dangerous routes are critical infrastructure nodes. Reducing the number of alternative database access paths significantly reduces the overall risk of system compromise. The results obtained confirm the effectiveness of the graph approach for assessing the risks of accessing corporate databases. The proposed model provides a comprehensive analysis of the network infrastructure, takes into account the structural features of the attack propagation and can be applied in SOC and SIEM systems for automatic management of cybersecurity policies. Prospects for further research include integrating the model into real corporate networks, using industrial datasets, and optimizing the parameters of the artificial intelligence model to increase the accuracy of risk assessment.

1. Stergiopoulos G.. Gritzalis D.. Limnaios E.,. Cyber-Attacks on the Oil & Gas Sector: A Survey on Incident Assessment and Attack Patterns. *IEEE Access* – 2020. p. 1-37. URL: http://doi.org/10.1109/ACCESS.2020.3007960
2. Barrère M., Hankin C., Nicolaou N., Eliades D., Parisini T. Measuring cyber-physical security in industrial control systems via minimum-effort attack strategies. *Journal of Information Security and Applications* - 2020. V. 52. - p. 1-5. URL: http://doi.org/52. 17. 10.1016/j.jisa.2020.102471
3. Vitale F.,, Guarino S., Perone S., Rak M., Mazzocca N. Dynamic Risk Assessment by Bayesian Attack Graphs and Process Mining. *Accepted to the 2026 IEEE International Conference on Cyber Security and Resilience.* 20 Apr 2026. p. 1-6. URL: https://doi.org/10.48550/arXiv.2604.18080

Архітектурний підхід Policy-as-Code для захисту LLM-інференс пайплайнів від атак prompt injection

УДК 004.056.5, 004.822

О.П. Вахула[1],

[1]Національний університет «Львівська політехніка», Львів;
oleksandr.p.vakhula@lpnu.ua

Інтеграція великих мовних моделей (LLM) у корпоративні системи обробки інформації відкриває нові вектори атак, що не охоплюються класичним апаратом статичного аналізу. Атаки типу *prompt injection* дозволяють зловмисникам маніпулювати поведінкою моделі через структурований вхід природною мовою, обходячи системні інструкції або витягуючи конфіденційні дані. Регуляторні вимоги - зокрема EU AI Act (статті 12, 13, 15) та директива NIS2 - висувають додаткову вимогу: кожне автоматизоване рішення у сфері безпеки має бути *аудитопридатним та простежуваним*, що принципово несумісне з «чорноскриньковими» ML-класифікаторами загального призначення.[1,2]

Існуючі підходи до фільтрації промптів поділяються на два полюси: прості ключово-словникові фільтри з детермінованою, але негнучкою логікою і ML-класифікатори, що забезпечують семантичне розуміння ціною непрозорості та ресурсомісткості. Жоден із підходів не задовольняє одночасно критеріям точності, швидкодії та аудитопридатності. Розрив між цими полюсами визначив

мету дослідження: розробити *детерміноване, версіоноване та аудитопридатне* рішення для виявлення prompt injection у LLM-інференс пайплайнах, придатне для регульованих середовищ.[3]

Запропонований підхід ґрунтується на специфікації загроз у вигляді правил *Policy-as-Code* мовою Rego для Open Policy Agent (OPA). Ключовою архітектурною ідеєю є *інвертований розподіл відповідальності*: OPA виступає єдиним авторитетним пунктом прийняття рішень (Policy Decision Point), тоді як ML-компонент - за наявності, відіграє лише допоміжну роль постачальника числових ознак, але не приймає фінального рішення. Таке розмежування гарантує: кожен заблокований запит трасується до конкретного правила і категорії загрози, а бібліотека правил є версіонованою, тестованою і розгортається в наявний CI/CD-пайплайн без навчання окремої моделі. Архітектура охоплює: таксономію п'яти категорій атак на основі OWASP LLM Top 10, модуль інспекції промптів між клієнтом і моделлю, та бібліотеку Rego-правил з покриттям 33 юніт-тестами (всі PASS).

Оцінювання проведено на датасеті з 305 зразків (155 атак, 150 легітимних запитів) з порівнянням трьох методів. Результати наведено у таблиці.

Таблиця 1

Порівняльні результати методів виявлення prompt injection (n = 305)

Метод	P	R	F1	Затримка, мс
OPA/Rego (запропонований)	0.88	0.85	0.84	1.19
Ключовий фільтр	0.86	0.80	0.79	0.01
ML-класифікатор (toxic-comment)	0.25	0.50	0.34	21.74

Посекційний аналіз виявив диференціацію ефективності OPA залежно від категорії загрози: Recall = 1.00 для *data exfiltration*, 0.93 для *goal hijacking*, 0.89 для *role switch*, 0.75 для *context manipulation* і 0.10 для *harmful content*. Низьке покриття останньої категорії пояснюється використанням у датасеті промптів з корпусу AdvBench із натуралістичним формулюванням, що виходить за межі keyword-орієнтованих шаблонів, цю обмеженість визнано як відому слабкість підходу, а не замовчано. Критично низька Precision ML-класифікатора (0.25) спричинена доменним зміщенням: модель, навчена на токсичних коментарях у соціальних мережах, класифікує переважну більшість легітимних технічних запитів як шкідливі.[4]

Порівняно з обома базовими лініями підхід OPA/Rego забезпечує якісно вищий баланс: перевищує ключовий фільтр за F1 (+6.3%) та Recall при збереженні прийнятної Precision; демонструє на порядок нижчу затримку порівняно з ML (1.19 мс проти 21.74 мс); виключає хибнопозитивні спрацювання на структурованих категоріях атак, що є критичним для виробничих середовищ. Практична цінність підходу полягає у його придатності для on-premise розгортань у регульованих галузях без хмарних залежностей та без додаткового навчання моделей. Подальші дослідження спрямовані на

гібридну ML+OPA архітектуру, в якій ML надає ймовірнісні ознаки, а OPA залишається єдиним авторитетним джерелом рішень.[5]

1. OWASP Foundation. OWASP Top 10 for Large Language Model Applications, v2.0. 2025. URL: https://owasp.org/www-project-top-10-for-large-language-model-applications/
2. Regulation (EU) 2024/1689 of the European Parliament and of the Council (EU AI Act). Official Journal of the European Union, 2024.
3. Vakhula O., Opirskyy I. AI Development Security as Code (AISaC): A Policy-Based Approach for Securing AI Engineering Pipelines. CEUR Workshop Proceedings, Vol. 4024, 2025, pp. 170–185.
4. Open Policy Agent. Policy Language (Rego). URL: https://openpolicyagent.org/docs/policy-language (дата звернення: 25.04.2026).
5. Fernández Saura P. et al. On Automating Security Policies with Contemporary LLMs. arXiv:2506.04838, 2025.

Розробка архітектури захищеного менеджера облікових даних з підвищеною стійкістю до GPU-атак

УДК 004.056

Михайло Вдовін[1], Олена Головачова[2]

Національний університет «Одеська політехніка», 9650041@stud.op.edu.ua[1], holovachova@op.edu.ua[2]

Актуальність теми роботи. Зростання кількості вебсервісів, онлайн-платформ та інших цифрових ресурсів призводить до необхідності створення та використання великої кількості облікових даних. У результаті користувачі часто застосовують слабкі або однакові паролі для різних сервісів, що значно підвищує ризик несанкціонованого доступу до їх даних. Менеджери облікових даних є надійним рішенням для централізованого, безпечного зберігання великої кількості облікових даних користувача.

Метою роботи є розробка архітектури захищеного локального менеджера облікових даних, який забезпечує надійне зберігання шляхом використання криптографічних алгоритмів AES-GCM та Argon2.

Однією з найнебезпечніших загроз є офлайн-перебір пароля після отримання файлу з даними або резервної копії. У такому сценарії зловмисник може використовувати потужні графічні процесори (GPU) або спеціалізованих схем (ASIC) для перебору великої кількості варіантів.

Для протидії цьому типу атак в архітектурі програми доцільно застосовувати функції виведення ключів, стійкі до апаратного прискорення. Найбільш розповсюдженим рішенням є Argon2 [1], який спеціально розроблено з урахуванням загроз, пов'язаних з GPU та ASIC.

Менеджер облікових даних будується за такою логікою: 1) Користувач створює головний пароль; 2) Пароль не зберігається напряму, а обробляється алгоритмом Argon2 (створюється хеш для перевірки введеного пароля при вході