УДК 004.056.5:004.75

Частоколенко Ігор Павлович, кандидат фізико-математичних наук, доцент,

доцент кафедри фізико-математичних дисциплін, Національний університет цивільного захисту України, Черкаси, Україна,

e-mail: igor.chst@gmail.com

ORCID: https://orcid.org/0000-0001-9323-2684

Газдун Назарій Володимирович,

Технічний Директор, Джиніусі Україна, Київ, Україна,

e-mail: nazariy.hazdun@gmail.com

ORCID: https://orcid.org/0009-0004-9096-1151

АРХІТЕКТУРА ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ API В МІКРОСЕРВІСНИХ ВЕБ-СИСТЕМАХ

Анотація.

У статті запропоновано архітектуру забезпечення безпеки API у мікросервісних веб-системах, що поєднує засоби аутентифікації та авторизації, захисту від типових вразливостей, виявлених у OWASP API Security Top 10, а також механізми контролю доступу на основі принципів zero trust. Показано, що зі зростанням кількості мікросервісів та публічних API суттєво зростає площина атаки, а класичні підходи до периметрового захисту виявляються недостатніми. Запропонована формальна модель мікросервісної

веб-системи розглядає множину сервісів, їхні кінцеві точки API та відповідні атрибути безпеки, зокрема політики аутентифікації, авторизації, швидкісні обмеження та механізми фільтрації трафіку. На основі цієї моделі визначено множину загроз, що відповідають основним категоріям уразливостей API, та побудовано матрицю відповідності між кінцевими точками і класами атак. Для кожної кінцевої точки вводиться функція ризику, яка залежить від імовірності реалізації загрози, потенційних збитків та ефективності застосованих засобів захисту. Запропоновано математичну модель зниження інтегрального ризику шляхом впровадження набору технічних контролів на рівні шлюзу API, сервісної сітки та окремих мікросервісів. Введено показники ефективності архітектури, що враховують кількість блокованих атак, частку захищених кінцевих точок, додаткові затримки обробки запитів та витрати ресурсів. Наведено схему архітектури безпеки API, яка ілюструє взаємодію клієнтів, шлюзу API, служби ідентифікації, сервісної сітки, журналювання та системи моніторингу. Отримані результати можуть бути використані для проєктування нових та модернізації існуючих мікросервісних веб-систем, а також для кількісного обґрунтування вибору засобів захисту. Додатково проаналізовано характерні сценарії атак на API у мікросервісних архітектурах, зокрема зловживання токенами доступу, маніпуляції з параметрами запитів, масове створення запитів з метою виснаження ресурсів та обхід міжсервісної автентифікації. На основі аналізу сформульовано рекомендації щодо побудови багаторівневої системи захисту, яка поєднує класифікацію запитів, централізоване керування політиками доступу та безперервний моніторинг.

Ключові слова: безпека API, мікросервісна архітектура, веб-системи, zero trust, сервісна сітка, шлюз API, математична модель ризику.

Ihor Chastokolenko,

Candidate of Physical and Mathematical Sciences, docent,

Associate Professor of the Department of Physics and Mathematics,

National University of Civil Defense of Ukraine, Cherkasy, Ukraine,

e-mail: igor.chst@gmail.com,

ORCID: https://orcid.org/0000-0001-9323-2684

Nazariy Hazdun,

Chief Technology Officer, Geniusee, Kyiv, Ukraine

e-mail: nazariy.hazdun@gmail.com

ORCID: https://orcid.org/0009-0004-9096-1151

ARCHITECTURE FOR API SECURITY IN MICROSERVICE-BASED WEB SYSTEMS

Abstract.

The paper proposes an architecture for ensuring API security in microservice-based web systems that combines authentication and authorisation mechanisms, protection against typical vulnerabilities identified in the OWASP API Security Top 10, and access control

mechanisms based on zero-trust principles. It is shown that as the number of microservices and public APIs grows, the attack surface increases significantly, and classical perimeter security approaches become insufficient. A formal model of a microservice web system is introduced, which considers the set of services, their API endpoints and corresponding security attributes, including authentication and authorisation policies, rate limiting and traffic filtering mechanisms. Based on this model, a set of threats corresponding to the main categories of API vulnerabilities is defined, and a mapping matrix between endpoints and classes of attacks is constructed. For each endpoint, a risk function is introduced that depends on the probability of threat realisation, potential impact and the effectiveness of deployed security controls. A mathematical model of reducing integral risk by implementing a set of technical controls at the API gateway, service mesh and microservice levels is proposed. Performance indicators of the architecture are defined, taking into account the number of blocked attacks, the share of protected endpoints, additional request processing latency and resource consumption. An architectural diagram is presented that illustrates the interaction of clients, API gateway, identity service, service mesh, logging and monitoring system. The obtained results can be used for designing new and modernising existing microservice-based web systems, as well as for quantitative justification of security control selection.

Keywords: API security, microservice architecture, web systems, zero trust, service mesh, API gateway, mathematical risk model.

Постановка проблеми.

Сучасні веб-системи дедалі частіше проєктуються на основі мікросервісної архітектури, яка дозволяє незалежно масштабувати окремі компоненти, пришвидшувати розробку та спрощувати розгортання змін. Водночас така архітектура призводить до суттєвого збільшення кількості мережевих взаємодій між

сервісами, появи численних публічних та внутрішніх API й ускладнює централізований контроль безпеки. Кожна кінцева точка API потенційно перетворюється на точку входу для зловмисника, а помилки авторизації, некоректна валідація даних або відсутність обмежень на використання ресурсів можуть призвести до компрометації всієї системи. Традиційні підходи, що орієнтуються на периметровий захист, не враховують багатоканальну взаємодію між мікросервісами та не забезпечують належного рівня контролю доступу на рівні кожної окремої кінцевої точки. Актуальною науковою проблемою є розробка архітектури забезпечення безпеки API мікросервісних веб-систем, яка спирається на формальну математичну модель ризику, враховує специфіку мікросервісних взаємодій та дозволяє кількісно оцінювати ефективність впроваджених засобів захисту.

Аналіз останніх досліджень і публікацій.

Міжнародні організації та дослідницькі спільноти приділяють значну увагу питанням безпеки API та мікросервісних архітектур. У рекомендаціях OWASP API Security Top 10 визначено основні класи вразливостей API, зокрема порушення контролю доступу до об’єктів, помилки автентифікації, надмірне розкриття даних, відсутність обмежень на використання ресурсів, ін’єкційні атаки та інші типові загрози [1]. Документ NIST SP 800-204A узагальнює підходи до побудови безпечних мікросервісних додатків із використанням сервісної сітки та підкреслює важливість централізованого керування сертифікатами, токенами доступу, політиками шифрування та міжсервісної аутентифікації [2]. Окремі роботи фокусуються на впровадженні принципів zero-trust у мікросервісних архітектурах, де жодна взаємодія не вважається апріорі довіреною, а доступ надається відповідно до мінімально необхідних прав [3].
В українських наукових публікаціях розглядаються питання

безпеки мікросервісних систем та веб-застосунків. Зокрема, у роботі проведено огляд мікросервісної архітектури та проаналізовано типові вразливості, пов’язані з помилками конфігурації контейнерних середовищ, управлінням доступом та шифруванням каналів взаємодії [4]. У дослідженнях, присвячених захисту веб-застосунків від шкідливих міждоменних запитів і помилок політики CORS, підкреслюється важливість правильної конфігурації заголовків, атрибутів cookie та поєднання браузерних і серверних механізмів захисту [5]. Робота, присвячена рекомендаціям із тестування безпеки API, пропонує методологію поетапного тестування публічних та внутрішніх інтерфейсів з використанням спеціалізованих інструментів аналізу [6]. Водночас існуючі публікації здебільшого або зосереджені на загальному огляді проблем безпеки мікросервісних архітектур, або описують окремі аспекти тестування та аналізу вразливостей API. Питання побудови цілісної архітектури безпеки API у мікросервісних веб-системах із використанням формальної математичної моделі ризику та кількісного оцінювання ефективності технічних контролів залишається недостатньо дослідженим, що й визначає актуальність даної роботи.

Мета статті.

Метою статті є розробка математично обґрунтованої архітектури забезпечення безпеки API у мікросервісних веб-системах, яка поєднує багаторівневі технічні контролі на рівні шлюзу API, сервісної сітки та окремих мікросервісів, а також дозволяє здійснювати кількісну оцінку ризику для окремих кінцевих точок і всієї системи в цілому.

Виклад основного матеріалу.

1. Формальна модель мікросервісної веб-системи.

Розглянемо мікросервісну веб-систему як впорядковану трійку:

$S = (M, E, C)$,

де $M = \{m_1, m_2, \ldots, m\_n\}$ — множина мікросервісів; E — множина кінцевих точок API; C — множина засобів захисту.

Кожен мікросервіс m_i має власну підмножину кінцевих точок $E\_i \subseteq E$, при цьому $E = \bigcup E\_i$. Кінцева точка $e \in E$ характеризується набором атрибутів безпеки:

$a(e) = (auth(e), scope(e), rl(e), enc(e), log(e))$,

де $auth(e) \in \{0,1\}$ позначає наявність механізму автентифікації; $scope(e)$ — множина дозволених прав доступу; $rl(e)$ — параметри обмеження швидкості (кількість запитів за одиницю часу); $enc(e) \in \{0,1\}$ — використання шифрування транспортного рівня; $log(e) \in \{0,1\}$ — наявність журналювання звернень.

Множина загроз $T = \{t_1, t_2, \ldots, t\_k\}$ будується на основі класифікації OWASP API Security Top 10, зокрема: t_1 — порушення контролю доступу до об'єктів, t_2 — помилки автентифікації, t_3 — надмірне розкриття даних, t_4 — неконтрольоване споживання ресурсів, t_5 — ін'єкційні

атаки тощо [1]. Для кожної пари (e, t_j) визначається бінарна змінна v_{e,j} ∈ {0,1}, що описує наявність уразливості даного класу для відповідної кінцевої точки.

$$S = (M, E, C)$$

$$M = \{m_1, m_2, \ldots, m\_n\}$$

$$E = \bigcup E\_i$$

$$a(e) = (auth(e), scope(e), rl(e), enc(e), log(e))$$

$$T = \{t_1, t_2, \ldots, t\_k\}$$

2. Математична модель ризику.

Нехай P_{e,j} ∈ [0,1] — умовна ймовірність успішної реалізації загрози t_j відносно кінцевої точки e за відсутності додаткових засобів захисту, а I_{e,j} ≥ 0 — оцінка потенційних збитків від такої реалізації (у відносних або грошових одиницях). Базовий ризик для кінцевої точки e визначимо як

$R_0(e) = \Sigma\_\{j=1\}^{k} v\_\{e,j\} \cdot P\_\{e,j\} \cdot I\_\{e,j\}$.

Розглянемо множину засобів захисту $C = \{c_1, c_2, \ldots, c\_m\}$, кожен з яких характеризується набором коефіцієнтів зниження ризику $\alpha\_\{c,t\_j\} \in [0,1]$ для відповідних класів загроз. Для кінцевої точки e впроваджується підмножина $C(e) \subseteq C$. Припустимо, що вплив засобів захисту на різні класи загроз є мультиплікативним. Тоді скоригований

ризик можна записати як

$R(e) = \Sigma_{j=1}^{k} v_{e,j} \cdot P_{e,j} \cdot I_{e,j} \cdot \Pi_{c \in C(e)} (1 - \alpha_{c,t_j})$.

Інтегральний ризик для всієї системи визначається сумою ризиків по всіх кінцевих точках:

$R_\Sigma = \Sigma_{e \in E} R(e)$.

Завдання проєктування архітектури безпеки API зводиться до вибору такої конфігурації засобів $C(e)$ для всіх $e \in E$, яка мінімізує R_Σ за обмежень на вартість та продуктивність системи. Оцінимо витрати на впровадження засобів захисту функцією

$Cost = \Sigma_{e \in E} \Sigma_{c \in C(e)} к_c$,

де $к_c \geq 0$ — питомі витрати на впровадження засобу c.

Тоді задача оптимізації має вигляд:

мінімізувати R_Σ за умов $Cost \leq C_{max}$,

де C_{max} — допустимий бюджет на заходи захисту.

$$P_{e,j} \in [0,1], \; I_{e,j} \geq 0$$

$$R_0(e) = \sum_{j=1}^{k} v_{e,j} \cdot P_{e,j} \cdot I_{e,j}$$

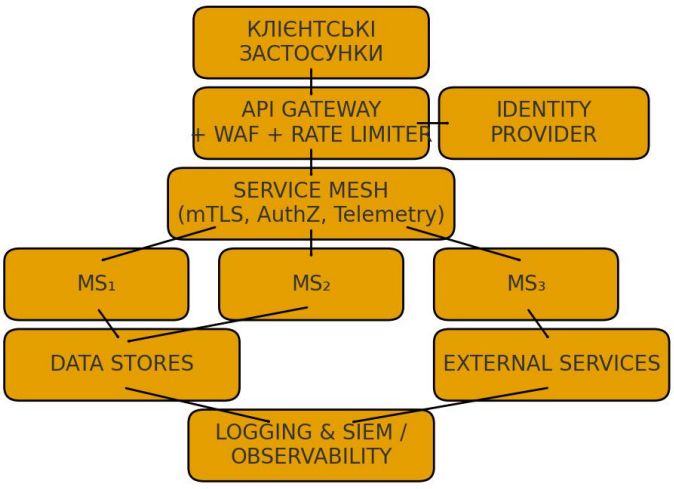
$$R(e) = \sum_{j=1}^{k} v_{e,j} \cdot P_{e,j} \cdot I_{e,j} \cdot \prod_{c \in C(e)} (1 - \alpha_{c,t_j})$$

$$R_\Sigma = \sum_{e \in E} R(e)$$

$$Cost = \sum_{e \in E} \sum_{c \in C(e)} к_c$$

3. Архітектура забезпечення безпеки API.
Запропонована архітектура базується на поєднанні шлюзу API, сервісної сітки та локальних механізмів захисту в кожному мікросервісі. На логічному рівні виділяються такі компоненти:
— клієнтські застосунки (браузери, мобільні застосунки, зовнішні системи);
— шлюз API з вбудованими функціями WAF, rate limiting та термінування TLS;
— служба ідентифікації та авторизації (Identity Provider);
— сервісна сітка (service mesh) з підтримкою mTLS, авторизації сервіс-до-сервісу та телеметрії;
— мікросервісний кластер;
— система журналювання та моніторингу (SIEM/observability-платформа).
Схематичне подання архітектури наведено нижче.

Схема 1 – Узагальнена архітектура забезпечення безпеки API мікросервісної веб-системи



У наведеній схемі всі зовнішні запити проходять через шлюз API, який забезпечує термінування TLS-з’єднання, первинну фільтрацію трафіку (WAF), а також обмеження швидкості запитів

відповідно до налаштованих політик. Шлюз перевіряє коректність маркерів доступу (наприклад, JWT), отриманих від служби ідентифікації, та додає до запиту атрибути, необхідні для подальшої авторизації на рівні сервісної сітки. Сервісна сітка забезпечує взаємну автентифікацію мікросервісів за допомогою сертифікатів, шифрування міжсервісного трафіку (mTLS), застосування політик доступу до окремих маршрутів і збирання телеметрії для подальшого аналізу. Локальні механізми захисту в мікросервісах включають валідацію вхідних даних, контроль прав доступу до доменних об'єктів, обмеження використання ресурсів та журналювання критичних операцій.

4. Приклад кількісного оцінювання ефективності архітектури. Розглянемо спрощений випадок, коли для кожної кінцевої точки е суттєвими є лише два класи загроз: t_1 (порушення контролю доступу до об'єктів) та t_4 (неконтрольоване споживання ресурсів). Припустимо, що базові параметри для певної публічної кінцевої точки e_0 задаються як
$P_{e_0,1} = 0{,}3$, $I_{e_0,1} = 10$;
$P_{e_0,4} = 0{,}5$, $I_{e_0,4} = 6$,
і обидва класи вразливостей є релевантними ($v_{e_0,1} = v_{e_0,4} = 1$). Базовий ризик дорівнює
$R_0(e_0) = 0{,}3 \cdot 10 + 0{,}5 \cdot 6 = 6$.
Впровадимо два засоби захисту: c_1 — централізовану авторизацію на рівні сервісної сітки, що зменшує ризик t_1 на 70 % ($\alpha_{c_1,t_1} = 0{,}7$), та c_2 — механізм rate limiting на рівні шлюзу API, який знижує ризик t_4 на 60 % ($\alpha_{c_2,t_4} = 0{,}6$). Тоді скоригований ризик становитиме
$R(e_0) = 0{,}3 \cdot 10 \cdot (1 - 0{,}7) + 0{,}5 \cdot 6 \cdot (1 - 0{,}6) = 0{,}9 + 1{,}2 = 2{,}1$.
Таким чином, інтегральний ризик для кінцевої точки зменшується у 2{,}86 раза. Якщо вартість впровадження засобів c_1 та c_2 є меншою за граничний бюджет C_max, така конфігурація може розглядатися як ефективна. Запропонована модель дозволяє

аналогічним чином оцінювати вплив комбінацій засобів захисту на інтегральний ризик для всієї системи.

$$P\_\{e_0, 1\} = 0\{,\}3, \quad I\_\{e_0, 1\} = 10$$

$$P\_\{e_0, 4\} = 0\{,\}5, \quad I\_\{e_0, 4\} = 6$$

$$R_0(e_0) = 0\{,\}3 \cdot 10 + 0\{,\}5 \cdot 6 = 6$$

$$R(e_0) = 0\{,\}3 \cdot 10 \cdot (1 - 0\{,\}7) + 0\{,\}5 \cdot 6 \cdot (1 - 0\{,\}6) = 0\{,\}9 + 1\{,\}2 = 2\{,\}1$$

Висновки.

У роботі розроблено архітектуру забезпечення безпеки API в мікросервісних веб-системах, що поєднує засоби захисту на рівні шлюзу API, сервісної сітки та окремих мікросервісів. Запропоновано формальну математичну модель мікросервісної системи, яка описує множину кінцевих точок API, відповідні класи загроз та засоби захисту, а також дозволяє обчислювати інтегральний ризик для окремих кінцевих точок і всієї системи. Введено модель зниження ризику за рахунок впровадження технічних контролів та сформульовано оптимізаційну задачу вибору конфігурації засобів безпеки за обмеженого бюджету. Практичне значення роботи полягає у можливості використання запропонованої архітектури та математичної моделі на етапах проєктування й модернізації мікросервісних веб-систем, а також у процесі аудиту їхньої безпеки. Подальші дослідження можуть бути спрямовані на уточнення моделей ймовірностей реалізації загроз на основі емпіричних даних журналювання, інтеграцію методів машинного навчання для динамічної оцінки ризику та розширення архітектури за рахунок автоматизованих механізмів реагування на інциденти.

Література

[1] OWASP Foundation. OWASP Top 10 API Security Risks – 2023. URL: https://owasp.org/API-Security/ (дата звернення: 01.12.2025).

[2] Chandramouli R. Building Secure Microservices-based Applications Using Service-Mesh Architecture. NIST Special Publication 800-204A. Gaithersburg, MD, 2020. 54 p.

[3] Zero-Trust for Microservices, a Practical Blueprint. Cerbos, 2025. URL: https://www.cerbos.dev/blog/zero-trust-for-microservices (дата звернення: 01.12.2025).

[4] Кудрявцев Д. О., Мичуда Л. З. Огляд мікросервісної архітектури та аналіз типових вразливостей. Computer Systems and Networks. 2025. Vol. 7, No. 2. С. 114–123.

[5] Ковальчук Л. П. Методи і способи захисту від шкідливих міждоменних запитів у веб-застосунках. Журнал прикладних інформаційних технологій. 2025. № 2. С. 1–32.

[6] Колесов О. О. Рекомендації із тестування безпеки API сервісів. Архів кваліфікаційних робіт. Донецький національний університет імені Василя Стуса. 2022.

References

[1] OWASP Foundation (2023), "OWASP Top 10 API Security Risks – 2023", available at: https://owasp.org/API-Security/ (accessed 01 December 2025).

[2] Chandramouli, R. (2020), Building Secure Microservices-based Applications Using Service-Mesh Architecture, NIST Special Publication 800-204A, Gaithersburg, MD.

[3] Zero-Trust for Microservices, a Practical Blueprint (2025), Cerbos, available at: https://www.cerbos.dev/blog/zero-trust-for-microservices (accessed 01 December 2025).

[4] Kudriavtsev, D. O. and Mychuda, L. Z. (2025), "Ohliad mikroservisnoi arkhitektury ta analiz typovykh vrazlyvostei" [Review of microservice architecture and analysis of typical vulnerabilities], Computer Systems and Networks, Vol. 7, No. 2, pp. 114–123.

[5] Kovalchuk, L. P. (2025), "Metody i sposoby zakhystu vid shkidlyvykh mizhdomenykh zapytiv u veb-zastosunkakh" [Methods and means of protection against malicious cross-domain requests in web applications], Journal of Applied Information Technologies, No. 2, pp. 1–32.

[6] Kolesov, O. O. (2022), "Rekomendatsii iz testuvannia bezpeky API servisiv" [Recommendations for testing the security of API services], Archive of Qualification Works, Vasyl’ Stus Donetsk National University.