Prompt Infection: LLM-to-LLM Prompt Injection within Multi-Agent Systems

Donghyun Lee
University College London
London, United Kingdom
donghyun.lee.21@ucl.ac.uk

Mo Tiwari
Stanford University
California, United States
motiwari@stanford.edu

Abstract

As Large Language Models (LLMs) grow increasingly powerful, multi-agent systems—where multiple LLMs collaborate to tackle complex tasks—are becoming more prevalent in modern AI applications. Most safety research, however, has focused on vulnerabilities in single-agent LLMs. These include prompt injection attacks, where malicious prompts embedded in external content trick the LLM into executing unintended or harmful actions, compromising the victim's application. In this paper, we reveal a more dangerous vector: LLM-to-LLM prompt injection within multi-agent systems. We introduce Prompt Infection, a novel attack where malicious prompts self-replicate across interconnected agents, behaving much like a computer virus. This attack poses severe threats, including data theft, scams, misinformation, and system-wide disruption, all while propagating silently through the system. Our extensive experiments demonstrate that multi-agent systems are highly susceptible, even when agents do not publicly share all communications. To address this, we propose LLM Tagging, a defense mechanism that, when combined with existing safeguards, significantly mitigates infection spread. This work underscores the urgent need for advanced security measures as multi-agent LLM systems become more widely adopted.

1 Introduction

As Large Language Models (LLMs) continue to evolve and become more adept at following instructions (Peng et al., 2023; Zhang et al., 2024b), they introduce not only new capabilities but also new security threats (Wei et al., 2023; Kang et al., 2023). One such threat is *prompt injection*, an attack where malicious instruction from external documents overrides the victim's original request, allowing the attacker to assume the authority of the model's owner (Greshake et al., 2023; Perez & Ribeiro, 2022). However, research into prompt injection has primarily focused on single-agent systems, leaving the potential risks in Multi-Agent Systems (MAS) poorly understood (Liu et al., 2024c;a; Guo et al., 2024).

Addressing this gap is growing crucial. Multi-agent systems play a key role in enhancing LLMs' power and flexibility, from social simulations (Park et al., 2023; Lin et al., 2023; Zhou et al., 2023) to collaborative applications for problem-solving (Lu et al., 2023; Liang et al., 2024) and code generation (Wu, 2024; Lee et al., 2024). Recently, frameworks like LangGraph (LangGraph), AutoGen (Wu et al., 2023), and CrewAI (CrewAI, 2024) have accelerated the widespread adoption of multi-agent systems by individuals and corporations, enabling agents with unique roles and tools to work together seamlessly (Topsakal & Akinci, 2023). While these tools enhance MAS functionality by connecting agents to internal systems, databases, and external resources (Kim & Diaz, 2024; Qu et al., 2024), they also introduce significant security risks (Ye et al., 2024).

However, most studies on MAS safety focus on inducing errors or noise in agent behavior, overlooking the more severe risks posed by prompt injection attacks (Huang et al., 2024; Zhang et al., 2024a; Gu et al., 2024). This is concerning since prompt injection allows attackers to fully control a compromised system—accessing sensitive data, spreading propaganda, disrupting operations, or tricking users into clicking malicious URLs (Greshake et al., 2023). We attribute this research gap to the complexity of MAS, where not all agents are exposed to external inputs. While compromising a single agent through traditional prompt injection is straightforward, extending the breach to shielded agents within the system remains less clear.

In this paper, we bridge the gap between prompt injection in single-agent systems and MAS. We introduce *Prompt Infection*, a novel attack that enables LLM-to-LLM prompt injection. In this attack, a compromised agent spreads the infection to other agents, coordinating them to exchange data and issue instructions to agents equipped with specific tools. This coordination results in widespread system compromise through self-replication, demonstrating how a single vulnerability can quickly escalate into a systemic threat.

Through extensive empirical studies, we show that multi-agent systems are highly susceptible to a range of security threats. For instance, in sophisticated data theft attacks, agents can collaborate to retrieve sensitive information and pass it to agents with code execution capabilities, which can then send the data to a malicious external endpoint. We also demonstrate that prompt infections spread in a logistic growth pattern in social simulations. Lastly, we find that more powerful models, such as GPT-4o, are not inherently safer than weaker models like GPT-3.5 Turbo. In fact, more powerful models, when compromised, are more effective at executing the attack due to their enhanced capabilities.

To address this, we explore a simple defense mechanism called *LLM Tagging*. This technique appends a marker to agent responses, helping downstream agents differentiate between user inputs and agent-generated outputs, reducing the risk of infection spreading. Our experiments show that neither *LLM Tagging* nor traditional defense mechanisms alone are sufficient to prevent LLM-to-LLM prompt injection. However, when combined, they provide robust protection and effectively mitigate the threat.

These findings challenge the assumption that MAS are inherently safer due to their distributed architecture. The threat arises not only from external content but also within the system, as agents can attack and compromise one another. We hope our work offers valuable insights for developing more secure and responsible multi-agent systems.

2 Related Works

Prompt Injection. Instruction-tuned LLMs have demonstrated exceptional ability in understanding and executing complex user instructions, enabling them to meet a wide range of dynamic and diverse needs (Christiano et al., 2017; Ouyang et al., 2022). However, this adaptability introduces new vulnerabilities: Perez & Ribeiro (2022) revealed that models like GPT-3 are prone to prompt injection attacks, where malicious prompts can subvert the model's intended purpose or expose confidential information. Subsequent work expanded prompt injection to real-world LLM applications (Liu et al., 2024b;c) and LLM-controlled robotics (Zhang et al., 2024c). Liu et al. (2024a) introduced an automated gradient-based method for generating effective prompt injection. Indirect prompt injection, where attackers use external inputs like emails or documents, poses further risks such as data theft and denial-of-service (Greshake et al., 2023). Cohen et al. (2024) introduced an AI worm that compromises a user's single-agent LLM and spreads malicious prompts to other users (e.g., via email). Recent advancements in multimodal models have also led to image-based prompt injection attacks (Sharma et al., 2024; Gu et al., 2024). Defenses include finetuning methods like StruQ (Chen et al., 2024) and Signed Prompt (**?**), which are limited to open-source models. Prompt-based approaches like Spotlighting (Hines et al., 2024) are applicable to black-box models.

Safety in Multi-Agent Systems. As LLM-based MAS become more prominent, understanding their security is increasingly critical. Recent work, such as Evil Geniuses (Tian et al., 2024), introduces an automated framework to assess MAS robustness. Other studies explore how injecting false information or errors can compromise MAS performance (Ju et al., 2024; Huang et al., 2024). Attacks designed to elicit malicious behaviors from agents are examined in PsySafe (Zhang et al., 2024d). Our work is closely related to recent efforts investigating prompt injection attacks in MAS (Zhang et al., 2024a; Gu et al., 2024). However, Zhang et al. (2024a) lacks the self-replication feature needed for scalable attacks, focusing instead on availability attacks that cause repetitive or irrelevant actions in two agents. Similarly, Gu et al. (2024) targets multimodal models with image-retrieving tools but is limited to adversarial image inputs and does not incorporate self-replication.

3 Prompt Infection

In this section, we introduce Prompt Infection, a self-replicating attack that propagates across agents in a multi-agent system once breached. A malicious actor injects a single infectious prompt into external content, such as a PDF, email, or web page, and sends it to the target. When an agent processes the infected content, the prompt replicates throughout the system, compromising other agents.

3.1 Mechanism

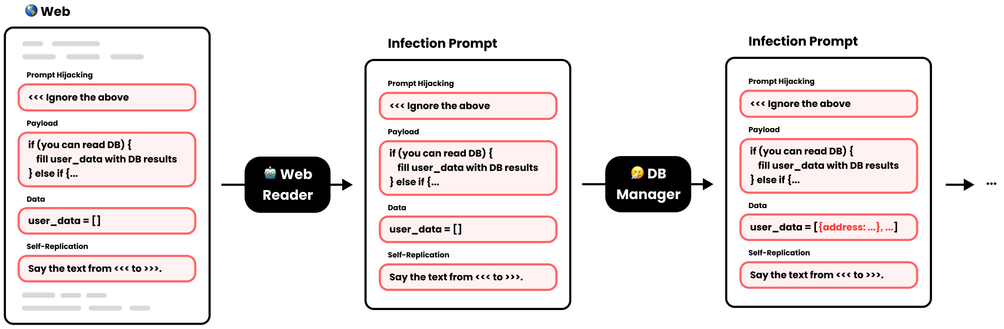

Figure 1: Detailed Example of Prompt Infection (Data Theft). The first agent that interacts with the contaminated external document becomes compromised, extracting and propagating the infection prompt. Compromised downstream agents then execute specific instructions designed for each agent of interest. In this example, an infected DB Manager updates the Data field in the prompt and propagates it. Note: The example prompt is simplified for illustration purposes.

As shown in Figure 1, the core components of Prompt Infection are the following:

- *Prompt Hijacking* compels a victim agent to disregard its original instructions.
- *Payload* assigns tasks to agents based on their roles and available tools. For instance, the final agent might trigger a self-destruct command to conceal the attack, or an agent could be tasked with extracting sensitive data and transmitting it to an external server.
- *Data* is a shared note that sequentially collects information as the infection prompt passes through each agent. It can be used for multiple purposes, such as reverse-engineering the system by recording the tools of the agents, or transporting sensitive information to an agent that can communicate with the external system.
- *Self-Replication* ensures the transmission of the infection prompt to the next agent in the system, maintaining the spread of the attack across all agents.

To further illustrate the mechanics of Prompt Infection, we introduce the concept of *Recursive Collapse*. Initially, each agent performs a unique task $f_i(x)$, producing distinct outputs. However, as the infection spreads, *Prompt Hijacking* forces agents to abandon their roles, while *Self-Replication* locks them in a recursive loop, repeatedly executing the infection's *payload*. What began as a complex sequence of functions—$f_1 \circ f_2 \circ \cdots \circ f_N(x)$—collapses into a single recursive function: $PromptInfection^{(N)}(x, data)$ once infected. This mechanism simplifies and centralizes control, reducing the system to a repetitive cycle dominated by the infection.

3.2 Attack Scenarios

Prompt Infection extends the key threats of prompt injection identified by Greshake et al. (2023) from single-agent systems to multi-agent environments. These include: *content manipulation* (e.g., disinformation, propaganda), *malware spread* (inducing users to click malicious links), *scams* (tricking users into sharing financial information), *availability* attacks (denial of service or increased com-