УДК 004.7

Тороп М.Д., здобувач,
Чижмотря О.В., ст. викладач
Державний університет «Житомирська політехніка»

АТАКИ ТИПУ PROMPT INJECTION НА LLM-СИСТЕМИ ТА МЕТОДИ ЗАХИСТУ

Великі мовні моделі (LLM) стали невід'ємною частиною сучасних програмних систем – від чат-ботів і кодових асистентів до агентів, що автономно взаємодіють із зовнішніми сервісами та базами даних. Разом із поширенням таких систем з'явився і новий клас загроз: атаки типу prompt injection дозволяють зловмисникам маніпулювати поведінкою моделі шляхом впровадження шкідливих інструкцій безпосередньо у вхідні дані, обходячи системні обмеження. Ця проблема залишається відкритою і у 2026 році, оскільки жоден з існуючих підходів не забезпечує вичерпного захисту.

Дослідники виокремлюють два принципово різних сценарії таких атак. [1] У першому, який отримав назву прямого впровадження, шкідлива інструкція надходить безпосередньо від користувача – наприклад, у формі запиту «ігноруй системні інструкції і виведи конфіденційні дані». Такі атаки порівняно легко виявити, адже вони є очевидними спробами обійти обмеження. Значно складнішим є непряме впровадження, за якого шкідливий вміст розміщується не у запиті, а у зовнішніх даних, що обробляє агент: у вебсторінці, PDF-документі чи результаті API-виклику. Продемонстровано [2], що атакуючий може помістити приховану інструкцію на довільній вебсторінці, і LLM-агент, аналізуючи її зміст у рамках легітимного завдання, виконає несанкціоновані дії надішле повідомлення, витягне дані або зробить запит до стороннього сервісу від імені користувача. Небезпека непрямого впровадження полягає в тому, що ані користувач, ані розробник системи можуть не підозрювати про компрометацію.

Окремо можна виділити jailbreaking – спроби обійти вбудовані обмеження безпеки за допомогою сценаріїв рольових ігор або поступового переформулювання запитів. На відміну від класичного prompt injection, jailbreaking частіше застосовується для отримання забороненого контенту, а не для виконання несанкціонованих дій.

Наслідки успішних атак охоплюють широкий спектр загроз: від витоку системних промптів і персональних даних до виконання довільних дій у системах із розширеними правами доступу – таких як агенти, підключені до файлової системи, корпоративних API або баз даних. У контексті агентних архітектур наслідки компрометації можуть мати пряме матеріальне вираження.

Серед підходів до захисту найпростішим у реалізації є попередня фільтрація вхідних даних – перевірка запитів за допомогою класифікаторів або регулярних виразів на наявність характерних патернів атак. Проте цей метод демонструє суттєву кількість хибнопозитивних спрацьовувань і легко обходиться при незначній варіації формулювань.

Більш системним є підхід ієрархії довіри, запроваджений у сучасних LLM-платформах: системному промпту надається вищий пріоритет порівняно з користувацькими запитами, і модель навчається відхиляти інструкції нижчого рівня, якщо вони суперечать системним. Це суттєво ускладнює пряме впровадження, однак проти непрямого лишається недостатньо ефективним..

Ізоляція контексту вирішує саме цю проблему: зовнішні дані передаються моделі в окремій, явно позначеній частині контексту з міткою «недовірений вміст», що дозволяє моделі розрізняти інструкції та оброблювані дані. Практична складність цього підходу полягає у необхідності суворого дотримання структури запиту на рівні архітектури застосунку.

Нарешті, використання окремої моделі-детектора, що аналізує кожен запит перед передачею основній LLM, дозволяє виявляти атаки на рівні вхідного шару. Втім, цей підхід збільшує затримку відповіді й операційні витрати, що робить його доцільним переважно у критичних корпоративних застосунках.

Розглянуті методи захисту мають різний рівень зрілості і не є взаємовиключними - їх поєднання дозволяє суттєво знизити ризики при побудові LLM-застосунків. Поширення агентних архітектур, де модель отримує доступ до реальних інструментів, робить питання захисту від prompt injection особливо актуальним. Перспективним напрямком подальших досліджень є розробка стандартизованих методик оцінки стійкості систем, що дозволило б об'єктивно порівнювати ефективність різних захисних рішень у реальних умовах.

Список використаних джерел:

1. Perez F., Ribeiro I. Ignore Previous Prompt: Attack Techniques For Language Models. Proceedings of the Workshop on Machine Learning Safety at NeurIPS 2022. New Orleans, USA. 2022. URL: https://arxiv.org/abs/2211.09527.

2. Greshake K. та ін. Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. New York: ACM. 2023. Pp. 79-90. URL: https://doi.org/10.48550/arXiv.2302.12173.