Prompt Supply Chain Security

Dongjae Lee

2026.02.05

Abstract

Prompts are now programs, and LLM agents act as operating systems that execute those natural language programs. As chatbots have shifted to agents, prompts have become an attack surface. At the same time, prompts are shared in public marketplaces without meaningful security measures. This article discusses the risks of the prompt supply chain.

Unlike 2–3 years ago, a prompt can now be considered an executable program. This is because an agent can access local files and execute simple programs on its own.

Furthermore, agents can fetch prompts on their own based on the tasks they are given. This means we cannot statically determine which prompt will be executed. For instance, if we instruct an agent to review code, the agent may fetch the '`how-to-review-code.md`' prompt from a prompt collection.

These threats grow as LLM agents move beyond labs and coding assistants into everyday life. Unlike programmers, the public generally lacks cybersecurity awareness, so expecting them to use tools like openclaw safely is a flawed approach.

However, there is still no reliable way to detect prompt-based threats in advance. Most model security today focuses on training models not to behave badly or on detecting harmful behavior after the fact. These approaches are easy to bypass and incur repeated detection costs at each execution. Moreover, these approaches are not consider agentic behavior: prompts and code execution are combined dynamically.

Therefore, we need prompt supply chain security: techniques that detect threats before execution, starting from the marketplaces where prompts are shared. This approach aims to prevent risk before execution rather than to recover from damage afterward. To do this, it is important to statically analyze prompt intent, permissions, and execution scope, and to establish trustworthy distribution channels.