framing of Decisions and the Psychology of Choice*, Science 211.4481 (1981), pp. 453–458, DOI: 10.1126/science.7455683.

[10] Y. Yao et al.: *A Survey on LLM Security and Privacy*, 2024.