***Скоринович Богдан Володимирович****, аспірант*

*Національний університет «Львівська політехніка», м. Львів*

*ORCID: https://orcid.org/0009-0007-8669-9172*

***Гавриляк Володимир Романович****, аспірант*

*Національний університет «Львівська політехніка», м. Львів*

*ORCID: https://orcid.org/0009-0002-4114-1232*

***Кулик Юрій Андрійович****, аспірант*

*Національний університет «Львівська політехніка», м. Львів*

*ORCID: https://orcid.org/0009-0003-9249-2697*

**ВИКЛИКИ ТА ПЕРСПЕКТИВИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ В УМОВАХ “CLOUD NATIVE” АРХІТЕКТУР**

**Вступ**. Хмарні технології та підхід “Cloud Native” стали ключовими рушіями сучасної цифрової трансформації, забезпечуючи можливість створення масштабованих, відмовостійких та оперативно розгортуваних ІТ-систем. Архітектура “Cloud Native” базується на поєднанні DevOps-практик, мікросервісної архітектури, оркестрації контейнерів (здебільшого за допомогою Kubernetes), API-first дизайну, serverless обчислень та інфраструктури як коду (IaC) [1]. Такі рішення надають підприємствам значну гнучкість, скорочують час виведення продукту на ринок та створюють нові можливості масштабування. Водночас, через свою динамічність, розподіленість і короткий цикл життя компонентів, вони створюють безпрецедентні виклики у сфері кібербезпеки.

На відміну від класичних, периметрально орієнтованих моделей, “cloud native” середовища характеризуються постійною зміною стану, високою автоматизацією та великою кількістю взаємопов’язаних сервісів. У таких умовах традиційні засоби захисту – статичні фаєрволи, ручне управління доступом, централізоване логування – втрачають ефективність, не встигаючи за темпами змін та розмиттям периметру. Фундаментальна зміна полягає в тому, що безпека більше не може розглядатися як фінальний етап життєвого циклу – вона має бути інтегрованою з самого початку, охоплюючи всі фази розробки, розгортання та експлуатації програмного забезпечення.

**Основні виклики безпеки в “Cloud Native” середовищах.** Перехід до “Cloud Native” архітектур супроводжується появою низки специфічних викликів у сфері кібербезпеки, що зумовлюють потребу у нових, адаптивних підходах до захисту. Одним із ключових ризиків є суттєве розширення поверхні атаки через велику кількість сервісів, API та точок взаємодії.

Зокрема, розподілені мікросервіси взаємодіють через мережу, і помилки в налаштуванні API можуть дозволити обійти автентифікацію або отримати доступ до внутрішніх ресурсів. У serverless архітектурах кожна функція є окремим об’єктом атаки, вразливим до експлуатації надмірних дозволів, ін’єкцій вхідних подій або вразливостей у бібліотеках.

Інфраструктура як код (IaC), попри значні переваги в автоматизації управління ресурсами, створює додаткові вектори ризику — від зберігання секретів (API-ключів, токенів) у відкритому коді до шаблонів із помилками конфігурації. Типові приклади включають створення загальнодоступних S3-бакетів, відкритих портів у security groups або надання надмірних прав у IAM-політиках, що створює загрозу компроментації середовища [2].

Контейнеризація додає ще один рівень складності: публічні базові образи часто містять застарілі або вразливі пакети, а некоректні конфігурації Kubernetes — наприклад, привілейовані контейнери, публічні дашборди або необмежені мережеві політики — відкривають широкі можливості для атаки. Актуальними залишаються і загрози середовища виконання: втеча з контейнера (container escape), експлуатація уразливостей процесів, компрометація файлових систем [3].

Крім загальних уразливостей, “Cloud Native” архітектури схильні до специфічних загроз і атак:

* *API-вразливості та проблеми авторизації.* API-first підхід висуває API на передній план як основний інтерфейс взаємодії, що робить їх привабливою ціллю. Типові атаки включають порушення авторизації на об’єктному рівні (BOLA), неефективну автентифікацію, витік метаданих та відсутність обмежень на запити (OWASP API Security Top 10) [4].
* *Supply Chain атаки.* Компрометація ланцюгів постачання охоплює CI/CD-інфраструктуру, публічні або внутрішні репозиторії, інструменти складання, а також контейнерні образи з вбудованим шкідливим кодом.
* *Компрометація облікових даних.* Credential stuffing, викрадення токенів, неправильне зберігання ключів у відкритому коді або контейнерних образах — поширені сценарії, що загрожують конфіденційності.
* *Атаки на інфраструктуру оркестрації.* Kubernetes стає окремою ціллю: уразливості в API-сервері, etcd, kubelet або service mesh, зловживання сервісними акаунтами, відсутність контролю доступу.
* *Server-Side Request Forgery (SSRF).* Особливо небезпечні у контексті хмарних провайдерів: дозволяють отримати доступ до метаданих-ендпоінтів і витягти тимчасові облікові дані.
* *Використання ресурсів для майнінгу (Cryptojacking).* Слабко захищені обчислювальні ресурси можуть бути несанкціоновано використані для генерації криптовалюти.
* *Dangling ресурси.* IP-адреси, DNS-записи, сертифікати, які не видаляються вчасно з конфігурацій, створюють ризик перехоплення трафіку або обману користувачів.
* *Розподілені атаки на відмову в обслуговуванні (DDoS).* Орієнтовані на API, serverless-функції, балансувальники навантаження — часто складні для ідентифікації й потребують адаптивного захисту.
* *Неправильно налаштовані сховища.* Публічно доступні S3-бакети, відкриті бази даних або storage-сервіси — типова причина витоків даних в хмарі.

Крім цього, моніторинг безпеки в “Cloud Native” середовищах стикається з низкою технічних обмежень. Ефемерна природа сервісів і контейнерів, велика кількість логів, що зберігаються у різних локаціях, а також висока швидкість змін в інфраструктурі значно ускладнюють виявлення загроз. Події безпеки можуть бути розподілені між десятками мікросервісів, а короткий життєвий цикл функцій у serverless-моделях обмежує доступ до інфраструктурного рівня. Як наслідок, класичні підходи до логування, кореляції подій та менеджменту інцидентів виявляються недостатніми, що вимагає впровадження нових, спеціалізованих рішень – здатних працювати у масштабованому, розподіленому та динамічному середовищі [2].

**Сучасні підходи та технології для забезпечення безпеки “Cloud Native” архітектур.** У відповідь на численні виклики, які постають у процесі забезпечення кібербезпеки “Cloud Native” архітектур, сформувався широкий спектр підходів, рішень та інструментів. Їх спільною рисою є орієнтація на інтеграцію безпеки на всіх рівнях, автоматизацію контрольних механізмів та використання інтелектуальних систем.

У таблиці 1 відображено напрямки захисту таких середовищ, типові загрози, відповідні практики та приклади інструментів.

**Таблиця 1**

Основні напрямки забезпечення безпеки в “Cloud Native” середовищах

|  |  |  |  |
| --- | --- | --- | --- |
| **Напрямок** | **Типові загрози** **/ проблеми** | **Підхід до захисту** | **Приклади** **інструментів** |
| DevSecOps та IaC | Уразливий код, неконтрольовані залежності, помилки конфігурацій в IaC | Автоматизовані перевірки в CI/CD, політики як код | SAST, DAST, SCA, tfsec, Checkov, OPA, Regoʼ |
| Контейнеризація та оркестрація | Вразливі образи, загрози середовищ виконання, привілейовані контейнери, слабкі політики Kubernetes | Сканування, підпис образів, моніторинг, Kubernetes policy enforcement | Trivy, Clair, Cosign, Falco, Sysdig, Kyverno, Vault |
| Serverless | Надмірні дозволи, вразливі тригери, короткотривалі функції без моніторингу | Мінімізація прав, обмеження часу виконання, валідація подій | IAM Policies, AWS Lambda Limits, CloudWatch, Datadog Serverless |
| Хмарна інфраструктура | Публічні ресурси, відсутність стандартизації, слабкий контроль змін | CSPM, KSPM, CIEM, аналіз конфігурацій, управління привілеями | Prisma Cloud, Wiz, Azure Defender, AWS Config, IAM Access Analyzer |
| Інтелектуальні системи (AI/ML) | Zero-day атаки, аномалії, складні інциденти, шум у телеметрії | LLM, ML-аналітика, поведінковий аналіз, автоматизоване реагування (SOAR) | GPT-4, CodeWhisperer, Splunk SOAR, UEBA, OpenAI Security Copilot |

Представлена таблиця демонструє, що безпека “Cloud Native” середовищ вимагає багаторівневого, скоординованого підходу, який поєднує профілактику (DevSecOps, IaC security), контроль середовища виконання (контейнери, serverless), моніторинг стану (CSPM, SIEM), а також адаптивну аналітику загроз з використанням інтелектуальних систем [1,2,3].

**Висновки та перспективи.** “Cloud Native” архітектури відкривають широкі можливості для створення гнучких, масштабованих і динамічних цифрових рішень, проте водночас істотно ускладнюють завдання забезпечення кібербезпеки. Динамічна природа, ефемерність компонентів, велика кількість точок взаємодії та автоматизація життєвого циклу ПЗ потребують якісно нових підходів до захисту.

Сучасна практика вимагає впровадження багаторівневого, проактивного та адаптивного захисту, який охоплює всі фази – від написання коду до середовища виконання. Безпека має бути вбудованою складовою архітектури, а не окремим етапом, що накладається ззовні.

Ключовими напрямками ефективного забезпечення безпеки в Cloud Native є:

* DevSecOps — як спосіб інтеграції безпеки у всі етапи SDLC;
* IaC безпека та політики як код — для формалізації вимог до конфігурацій;
* Контейнерна та serverless безпека — як сфери активного технічного розвитку;
* Автоматизований моніторинг та реагування — із залученням CSPM, KSPM, SIEM/UEBA;
* Інтелектуальні системи (AI/LLM) — як перспективний напрям аналізу загроз і коду.

У найближчій перспективі одним із пріоритетів стане синергія між DevSecOps-практиками та інтелектуальними системами, які здатні самостійно аналізувати, прогнозувати й реагувати на загрози у реальному часі. Значне поширення отримає підхід CNAPP (Cloud Native Application Protection Platform), який прагне поєднати CSPM, CWPP, CIEM та інші рішення в єдину платформу наскрізної безпеки.

Таким чином, побудова надійної моделі кібербезпеки в умовах “Cloud Native” архітектур – це не лише технологічне, а й методологічне завдання, що вимагає глибокої інтеграції процесів, інструментів і рішень на всіх рівнях цифрової інфраструктури.

**Література**

1. Гавриляк В. Р., Опірський І. Р. Аналіз використання штучного інтелекту в DevSecOps та кібербезпеці // Сучасний захист інформації. – 2024. – № 4(60). – С. 250–260. – DOI: 10.31673/2409-7292.2024.040008.

2. Скоринович Б. В., Лах Ю. В. Аналіз методів моніторингу стану безпеки в хмарному середовищі // Сучасний захист інформації. – 2025. – № 1(61). – С. 298–310. – DOI: 10.31673/2409-7292.2025.012256.

3. Кулик Ю. А., Лах Ю. В. Аналіз безпеки мережевих плагінів Kubernetes // Сучасний захист інформації. – 2025. – № 1(61). – С. 311–317. – DOI: 10.31673/2409-7292.2025.015886.

4. OWASP Foundation. API Security Top 10. – [Електронний ресурс]. – Режим доступу:<https://owasp.org/API-Security/> – Дата звернення: 08.05.2025.