подальших досліджень є інтеграція мультимодальних LLM для аналізу зображень і відео, а також розробка модуля класифікації токсичного контенту.

1. EUvsDisinfo. How Russian Special Information Operations Try to Undermine Mobilisation in Ukraine. — 2024. URL: https://euvsdisinfo.eu/how-russian-special-information-operations-try-to-undermine-mobilisation-in-ukraine (дата звернення: 05.05.2026).
2. Kuperstein L. M., Lukichov V. V., Radetska A. O., Dudatyev A. V. System for Organizing Cyber Operations in the Context of Military Aggression // Science and Innovation. — 2025. — Vol. 21, № 3. — P. 86–98. https://doi.org/10.15407/scine21.03.086 (дата звернення: 05.05.2026).
3. Freedom House. Ukraine: Freedom on the Net 2023. — 2023. URL: https://freedomhouse.org/country/ukraine/freedom-net/2023 (дата звернення: 05.05.2026).
4. Baryshev Y., Kupershtein L., Maidanovych V., Voitovych O., Prokopenko S. Information System for the Fact-checker Support // CEUR Workshop Proceedings. — 2023. — Vol. 3646. — P. 127–138. URL: https://ceur-ws.org/Vol-3646/Paper_13.pdf (дата звернення: 05.05.2026).
5. Google. Gemini API Reference. — 2024. URL: https://ai.google.dev/gemini-api/docs (дата звернення: 05.05.2026).

Аналіз безпеки serverless-архітектур на основі моделювання подій

УДК 004.056:004.738.5 (043.2)

Петро Венгерський[1], Святослав Златоус[2]

Львівський національний університет імені Івана Франка,
[1]petro.venherskyy@lnu.edu.ua, [2]sviatoslav.zlatous@lnu.edu.ua

Сучасний розвиток хмарних обчислень зумовив широке впровадження serverless-архітектур, що базуються на концепції Function-as-a-Service (FaaS). Такий підхід дозволяє створювати масштабовані програмні системи без необхідності управління серверною інфраструктурою, що значно спрощує процес розробки та експлуатації додатків [1]. Водночас використання serverless-архітектур супроводжується появою нових викликів у сфері кібербезпеки, які пов'язані з динамічністю виконання функцій, складною структурою взаємодій між компонентами системи та обмеженим контролем користувачів над середовищем виконання [3]

У сучасних дослідженнях безпеки serverless-систем основна увага приділяється окремим аспектам їх функціонування, зокрема ізоляції функцій, управлінню доступом та аналізу продуктивності [3]. Водночас питання комплексного аналізу поведінки системи, що враховує взаємодії між функціями, подіями та сервісами хмарної інфраструктури, залишається недостатньо дослідженим [2]. Це ускладнює виявлення потенційних загроз безпеці, які можуть виникати у результаті нетипових сценаріїв функціонування системи.

У роботі запропоновано підхід до аналізу безпеки serverless-архітектур, що базується на дослідженні подій та взаємодій між компонентами системи. Основна ідея підходу полягає у використанні централізованого моніторингу

подій для формування моделі поведінки системи та виявлення відхилень від нормального режиму її функціонування.

Для перевірки ефективності запропонованого підходу було реалізовано експериментальне serverless-середовище у хмарній платформі AWS. Моніторинг функціонування системи здійснювався за допомогою сервісу AWS CloudWatch, який забезпечує збір та аналіз метрик і журналів подій.

Ingress Function Metrics (метрики інгрес лямбда-функції)

	Min	Max	Sum	Average (середнє)
Invocations (виклики)	23	23	23	23
Duration (тривалість)	821мс	821мс	821мс	821мс
Errors (помилки)	0	0	0	0
Throttles (тротлінг)	0	0	0	0

Processing Function Metrics (метрики процесінг лямбда-функції)

	Min	Max	Sum	Average (середнє)
Invocations (виклики)	15	127	339	56.5
Duration (тривалість)	4.81с	41.3с	116с	19.3с
Errors (помилки)	0	0	0	0

Consumer Function Metrics (метрики консюмер лямбда-функції)

	Min	Max	Sum	Average (середнє)
Invocations (виклики)	10	53	364	30.3
Duration (тривалість)	633мс	4.28с	28.2с	2.35с
Errors (помилки)	0	0	0	0

SQS Queue Metrics (метрики SQS черги)

	Min	Max	Sum	Average (середнє)
Messages sent (відправленні повідомлення)	0	127	340	0.39
Messages received (отримані повідомлення)	0	142	874	1.01
Messages deleted (видалені повідомлення)	0	0	0	0

DynamoDB Metrics (метрики DynamoDB бази данних)

	Min	Max	Sum	Average (середнє)
Consumed Read Capacity Units (Використані одиниці пропускної здатності читання)	0	0	0	0
Consumed Write Capacity Units (Використані одиниці пропускної здатності запису)	0	0.29	0.29	0
Put Item Successful Request Latency (Затримка успішного виконання операції запису)	32.2мс	32.2мс	32.2мс	32.2мс

Рис.1. Метрики функціонування serverless-системи у середовищі AWS CloudWatch

На рис. 1 наведено приклад панелі моніторингу AWS CloudWatch, що використовується для аналізу поведінки serverless-системи.

У процесі експерименту було досліджено поведінку системи у нормальному та аномальному режимах функціонування. У нормальному режимі середній час виконання функцій становив 120–250 мс, а частота викликів мала стабільний характер із незначними коливаннями. Структура взаємодій між компонентами залишалася сталою та відповідала визначеному сценарію обробки запитів.

Для моделювання аномалій було виконано штучне збільшення частоти викликів функцій у 2–3 рази, а також змінено послідовність взаємодій між компонентами системи. У результаті зафіксовано збільшення середнього часу виконання до 300–450 мс та появу нетипових послідовностей викликів функцій.

Аналіз отриманих результатів показав, що запропонований підхід дозволяє ефективно виявляти відхилення у функціонуванні serverless-систем. Зокрема, зміни у частоті викликів функцій, часі їх виконання та структурі взаємодій можуть бути використані як індикатори потенційних загроз безпеці.

1. Baldini I., Castro P., Chang K., et al. *Serverless computing: Current trends and open problems* // Research Advances in Cloud Computing. – 2017. – С. 1–20.
2. Jonas E., Schleier-Smith J., Sreekanti V., et al. *Cloud programming simplified: A Berkeley view on serverless computing* // arXiv:1902.03383. – 2019. – С. 1–28.
3. Zhang Y., Chen X., Li J. *Security and privacy in serverless computing: A systematic literature review* // ACM Computing Surveys. – 2023. – Vol. 55, No. 12. – С. 1–36.