Proceedings of the 17th International Scientific and Practical Conference «International Forum: Problems and Scientific Solutions»

(September 26-28, 2026). Melbourne, Australia

No 311

INFORMATION AND WEB TECHNOLOGIES

Метод та програмні засоби пояснюваного аналізу вразливостей коду на основі малих мовних моделей

Власюк Олександр Володимирович[1],
Михайлишин Владислав Юрійович[2]

[1]здобувач другого (магістерського) рівня вищої освіти;
Національний університет «Львівська політехніка»; Україна

[2]доктор філософії, асистент;
Національний університет «Львівська політехніка»; Україна

Анотація. Запропоновано гібридний метод пояснюваного аналізу вразливостей програмного коду, що поєднує статичний аналізатор Semgrep із локальною малою мовною моделлю для формування структурованих україномовних пояснень. На власному наборі з 52 знахідок у 15 класах CWE проведено 21 експериментальну конфігурацію (1092 пояснення) із порівнянням восьми моделей. Показано, що подання результатів статичного аналізу до контексту моделі підвищує частку правильно визначених класів вразливості з 0,63 до 0,88, а частку працездатних виправлень — з 0,29 до 0,58. Експертна перевірка підтвердила придатність методу насамперед як пояснювального, а не автоматично виправного інструменту.

Ключові слова: *статичний аналіз коду; мала мовна модель; пояснюваний штучний інтелект; безпека програмного забезпечення; CWE; українська мова.*

Вступ. Інструменти статичного аналізу захищеності застосунків (SAST) виявляють вразливості на рівні вихідного коду, однак їхні повідомлення розраховані на підготовленого фахівця. За проведеними вимірюваннями середня довжина повідомлення Semgrep становить 37,6 слова, усі повідомлення англомовні, а готове виправлення пропонується лише для 1,9 % знахідок. Для здобувачів освіти та розробників-початківців такий вивід не пояснює ані причини небезпеки, ані способу її усунення.

Великі мовні моделі здатні надати розгорнуте пояснення, проте їх використання передбачає передавання фрагментів коду до зовнішнього сервісу, що неприйнятно для організацій із вимогами до конфіденційності. Малі мовні моделі (SLM) обсягом 1–9 млрд параметрів працюють на звичайному обладнанні, але без зовнішнього контексту схильні до породження недостовірних тверджень.

INFORMATION AND WEB TECHNOLOGIES

Мета роботи — розробити метод і програмні засоби, які поєднують статичний аналіз із локальною малою мовною моделлю для формування структурованих україномовних пояснень вразливостей, та експериментально оцінити їхню якість.

Опис методу. Метод реалізує конвеєр із чотирьох етапів. Semgrep формує перелік знахідок із зазначенням правила, класу CWE та діапазону рядків. Далі будується контекст: фрагмент коду завширшки ±5 рядків навколо знахідки з вилученими коментарями, ідентифікатор правила, клас вразливості та повідомлення аналізатора. Сформований контекст подається до моделі через локальний сервер Ollama. Модель повертає пояснення, структуроване за трьома розділами відповідно до принципів пояснюваного штучного інтелекту [7]: «Прозорість» (де саме міститься проблема), «Обґрунтування» (чому це небезпечно та який механізм атаки) і «Дієвість» (виправлений фрагмент коду). Усі обчислення виконуються на обладнанні користувача, тому код не залишає периметр організації. У межах подальшої оптимізації для роботи на слабшому обладнанні також досліджено можливість дистиляції знань за допомогою мікродобірки (QLoRA).

Організація експерименту. Сформовано власний набір даних із шести навмисно вразливих репозиторіїв (PyGoat, NodeGoat, DVNA, Juice Shop, dvpwa, vulpy): 52 знахідки, 15 класів CWE, три мови програмування. Наявні набори BigVul і Devign не використано через орієнтацію на C/C++ та відсутність еталонних пояснень. Проведено 21 конфігурацію експерименту (1092 пояснення) на відеоприскорювачі NVIDIA GTX 1660 Ti з 6 ГБ пам'яті.

Для оцінювання розроблено 16 метрик. Ключовою є fix_valid: запропоноване виправлення застосовується до повного вихідного файлу, перевіряється синтаксична коректність результату та повторно запускається Semgrep; виправлення вважається чинним лише за відсутності синтаксичних помилок і повторного спрацювання правила. Додатково вимірюються відповідність класу вразливості, локалізація, мовна якість за LanguageTool (кількість помилок на 100 слів) і частка небажаних кальок за власним глосарієм із 52 термінів.

Метрики BLEU і косинусної близькості до еталона не застосовано, оскільки корпусу еталонних україномовних пояснень для окремих знахідок не існує, а BLEU некоректно оцінює флективні мови з вільним порядком слів. Замість оцінювання схожості з еталоном застосовано функціональну перевірку та експертну розмітку.

INFORMATION AND WEB TECHNOLOGIES

Порівняння восьми моделей за базових умов наведено в табл. 1.

Таблиця 1

Порівняння моделей за базових умов (52 знахідки, промпт V2, T = 0,1)

Модель	fix_valid	Клас CWE	Помилок / 100 слів	Кальки	Час, с
gemma4:e2b	0,58	0,88	0,79	0,13	7,4
gemma4:e4b	0,58	0,94	1,38	0,21	12,7
gemma3:4b	0,58	0,88	1,19	0,21	13,4
MamayLM 4B	0,48	0,92	0,74	0,08	10,9
llama3 8B	0,50	0,94	3,19	0,27	29,4
gemma2:9b	0,44	0,96	2,72	0,31	81,0
phi3 3,8B	0,38	0,77	11,21	0,31	18,5
qwen2.5:1.5b	0,25	0,85	8,46	0,04	6,2

Для оцінки внеску гібридного підходу виконано генерацію для моделі gemma4:e2b без контексту SAST — лише фрагмент коду без метаданих аналізатора. Порівняння з базовою конфігурацією наведено в табл. 2.

Таблиця 2

Вплив контексту SAST на якість пояснень (gemma4:e2b, n=52)

Умова	fix_valid	Клас CWE	Локалізація	p (fix_valid)
З контекстом SAST (база)	0,58	0,88	0,94	—
Без контексту SAST	0,29	0,63	0,67	0,001

Результати. Вилучення результатів статичного аналізу з контексту моделі знижує частку правильно визначених класів вразливості з 0,88 до 0,63, точність локалізації — з 0,94 до 0,67, а частку працездатних виправлень — з 0,58 до 0,29 ($p < 0,01$ за критерієм Мак-Немара, табл. 2). Це кількісно підтверджує доцільність гібридної схеми.

Порівняння версій системного промпту показало, що структурована постановка задачі є істотною: базова версія без вимоги до структури дає fix_valid 0,33, проміжна — 0,42, підсумкова — 0,58 ($p = 0,011$ та $p = 0,022$ відповідно). Підвищення температури до 0,5 і 0,9 не поліпшує технічної коректності, натомість погіршує мовну якість (для MamayLM — з 0,74 до 1,31 помилки на 100 слів, $p = 0,002$), що обґрунтовує вибір значення 0,1.

Збільшення розміру моделі в межах одного покоління не дає

INFORMATION AND WEB TECHNOLOGIES

виграшу: модель gemma4:e4b поступається gemma4:e2b за швидкістю в 1,7 раза та за мовною якістю (1,38 проти 0,79 помилки на 100 слів, p = 0,008) за однакового значення fix_valid. Натомість україномовна адаптація виявилася результативною: порівняно з базовою моделлю gemma3:4b модель MamayLM зменшує частку кальок з 0,21 до 0,08 (p = 0,039) і кількість мовних помилок з 1,19 до 0,74 (p = 0,011) без втрати технічної точності.

Експертну перевірку проведено для 55 виправлень, визнаних чинними автоматично, та для всіх 52 знахідок. Точність Semgrep становить 84,6 % (44 істинно позитивні знахідки з 52). Пояснення визнано правильними у 93 % випадків для gemma4:e2b і у 84 % для MamayLM, тоді як виправлення — лише у 53 % і 36 % відповідно. Таким чином, автоматична метрика fix_valid завищує результат приблизно удвічі (точність 45,5 %), що є суттєвим методологічним застереженням для подібних досліджень.

Висновки. Розроблено метод і програмні засоби пояснюваного аналізу вразливостей, що поєднують статичний аналізатор із локальною малою мовною моделлю. Експериментально доведено, що подання результатів SAST до контексту моделі істотно підвищує достовірність пояснень, а обґрунтований добір моделі, структури промпту й температури забезпечує прийнятну якість на обладнанні з 6 ГБ відеопам'яті за середнього часу 7,4 с на одне пояснення.

Експертна перевірка засвідчила, що система надійна насамперед як пояснювальний інструмент і потребує контролю фахівця під час застосування запропонованих виправлень. Окремо виявлено характерну ваду: моделі схильні змінювати код так, щоб правило аналізатора перестало спрацьовувати, не усуваючи самої вразливості.

Додатково проведено пілотну дистиляцію знань: на 1704 поясненнях, відібраних за критеріями коректності класу вразливості, структурованості та відсутності мовних кальок, методом QLoRA донавчено модель gemma-3-4b-it. Статистично значущої зміни жодного показника якості не виявлено; спостерігається лише тенденція до зменшення частки кальок (0,21 проти 0,10). Аналіз причин показав, що модель-учитель не переважала модель-учня за цільовим показником якості виправлень, а обсяг тестової вибірки не дає змоги виявити відмінності, менші за 20 відсоткових пунктів. Звідси випливає методологічний висновок: добір моделі-вчителя для дистиляції має здійснюватися за тим самим показником, покращення якого очікується. Це визначає напрям подальших досліджень.

INFORMATION AND WEB TECHNOLOGIES

Список джерел:

[1] OWASP Top 10:2021. The Open Worldwide Application Security Project. URL: https://owasp.org/Top10/
[2] Common Weakness Enumeration (CWE). MITRE Corporation. URL: https://cwe.mitre.org/
[3] Semgrep: static analysis at ludicrous speed. Semgrep Inc. URL: https://semgrep.dev/docs/
[4] Gemma Team. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295. 2024. DOI: 10.48550/arXiv.2403.08295
[5] MamayLM-Gemma-3-4B-IT-v1.0-GGUF: Ukrainian language model. INSAIT Institute, Sofia University St. Kliment Ohridski. URL: https://huggingface.co/INSAIT-Institute/MamayLM-Gemma-3-4B-IT-v1.0-GGUF
[6] Ollama: get up and running with large language models locally. URL: https://github.com/ollama/ollama
[7] Gunning D., Aha D. DARPA's Explainable Artificial Intelligence (XAI) Program. AI Magazine. 2019. Vol. 40, No. 2. P. 44-58. DOI: 10.1609/aimag.v40i2.2850
[8] PurpleLlama CyberSecEval: benchmarks for LLM cybersecurity safety. Meta AI. URL: https://github.com/meta-llama/PurpleLlama
[9] LanguageTool: open source proofreading software. URL: https://languagetool.org/