УДК 004.056:004.738.5 *http://dx.doi.org/10.30970/vam.2026.36.00000*

АНАЛІЗ БЕЗПЕКИ SERVERLESS-АРХІТЕКТУР НА ОСНОВІ ПОДІЙ

П. Венгерський, С. Златоус

Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79000, Україна,
e-mail: *petro.venherskyy@lnu.edu.ua*, *sviatoslav.zlatous@lnu.edu.ua*

У роботі проведено аналіз сучасних досліджень, присвячених безпеці serverless-архітектур та платформ Function-as-a-Service (FaaS) з урахуванням особливостей взаємодії між компонентами системи та подієво-орієнтованого характеру їх функціонування. Актуальність теми обумовлена стрімким поширенням serverless-підходу у хмарних обчисленнях, що дає змогу створювати масштабовані додатки без необхідності управління серверною інфраструктурою. Попри значні переваги цієї моделі, використання serverless-архітектур супроводжується появою нових загроз безпеці, пов'язаних із динамічністю виконання функцій, складністю взаємодії між компонентами хмарної інфраструктури та обмеженим контролем користувачів над обчислювальним середовищем.

Метою роботи є аналіз існуючих досліджень у сфері безпеки serverless-архітектур із фокусом на дослідженні взаємодій між функціями, подіями та сервісами хмарної інфраструктури, визначення ключових проблем, які залишаються недостатньо дослідженими, а також формування підходів до їх вирішення. У роботі проаналізовано наукові публікації, присвячені безпеці FaaS-платформ, зокрема дослідження, що розглядають можливі вектори атак serverless-систем, механізми захисту під час виконання функцій та проблеми управління безпекою.

У результаті дослідження встановлено, що сучасні підходи до забезпечення безпеки serverless-систем переважно зосереджені на окремих компонентах архітектури, тоді як системний аналіз взаємодій між функціями, подіями та хмарними сервісами залишається недостатньо розробленим. Запропоновано концептуальний підхід до підвищення безпеки serverless-систем, що базується на аналізі послідовностей подій, використанні методів поведінкового аналізу, централізованого моніторингу подій та автоматизованого виявлення аномалій.

Ключові слова: serverless-обчислення, Function-as-a-Service, serverless-архітектура, безпека хмарних обчислень, кібербезпека, моніторинг подій, аналіз поверхні атак, виявлення аномалій.

1. Вступ

Розвиток хмарних обчислень призвів до появи нових моделей побудови програмних систем, що забезпечують ефективне використання обчислювальних ресурсів та спрощують розробку масштабованих застосунків. Одним із сучасних напрямів розвитку хмарних технологій є serverless-обчислення, у яких прикладна логіка реалізується у вигляді функцій, що виконуються у відповідь на події, тоді як управління інфраструктурою здійснюється постачальником хмарних сервісів [1]- [3]. Такий підхід дає змогу розробникам зосередитися на реалізації прикладної логіки програмного забезпечення без необхідності адміністрування серверної інфраструктури.

Serverless-архітектури, що реалізуються на основі платформ Function-as-a-Service, активно використовуються у сучасних хмарних системах [3], [5]. У таких архітектурах виконання програмного коду відбувається у вигляді незалежних функцій, які

активуються подіями та взаємодіють із різними компонентами хмарної інфраструктури, такими як системи зберігання даних, черги повідомлень або API-шлюзи [8], [10]. Подієво-орієнтована модель виконання функцій забезпечує гнучкість системи та дає змогу масштабувати обчислювальні ресурси відповідно до навантаження.

Разом з тим використання serverless-архітектур створює нові виклики у сфері кібербезпеки. Динамічний характер виконання функцій, розподілена структура системи та складна взаємодія між компонентами хмарної інфраструктури ускладнюють аналіз функціонування serverless-додатків і контроль їх безпеки [6], [7]. Крім того, обмежений доступ користувачів до середовища виконання функцій ускладнює проведення аудиту безпеки та дослідження інцидентів у хмарному середовищі [9], [11].

У сучасних наукових дослідженнях розглядаються різні аспекти функціонування serverless-платформ, зокрема архітектурні особливості, механізми виконання функцій та підходи до управління доступом до ресурсів хмарної інфраструктури [4], [12], [17]. Водночас більшість досліджень зосереджується на окремих компонентах serverless-систем, тоді як комплексний аналіз взаємодій між функціями, подіями та сервісами хмарної інфраструктури у контексті забезпечення безпеки представлений у літературі обмежено.

У зв'язку з цим метою роботи є аналіз сучасних досліджень у сфері serverless-архітектур та визначення основних проблем забезпечення безпеки таких систем.

2. Архітектурні особливості serverless-систем

Serverless-обчислення є одним із сучасних напрямів розвитку хмарних технологій, що передбачає виконання програмного коду у вигляді функцій, які активуються подіями та виконуються у керованому середовищі хмарного провайдера. У сучасних дослідженнях детально розглядаються принципи функціонування serverless-архітектур та їх роль у побудові масштабованих розподілених систем [1], [2].

Показано, що serverless-платформи характеризуються подієво-орієнтованою моделлю виконання функцій, автоматичним масштабуванням обчислювальних ресурсів та інтеграцією з різними сервісами хмарної інфраструктури [3], [5]. У таких системах обробка запитів користувачів реалізується через виконання незалежних функцій, які можуть взаємодіяти з компонентами хмарного середовища, зокрема базами даних, системами зберігання даних або сервісами обробки повідомлень [8], [10].

У науковій літературі також аналізуються архітектурні підходи до проектування serverless-додатків та організації взаємодії між функціями і сервісами хмарної інфраструктури [7], [16]. Показано, що взаємодія між функціями та іншими компонентами системи формує складну структуру виконання операцій у serverless-додатках.

Разом з тим більшість існуючих досліджень зосереджується переважно на архітектурних та технологічних аспектах serverless-платформ. Питання аналізу взаємодій між компонентами serverless-систем у контексті забезпечення безпеки розглядаються у цих роботах обмежено.

3. Короткий огляд сучасних досліджень

Окремий напрям досліджень присвячений аналізу проблем безпеки serverless-систем. У сучасній науковій літературі розглядаються різні аспекти забезпечення

безпеки таких архітектур, зокрема питання ізоляції виконання функцій, управління доступом до ресурсів та захисту хмарної інфраструктури [6], [13].

У serverless-середовищах функції виконуються у динамічно створюваних середовищах виконання, що може створювати додаткові виклики для забезпечення безпеки. Зокрема, досліджуються ризики, пов'язані з використанням програмних залежностей та сторонніх бібліотек у serverless-функціях [13], [18].

Крім того, у наукових роботах аналізуються загрози, пов'язані з взаємодією serverless-функцій із різними сервісами хмарної інфраструктури. Така взаємодія може створювати складні ланцюги виконання операцій, що ускладнює аналіз функціонування системи та контроль доступу до ресурсів [14], [19].

Разом з тим більшість існуючих досліджень зосереджується на окремих аспектах забезпечення безпеки serverless-платформ. Комплексний аналіз взаємодій між компонентами serverless-архітектур у контексті забезпечення безпеки розглядається у літературі обмежено.

Важливим аспектом дослідження serverless-архітектур є аналіз взаємодій між компонентами системи під час виконання функцій. У serverless-додатках виконання одного запиту може ініціювати послідовність викликів функцій, які взаємодіють із різними сервісами хмарної інфраструктури [7].

У результаті формується складна структура взаємодій між компонентами системи, що відображає процес виконання операцій у serverless-додатку. Аналіз таких взаємодій є важливим для дослідження функціонування serverless-систем та забезпечення їх безпеки.

У сучасних дослідженнях також розглядаються підходи до моніторингу подій, що виникають під час виконання serverless-функцій. Моніторинг подій дає змогу відстежувати послідовності виконання операцій у системі та аналізувати взаємодію між її компонентами [17], [18].

Водночас існуючі підходи до моніторингу serverless-систем мають обмеження, пов'язані з короткотривалістю виконання функцій та динамічним створенням середовищ виконання. Це ускладнює отримання повної інформації про структуру взаємодій між компонентами serverless-додатків.

Таким чином, проведений аналіз наукових досліджень показує, що питання аналізу взаємодій між функціями, подіями та сервісами хмарної інфраструктури потребують подальшого дослідження.

4. Методологія дослідження

Проведений аналіз наукових джерел показує, що сучасні дослідження безпеки serverless-архітектур здебільшого зосереджуються на окремих аспектах функціонування таких систем, зокрема на ізоляції функцій, управлінні доступом або аналізі продуктивності. Водночас комплексне дослідження поведінки serverless-додатків та взаємодії між їхніми компонентами залишається недостатньо розвиненим напрямом. З огляду на це виникає потреба у розробці методології дослідження, яка дозволить систематично аналізувати безпеку serverless-інфраструктур.

Методологія подальшого дослідження повинна базуватися на комплексному аналізі подій, що виникають у процесі функціонування serverless-додатків. Особливістю serverless-систем є те, що виконання функцій відбувається у відповідь на певні події, які можуть генеруватися користувачами, іншими сервісами або компонентами хмарної інфраструктури. Таким чином, аналіз потоків подій може надати важливу інформацію щодо поведінки системи та потенційних загроз безпеці.

Концептуальну схему взаємодії подій та компонентів serverless-системи, що використовується як основа запропонованої методології, наведено на рис. 1.

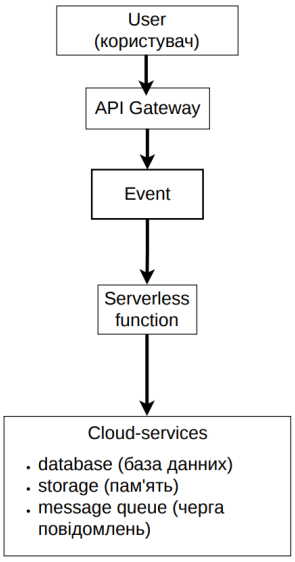

Рис. 1. Концептуальна архітектура serverless-системи

Як показано на рис. 1, виконання serverless-функцій ініціюється подіями, що виникають у результаті запитів користувачів або взаємодії з іншими сервісами хмарної інфраструктури. У процесі виконання функції можуть взаємодіяти з різними компонентами системи, зокрема базами даних, системами зберігання даних або сервісами обробки повідомлень. Аналіз таких взаємодій дає змогу досліджувати структуру виконання операцій у serverless-додатку та визначати потенційні точки виникнення загроз безпеці.

Першим етапом методології є збір та агрегація подій, що виникають у serverless-інфраструктурі. Джерелами таких подій можуть бути API-шлюзи, системи зберігання даних, черги повідомлень та інші хмарні сервіси. У процесі дослідження необхідно формувати централізоване сховище журналів подій, яке дозволить здійснювати подальший аналіз поведінки системи.

Наступним етапом є аналіз структури взаємодії між компонентами serverless-додатку. У межах цього етапу доцільно досліджувати, які функції викликають одна одну, які сервіси використовуються для обробки даних та які події ініціюють виконання функцій. Такий аналіз дає змогу сформувати модель взаємодії компонентів системи.

Важливим напрямом подальших досліджень є використання методів поведінкового аналізу для виявлення аномалій у роботі serverless-систем. Зокрема, можна досліджувати частоту викликів функцій, структуру подій та характер взаємодії між сервісами. Відхилення від типових сценаріїв роботи системи можуть свідчити про потенційні атаки або зловмисну активність.

Ще одним етапом методології є дослідження залежностей програмного забезпечення, що використовується у serverless-функціях. Оскільки такі функції часто використовують сторонні бібліотеки, важливо оцінювати їхню безпечність та потен-

ційні уразливості. Аналіз залежностей дає змогу виявляти ризики, пов'язані з використанням ненадійних або застарілих компонентів програмного забезпечення.

Завершальним етапом методології є оцінка ефективності механізмів безпеки serverless-платформ. У межах цього етапу доцільно проводити експериментальні дослідження, які дозволяють оцінити здатність системи виявляти та запобігати різним типам атак. Такі дослідження можуть включати моделювання атак, аналіз поведінки системи під час навантаження та оцінку ефективності механізмів моніторингу.

5. Проблеми та перспективи безпекових рішень у serverless-системах

Активне використання serverless-архітектур у сучасних хмарних середовищах створює нові виклики у сфері забезпечення кібербезпеки. Особливості організації serverless-систем, зокрема подієво-орієнтована модель виконання функцій, динамічне створення середовищ виконання та інтеграція з різними сервісами хмарної інфраструктури, формують складну структуру взаємодій між компонентами системи. У результаті забезпечення безпеки serverless-додатків потребує врахування не лише окремих компонентів системи, але й аналізу взаємодій між ними.

Однією з ключових проблем є обмежений рівень контролю користувачів над середовищем виконання функцій. У serverless-платформах управління обчислювальною інфраструктурою здійснюється постачальником хмарних сервісів, що обмежує можливості користувачів щодо контролю конфігурації системи та аналізу її функціонування. У таких умовах дослідження безпеки системи значною мірою базується на аналізі журналів подій та взаємодій між компонентами хмарної інфраструктури.

Іншою важливою проблемою є складність аналізу взаємодій між функціями та сервісами хмарної інфраструктури. У serverless-додатках виконання одного запиту користувача може ініціювати послідовність викликів функцій, які взаємодіють із різними компонентами системи. У результаті формується складна структура взаємодій між компонентами serverless-додатка, що ускладнює аналіз функціонування системи та виявлення потенційних загроз безпеці.

Окрему увагу слід приділити використанню програмних залежностей у serverless-функціях. Під час розробки serverless-додатків часто використовуються сторонні бібліотеки та програмні компоненти. Наявність уразливостей у таких компонентах може створювати додаткові ризики для безпеки системи. У зв'язку з цим важливим напрямом досліджень є аналіз залежностей програмного забезпечення та оцінка їхнього впливу на безпеку serverless-додатків.

Значною проблемою є також моніторинг функціонування serverless-систем. Через короткотривалість виконання функцій та динамічне створення середовищ виконання традиційні інструменти моніторингу не завжди дозволяють отримати повну інформацію про послідовність виконання операцій у системі. У результаті ускладнюється аналіз взаємодій між компонентами serverless-додатків та своєчасне виявлення аномалій у їх функціонуванні.

Одним із перспективних підходів до підвищення безпеки serverless-систем є використання систем централізованого моніторингу подій, які дозволяють здійснювати збір, зберігання та аналіз інформації про події, що виникають під час виконання функцій. Концептуальну схему системи моніторингу подій у serverless-архітектурі

наведено на рис. 2.

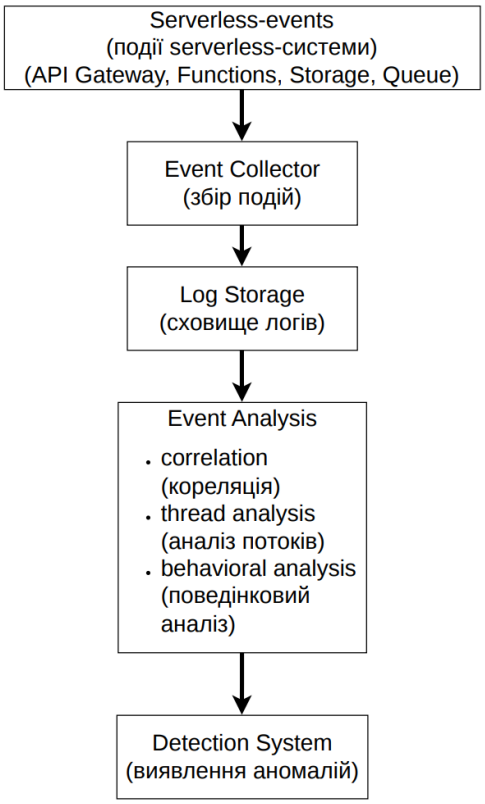

Рис. 2. Система централізованого моніторингу подій у serverless-архітектурі

Як показано на рис. 2, система моніторингу подій передбачає централізований збір інформації про події, що виникають у різних компонентах serverless-інфраструктури. Отримані дані можуть зберігатися у спеціалізованих сховищах журналів подій та використовуватися для подальшого аналізу. Аналіз потоків подій дає змогу досліджувати взаємодії між функціями, сервісами хмарної інфраструктури та іншими компонентами системи.

Використання централізованого моніторингу подій створює передумови для застосування методів поведінкового аналізу з метою виявлення аномалій у функціонуванні serverless-систем. Зокрема, можна досліджувати частоту викликів функцій, структуру подій та характер взаємодії між сервісами. Виявлення відхилень від типових сценаріїв функціонування системи може свідчити про потенційні інциденти безпеки або спроби несанкціонованого доступу.

Таким чином, забезпечення безпеки serverless-архітектур потребує комплексного підходу, що включає аналіз взаємодій між компонентами системи, дослідження потоків подій та використання механізмів централізованого моніторингу. Використання таких підходів дає змогу підвищити рівень спостережуваності serverless-систем та створює передумови для розробки ефективних механізмів забезпечення їх безпеки.

6. Експериментальна апробація запропонованої методології

Для підтвердження можливості практичного застосування запропонованої методології було розглянуто спрощений сценарій її реалізації у хмарному середовищі. Як тестове середовище обрано платформу AWS, яка надає засоби для розгортання serverless-додатків на основі сервісу AWS Lambda.

У межах експерименту було змодельовано набір serverless-функцій, що імітують обробку запитів користувачів. Зокрема, реалізовано ланцюг взаємодій між функціями, де виклик однієї функції ініціює виконання наступної, а також взаємодію з додатковими сервісами, такими як сховище даних та черги повідомлень. Загальну структуру експериментального середовища наведено на рис. 3.

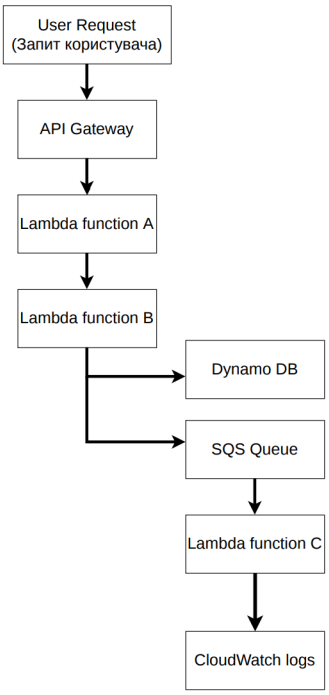

Рис. 3. Архітектура експериментального serverless-середовища (AWS Lambda)

Як показано на рис. 3, експериментальна система включає кілька взаємодіючих serverless-функцій та базові сервіси хмарної інфраструктури, що дає змогу моделювати типові сценарії обробки запитів у serverless-додатках.

Усі події, пов'язані з виконанням функцій, фіксувалися за допомогою механізмів журналювання AWS CloudWatch. На першому етапі здійснювався збір та агрегація даних про події, включаючи інформацію про виклики функцій, час виконання, частоту викликів та послідовність взаємодій між компонентами системи.

На рис. 4 наведено приклад панелі моніторингу AWS CloudWatch, яка відображає основні метрики функціонування serverless-системи, зокрема кількість викликів

функцій, тривалість виконання, наявність помилок, а також характеристики обміну повідомленнями через черги SQS і доступу до бази даних DynamoDB.

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 повідомелння)	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. 4. Метрики функціонування serverless-системи у середовищі AWS CloudWatch

Як видно з рис. 4, у нормальному режимі функціонування система характеризується стабільними значеннями метрик, зокрема рівномірною частотою викликів функцій і відносно невеликим часом виконання.

У ході експерименту було встановлено, що у нормальному режимі функціонування середній час виконання функцій становить 120-250 мс, а частота викликів окремих функцій має стабільний характер із незначними коливаннями. Структура взаємодій між компонентами системи при цьому залишається сталою та відповідає визначеному сценарію обробки запитів.

Для оцінки можливості виявлення аномалій було змодельовано відхилення від нормального режиму функціонування системи. Зокрема, виконано штучне збільшення частоти викликів окремих функцій у 2-3 рази, а також змінено послідовність взаємодій між компонентами системи. У результаті зафіксовано зміну характеристик подій, зокрема збільшення середнього часу виконання до 300-450 мс та появу нетипових послідовностей викликів функцій.

Як відображено на рис. 4, у цьому випадку спостерігається зростання тривалості виконання функцій, збільшення кількості повідомлень у чергах та зміна характеру навантаження на систему.

Проведений аналіз показує, що такі відхилення можуть бути виявлені на основі змін у структурі подій, частоті викликів функцій та характері взаємодії між компонентами системи. Це підтверджує можливість використання запропонованого підходу для виявлення нетипових режимів функціонування serverless-додатків.

Отримані результати свідчать, що запропонована методологія, заснована на аналізі подій та взаємодій між компонентами serverless-систем, може бути використана для дослідження поведінки таких систем та підвищення ефективності виявлення потенційних інцидентів безпеки. Подальші дослідження доцільно спрямувати на використання реальних навантажень, розширення експериментального середовища та застосування методів машинного навчання для автоматизованого аналізу поведінки систем.

7. Висновки

У роботі проведено аналіз сучасних досліджень, присвячених serverless-архітектурам та платформам Function-as-a-Service. Розглянуто архітектурні особливості serverless-систем та основні проблеми їх безпеки.

Проведений аналіз літератури показав, що сучасні дослідження значною мірою зосереджуються на архітектурних особливостях serverless-платформ та механізмах виконання функцій. Водночас питання комплексного аналізу взаємодій між компонентами serverless-систем у контексті забезпечення безпеки залишаються недостатньо дослідженими.

У роботі запропоновано концептуальний підхід до аналізу serverless-систем, що базується на дослідженні послідовностей подій та взаємодій між функціями і сервісами хмарної інфраструктури.

Подальші дослідження доцільно спрямувати на розробку методів моніторингу подій у serverless-системах та аналіз взаємодій між їх компонентами.

Список використаної літератури

1. Baldini I. Serverless computing: Current trends and open problems / I. Baldini, P. Castro, K. Chang, etc. // Research Advances in Cloud Computing. – 2017. – Vol. 2017. – P. 1–20. – DOI: https://doi.org/10.1007/978-981-10-5026-8_1.
2. Jonas E. Cloud programming simplified: A Berkeley view on serverless computing / E. Jonas, J. Schleier-Smith, V. Sreekanti, etc // arXiv preprint arXiv:1902.03383. – 2019. – P. 1–28. – DOI: https://doi.org/10.48550/arXiv.1902.03383.
3. Castro P. The rise of serverless computing / P. Castro, V. Ishakian, V. Muthusamy, A. Slominski // Communications of the ACM. – 2019. – Vol. 62, № 12. – P. 44–54. – DOI: https://doi.org/10.1145/3368454.
4. Spillner J. Faaster, better, cheaper: The prospect of serverless scientific computing and HPC / J. Spillner // Future Generation Computer Systems. – 2019. – Vol. 90. – P. 1–10. – DOI: https://doi.org/10.1016/j.future.2018.07.048.
5. Hendrickson S. Serverless computation with OpenLambda / S. Hendrickson, S. Sturdevant, T. Harter, etc. // IEEE Cloud Computing. – 2016. – Vol. 3, № 6. – P. 44–51. – DOI: https://doi.org/10.1109/MCC.2016.124.
6. Shafiei M. Security challenges in serverless computing / M. Shafiei, A. Khonsari // Journal of Cloud Computing. – 2020. – Vol. 9. – Art. 55. – DOI: https://doi.org/10.1186/s13677-020-00208-0.
7. Wang L. Peeking behind the curtains of serverless platforms / L. Wang, M. Zhang, Y. Chen, etc. // USENIX Annual Technical Conference. – 2018. – P. 133–146.
8. Sbarski P. Serverless Architectures on AWS / P. Sbarski. – Shelter Island, NY: Manning Publications, 2017. – 350 p. – ISBN 978-1617293825.
9. Villamizar M. Infrastructure cost comparison of running web applications in the cloud using AWS Lambda and EC2 / M. Villamizar, O. Garcés, H. Castro, etc. // IEEE Latin America Transactions. – 2017. – Vol. 15, № 12. – P. 2397–2404. – DOI: https://doi.org/10.1109/TLA.2017.8071211.
10. Adzic G. Serverless Applications with Node.js / G. Adzic, R. Chatley. – Shelter Island, NY: Manning Publications, 2017. – 264 p. – ISBN 978-1617294846.
11. Kritikos K. Security and privacy in cloud computing: A systematic literature review / K. Kritikos, D. Plexousakis // ACM Computing Surveys. – 2020. – Vol. 52, № 5. – Art. 97. – DOI: https://doi.org/10.1145/3345509.
12. Eivy A. Be wary of the economics of serverless cloud computing / A. Eivy // IEEE Cloud Computing. – 2017. – Vol. 4, № 2. – P. 6–12. – DOI: https://doi.org/10.1109/MCC.2017.32.

13. Zhang Y. Security and privacy in serverless computing: A systematic literature review / Y. Zhang, X. Chen, J. Li // ACM Computing Surveys. – 2023. – Vol. 55, № 12. – Art. 240. – DOI: https://doi.org/10.1145/3579856.
14. Li H. Security issues and challenges in serverless computing: A survey / H. Li, Y. Wang, K. Zhang // Journal of Cloud Computing. – 2024. – Vol. 13. – Art. 9. – DOI: https://doi.org/10.1186/s13677-024-00521-3.
15. Singh P. Serverless computing: Architecture, security issues and future trends / P. Singh, R. Sharma // Future Generation Computer Systems. – 2023. – Vol. 139. – P. 368–385. – DOI: https://doi.org/10.1016/j.future.2022.10.021.
16. Spillner J. Serverless computing: Design, implementation and performance / J. Spillner, S. Mateos // IEEE Internet Computing. – 2020. – Vol. 24, № 4. – P. 72–79. – DOI: https://doi.org/10.1109/MIC.2020.2995183.
17. Zhang Q. Serverless computing: State of the art and research challenges / Q. Zhang, M. Chen, L. Chen // IEEE Access. – 2022. – Vol. 10. – P. 115364–115379. – DOI: https://doi.org/10.1109/ACCESS.2022.3213420.
18. Lin C. Security analysis of serverless architectures in cloud environments / C. Lin, Y. Wu, J. Chen // Journal of Information Security and Applications. – 2021. – Vol. 58. – Art. 102726. – DOI: https://doi.org/10.1016/j.jisa.2021.102726.
19. Tariq M. Security threats and mitigation techniques in serverless computing / M. Tariq, S. Hameed // Computers & Security. – 2022. – Vol. 114. – Art. 102594. – DOI: https://doi.org/10.1016/j.cose.2021.102594.

Стаття: надійшла до редколегії 11.02.2026
доопрацьована 03.03.2026
прийнята до друку 16.04.2026

SECURITY ANALYSIS OF SERVERLESS ARCHITECTURES BASED ON EVENT-DRIVEN INTERACTION MODELING

P. Vengerskyi, S. Zlatous

Ivan Franko National University of Lviv,
1, Universytetska str., 79000, Lviv, Ukraine,
e-mail: *petro.venherskyy@lnu.edu.ua*, *sviatoslav.zlatous@lnu.edu.ua*

This paper presents an analysis of recent studies on the security of serverless architectures and Function-as-a-Service (FaaS) platforms, taking into account the interaction between system components and the event-driven nature of their operation. The relevance of the topic is driven by the rapid adoption of the serverless paradigm in cloud computing, which enables the development of scalable applications without the need to manage server infrastructure. Despite the significant advantages of this model, the use of serverless architectures introduces new security challenges related to the dynamic execution of functions, the complexity of interactions between cloud infrastructure components, and the limited control of users over the execution environment.

The objective of this work is to analyze existing research in the field of serverless security with a focus on the interactions between functions, events, and cloud services, identify key problems that remain insufficiently explored, and propose approaches to address them. The study includes an analysis of scientific publications devoted to the security of FaaS platforms, particularly those examining possible attack vectors in serverless systems, protection mechanisms during function execution, and issues of security management.

The results show that current approaches to ensuring the security of serverless systems are primarily focused on individual components of the architecture, while a systematic analysis of interactions between functions, events, and cloud services remains insufficiently developed. A conceptual approach to improving the security of serverless systems is proposed, based on event sequence analysis, behavioral analysis techniques, centralized event

monitoring, and automated anomaly detection.

Key words: serverless computing, Function-as-a-Service (FaaS), serverless architecture, cloud computing security, cybersecurity, event monitoring, attack surface analysis, anomaly detection.