ІН'ЄКЦІЯ ПРОМПТІВ ЯК АРХІТЕКТУРНА ЗАГРОЗА АГЕНТНИХ СИСТЕМ: КЛАСИФІКАЦІЯ КОНТРЗАХОДІВ ЗА РІВНЕМ ДІЇ

Киселевич Володимир Володимирович
асистент кафедри комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка

Надання агентові на основі великої мовної моделі (LLM) доступу до інструментів змінює характер загрози безпеці. Поки модель лише відповідає користувачеві, наслідком зловмисного впливу є хибний текст. Щойно агент дістає змогу надіслати запит, записати файл або викликати зовнішній програмний інтерфейс, той самий вплив перетворюється на несанкціоновану дію в реальній системі.

Причина цього має архітектурний характер. Вміст веб-сторінки, тіло документа, відповідь стороннього сервісу потрапляють у те саме контекстне вікно, що й інструкції розробника, і подаються моделі як однорідна послідовність токенів. Класичний принцип розділення коду й даних, на якому побудовано захист від ін'єкцій у традиційному програмному забезпеченні, тут не діє: у праці [1] прямо зазначено, що застосунки, інтегровані з LLM, розмивають межу між даними та інструкціями. Отже, вразливість закладено не в конкретній реалізації, а в самій організації взаємодії з моделлю.

Аналіз останніх досліджень підтверджує цю тезу з різних боків. У праці [1] уведено поняття непрямої ін'єкції промптів, коли зловмисник не взаємодіє із системою безпосередньо, а розміщує інструкцію в даних, які агент згодом отримає. У праці [2] на першому спеціалізованому наборі тестів показано, що наявні моделі вразливі до непрямих ін'єкцій універсально, а причинами названо нездатність моделі відрізнити інформаційний контекст від інструкції до виконання. У праці [3] запропоновано набір із 1054 тестових випадків, що охоплює 17 інструментів користувача та 62 інструменти зловмисника, спеціально для агентів з доступом до інструментів. У праці [4] побудовано формальну систему опису ін'єкційних атак і виконано зіставлення 5 атак та 10 захисних механізмів на 10 моделях і 7 задачах, тобто емпіричне порівняння захистів на спільному наборі вже здійснено. Оглядову картину загроз агентних екосистем подано в [5], де понад тридцять технік атак згруповано за чотирма категоріями із зазначенням рівнів, яких кожна торкається, а перетин великих мовних моделей і кібербезпеки систематизовано в [6].

Мета дослідження полягає в тому, щоб обґрунтувати архітектурну природу ін'єкції промптів і систематизувати контрзаходи за рівнем, на якому вони діють, із зазначенням загроз, що їх кожен рівень принципово не покриває.

Обмеженість фільтрації випливає з попереднього. Фільтрація вхідних даних намагається відокремити інструкцію від інформації за формою тексту, тобто

розв’язати задачу розпізнавання наміру в природній мові. Ця задача не має точного розв’язку: множина формулювань, що спонукають до дії, не є ані скінченною, ані замкненою щодо перефразування. Тому фільтр знижує ймовірність успішної атаки, проте не дає гарантії, і кожен новий спосіб формулювання потребує оновлення правил. Показово, що причиною вразливості в [2] названо саме нездатність відрізнити контекст від інструкції, а не недосконаліть конкретного фільтра.

Ознакою систематизації обрано рівень, на якому діє контрзахід. На відміну від [4], де захисти зіставлено емпірично за часткою відбитих атак, і від [5], де за рівнями впорядковано атаки, тут за рівнями впорядковано саме контрзаходи, причому для кожного рівня зазначено не виміряну ефективність, а клас загроз, який цей рівень не покриває принципово, тобто незалежно від якості реалізації. Ідея спирається на усталений у проєктуванні операційних систем принцип розмежування привілеїв: безпеку забезпечує не перевірка вмісту звернення, а неможливість виконати привілейовану операцію з непривілейованого контексту. За цією ознакою контрзаходи утворюють чотири рівні (таблиця 1).

Таблиця 1. Класифікація контрзаходів проти ін’єкції промптів за рівнем дії

№	Рівень	Механізми	Що рівень не покриває принципово
1	Рівень тексту	Фільтрація вхідних даних, виявлення інструкцій у вмісті, розмітка ненадійних фрагментів [4]	Перефразування та нові формулювання; межа інструкції й даних лишається семантичною [2]
2	Рівень моделі	Донавчання на стійкість, системні настанови ігнорувати вкладені інструкції [4]	Не дає гарантій: вихід лишається ймовірнісним, атака зводиться до пошуку обхідного формулювання
3	Рівень виконання	Моніторинг дій, політика втручання, підтвердження користувача перед незворотними діями	Охоплює лише передбачені політикою дії; не запобігає витоку даних дозволеними каналами
4	Рівень потоків	Розділення потоків керування й даних, видача повноважень на виклик інструментів [7; 8]	Потребує явної специфікації політики та звужує функційність системи

Практичне значення цього впорядкування полягає в тому, що рівні відрізняються не ступенем надійності, а самим типом твердження, яке вони дозволяють зробити. Рівні 1 і 2 дають лише ймовірнісне зниження ризику, оскільки покладаються на розпізнавання наміру. Рівні 3 і 4 дають перевірювані обмеження, проте лише в межах заданої політики. Принципово важливо, що рівні не є взаємозамінними: посилення фільтрації не компенсує відсутності розмежування потоків, бо ці механізми діють на різних рівнях і не покривають загроз один одного.

Найвищий рівень уже реалізовано в дослідницьких системах. У праці [7] запропоновано підхід, що відокремлює потік керування, здобутий із довіреного запиту, від потоку ненадійних даних, і застосовує повноваження для запобігання витоку приватних даних під час виклику інструментів. Заявлену стійкість здобуто ціною функційності: за наведеними авторами даними, система розв'язує 77 % завдань відповідного тестового набору проти 84 % у незахищеній конфігурації. У праці [8] узагальнено набір проєктних патернів побудови агентів, стійких до ін'єкції, і проаналізовано їхні компроміси між корисністю та безпекою. Обидві роботи підтверджують ключове положення: захист досягається зміною архітектури, а не вдосконаленням перевірки тексту.

Обмеження. Праці [7; 8] на момент підготовки роботи є препринтами, тому наведені числові показники слід трактувати як попередні. Запропонована класифікація є аналітичною систематизацією опублікованих механізмів, а не результатом власного експериментального порівняння; зіставлення рівнів за часткою відбитих атак виконано в [4] для механізмів переважно першого та другого рівнів, тимчасом як порівняння усіх чотирьох рівнів за єдиною методикою, зокрема з урахуванням втрати функційності, у розглянутих джерелах не виявлено. Крім того, віднесення конкретного механізму до рівня подекуди неоднозначне, оскільки реальні системи поєднують механізми різних рівнів.

Висновки. Обґрунтовано, що ін'єкція промптів є архітектурною, а не імплементаційною загрозою: вона виникає з подання інструкцій розробника та ненадійних даних як однорідної послідовності в спільному контекстному вікні. Показано, що захист фільтрацією принципово неповний, оскільки зводиться до розпізнавання наміру в природній мові. Запропоновано класифікацію контрзаходів за рівнем дії, яка для кожного з чотирьох рівнів зазначає клас непокритих загроз; ключовим є висновок про невзаємозамінність рівнів. Напрямом подальшої роботи є зіставлення всіх чотирьох рівнів за єдиною методикою з одночасним вимірюванням частки відбитих атак і втрати функційності.

Список літератури

1. Greshake K., Abdelnabi S., Mishra S., Endres C., Holz T., Fritz M. 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 (AISec '23). New York : ACM, 2023. P. 79–90. DOI: 10.1145/3605764.3623985.

2. Yi J., Xie Y., Zhu B., Kıcıman E., Sun G., Xie X., Wu F. Benchmarking and Defending against Indirect Prompt Injection Attacks on Large Language Models. Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York : ACM, 2025. P. 1809–1820. DOI: 10.1145/3690624.3709179.

3. Zhan Q., Liang Z., Ying Z., Kang D. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents. Findings of the

Association for Computational Linguistics: ACL 2024. Stroudsburg : ACL, 2024. P. 10471–10506. DOI: 10.18653/v1/2024.findings-acl.624.

4. Liu Y., Jia Y., Geng R., Jia J., Gong N. Z. Formalizing and Benchmarking Prompt Injection Attacks and Defenses. arXiv preprint arXiv:2310.12815. 2023. DOI: 10.48550/arXiv.2310.12815.

5. Ferrag M. A., Tihanyi N., Hamouda D., Maglaras L., Lakas A., Debbah M. From Prompt Injections to Protocol Exploits: Threats in LLM-Powered AI Agents Workflows. ICT Express. 2026. Vol. 12, No. 2. P. 353–383. DOI: 10.1016/j.icte.2025.12.001.

6. Zhang J., Bu H., Wen H., Liu Y., Fei H., Xi R. et al. When LLMs Meet Cybersecurity: A Systematic Literature Review. Cybersecurity. 2025. Vol. 8, No. 1. DOI: 10.1186/s42400-025-00361-w.

7. Debenedetti E., Shumailov I., Fan T., Hayes J., Carlini N., Fabian D., Kern C., Shi C., Terzis A., Tramèr F. Defeating Prompt Injections by Design. arXiv preprint arXiv:2503.18813. 2025. DOI: 10.48550/arXiv.2503.18813.

8. Beurer-Kellner L., Buesser B., Creţu A.-M., Debenedetti E., Dobos D., Fabian D. et al. Design Patterns for Securing LLM Agents against Prompt Injections. arXiv preprint arXiv:2506.08837. 2025. DOI: 10.48550/arXiv.2506.08837.