УДК 004.75 http://doi.org/10.15407/pp2025.04.032

І.П. Малініч, Я.В. Іванчук

ВИЗНАЧЕННЯ ПРІОРИТЕТНОСТІ ХМАРНИХ СЕРВІСІВ ДЛЯ ДИНАМІЧНОГО СТВОРЕННЯ ПРАВИЛ WAF

В статті розглянуто процес визначення пріоритетності хмарних сервісів, а також їх покриття мережевим екраном. Проаналізовано структуру та основні параметри раніше зібраних хмарних гібридних конфігурацій. Приділено увагу особливостям розміщення хмарних сервісів у гібридних хмарах та їх покриттю мережевими екранами веб-додатків. Часто такі мережеві екрани входять до набору типових послуг таких сервісів як CloudFlare, надаючи можливість покривати одночасно всю гібридну хмару. Також розглянуто різні типи доступу до хмарних сервісів, які можуть забезпечувати як безпосередній доступ, так і використовувати реверсивне проксіювання, в якому відбувається термінування захищених з'єднань та застосування статичних та динамічних правил мережевого екрана. В даному дослідженні приділяється увага зібраним описовим даним про гібридні хмари з попередніх досліджень, зокрема, хмарним сервісам, а також їхнім зв'язкам. У контексті даного дослідження патерни пріоритетності мають використовуватись для динамічного створення правил мережевого екрана. Патерни пріоритетності необхідні для динамічного створення дозволяючих або забороняючих правил. Це особливо актуально під час автоматизації налаштування мережевого екрана за допомогою інструментів генеративного штучного інтелекту. У статті запропоновано два показники для створення патернів пріоритетності правил мережевого екрана – пріоритет доступності та пріоритет покриття мережевим екраном. Пріоритет доступності визначає рівень критичності забезпечення беззбійного доступу певного хмарного сервісу. Пріоритет покриття мережевим екраном у свою чергу визначає рівень обмеження доступу до хмарного сервісу. В даному дослідженні проведено експертне опитування щодо параметрів доступності та захисту всіх хмарних сервісів, зібраних у попередньому дослідженні. В статті пропонується використання цих двох показників для створення патернів пріоритетності для мережевого екрана веб-додатків.

Ключові слова: обчислювальна хмара, мережевий екран, веб-додатки, динамічні правила

I.P. Malinich, Y.V. Ivanchuk

DEFINING OF CLOUD SERVICE PRIORITY FOR DYNAMIC CREATING WAF RULES

The article examines the process of determining the prioritization of cloud services and their coverage by network firewalls. The structure and key parameters of previously collected hybrid cloud configurations are analyzed. Particular attention is given to the specifics of cloud service deployment within hybrid clouds and their coverage by web application firewalls. Frequently, such firewalls are included among the standard services offered by providers such as Cloudflare, allowing comprehensive protection of the entire hybrid cloud environment.

The article also discusses different types of access to cloud services, which may provide either direct access or employ reverse proxying. In the latter case, secure connections are terminated, and both static and dynamic firewall rules are applied. This study focuses on descriptive data collected from previous research on hybrid clouds, particularly concerning cloud services and their interconnections. Within the context of this study, priority patterns are intended to be used for the dynamic generation of firewall rules. These priority patterns are necessary for dynamically creating either permissive or restrictive rules. This approach is especially relevant for automating firewall configuration using generative artificial intelligence tools. The article proposes two indicators for the development of firewall rule priority patterns: the availability priority and the firewall coverage priority. The availability priority determines the level of criticality in ensuring uninterrupted access to a specific cloud service, whereas the firewall coverage priority defines the degree of access restriction to that service. An expert survey was conducted as part of this research to evaluate the availability and protection parameters of all cloud services collected in previous studies. The article proposes using these two metrics for creating priority patterns for the web application firewall.

Key words: cloud computing, firewall, web applications, dynamic rules.

© І.П. Малініч, Я.В. Іванчук, 2025

Вступ

У використанні обчислювальних хмар важливим аспектом є налаштування мережевого екрана (англ. Firewall). Мережевий екран у обчислювальній хмарі потрібен не лише для запобігання несанкціонованому доступу, а й є одним із компонентів для запобігання DDoS-атакам. Одним із різновидів мережевого екрана є WAF (мережевий екран веб-додатків, англ. Web application firewall). Крім безпосередньо хмари WAF також надаються сервісами для протидії DDoS-атакам, такими як CloudFlare, Fastly та іншими [1]. Ці сервіси можуть динамічно створювати конфігурації WAF, подібне застосування є ефективним у поєднанні з широко використовуваними фреймворками та/або дригунами на кшталт WordPress, Drupal, Joomla та іншими [1]. Однак для нових API та веб-додатків, які не використовують стандартизовані шаблони доступу до контенту створення динамічних правил WAF може бути складнішим і потребувати більш тонкого налаштування з боку інженерів. Занадто складні правила мережевого екрана можуть сповільнювати доступ до API чи веб-додатків та споживати багато процесорних ресурсів на стороні мережевого екрану. Наукова цінність даного дослідження полягає у визначенні міри впливу атрибутів хмарних сервісів та їхніх зв'язків на покриття мережевим екраном WAF.

Аналіз останніх досліджень і публікацій

Найбільш детально розглянуто принцип роботи WAF у статті [1]. Там моделюються різні види атак та пояснюється, як мережевий екран WAF може їх долати. У статті безпосередньо не розглядається створення правил WAF, не зважаючи на те, що сервіс CloudFlare, показаний у статті, має багато можливостей для налаштування різних правил WAF. Сервіс CloudFlare має багато можливостей також для захисту гібридних хмар, проте цього не було розглянуто у статті, оскільки досліджувалося використання одного сервера. Структуру та зв'язки всередині гібридних хмар розглянуто у публікаціях [2, 3], однак приділено замало уваги мережевим екранам WAF у поєднанні з гібридними хмарами. Автори статті [4] розглядають можливість балансування навантаження у мультихмарних розгортаннях. Попри те, що у статті безпосередньо не розглядається конфігурація чи використання мережевих екранів у хмарах, результати дослідження можна використати при структуризації даних. У іншій статті [5] розглядаються сценарії застосування гібридних хмар та описуються пов'язані з ними безпекові ризики, яким можливо запобігти за допомогою мережевих екранів, однак не розглядається їх практичне застосування. Робота [6] показує різні варіанти організації хмарної інстраструктури та принцип їхньої взаємодії. В статті пояснюється, як організовується взаємодія приватної та публічної частини гібридної хмари, але водночас не розкривається роль мережевого екрана. В публікації [7] розглядається приклад мультихмарного розгортання Kubernetes з використанням сценаріїв Terraform. Сценарій, який створив автор, передбачав розгортання, зокрема, у справжній гібридній хмарі, але автор не використав її через те, що провайдери приватних хмар працюють переважно за моделлю передплати, що ускладнює їх використання у навчальних цілях.

У попередньому дослідженні [8] було зібрано дані про гібридні хмарні конфігурації з різних джерел. В ньому було зібрано структуровані дані про 158 гібридних хмарних конфігурацій, в яких сумарно знаходяться дані про 1207 хмарних сервісів. Для видобутку застосовано засоби штучного інтелекту, зокрема, машинний зір [9] для розпізнавання схем, а також можливості LLM-моделей для структурування даних про хмарні конфігурації [10].

Перед дослідженням ставилися наступні цілі:

– подальше доповнення та структуризація даних, зібраних у попередньому дослідженні [8];

– отримання експертних рекомендацій щодо налаштування мережевого екрана;

– аналіз отриманих даних і рекомендацій експертів.

Виклад основного матеріалу

Після проведення першого дослідження було зібрано колекцію хмарних конфігурацій, в якій кожна конфігурація була представлена у вигляді окремих файлів JSON. Ієрархія об'єктів зображена на рис. 1. Гібридна хмара передбачає існування як мінімум двох "підхмар", одна з яких приватна, а інша – публічна. Під час збору даних вони могли бути отримані із сценаріїв розгортання, таких як Terraform. За допомогою LLM-моделей OpenAI GPT-4.1, OpenAI GPT-5 mini та DeepSeek-v3.1 сценарії перетворювались у статичні дані у форматі JSON [8].

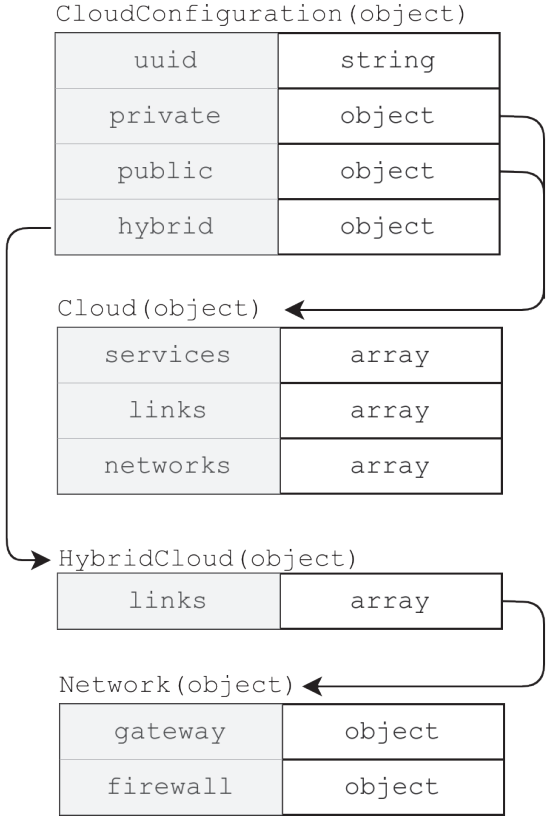

Рис. 1. Ієрархія об'єктів хмарної конфігурації у форматі JSON

Для кожного виявленого сервіса було визначено тип, схожі типи об'єднано. В цілому виділено 20 типів, зокрема, веб-додатки, API-додатки, бази даних, проксі-сервери, кеші, DNS-сервери та інші [8]. Розподіл сервісів за основними типами наведено на рис. 2.

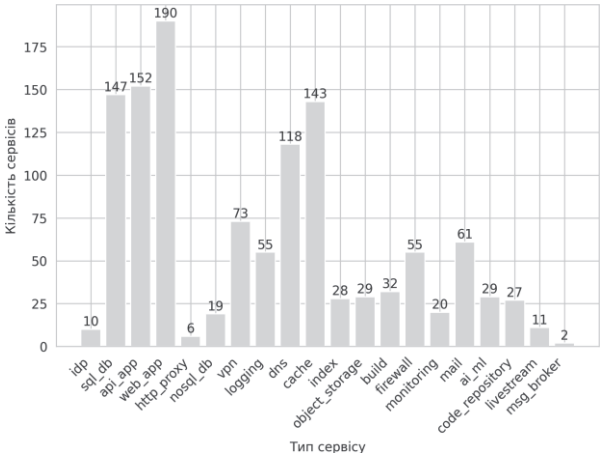

Рис. 2. Розподіл сервісів за основними типами

У процесі обробки вхідних даних LLM-моделями було визначено додаткові параметри, які вони мали знайти або визначити самостійно: підтримка кешування (cacheable), реплікації (replicable), масштабування (extensible) та запуску на вимогу (ondemand). Приклад одного із сервісів наведено на рис. 3.

Parameter	Value
name	pri_service_api
type	api_app
exposed	false
cacheable	true
replicable	false
extensible	true
ondemand	false
location	private
ports	3000/tcp

Рис. 3. Атрибути хмарного сервісу

Іншим завданням було визначення зв'язків між сервісами. Значна частина сервісів уже мали зв'язки, особливо у випадках, коли вони були явно передбачені у сценаріях Terraform (за допомогою аргументу depends_on) чи були сполучені лініями на зображеннях, де було створено зв'язки (масив "links" на рис. 1). Для ви-

значення неявних зв'язків, зокрема, у випадках виявлення відособлених сервісів, було застосовано LLM-модель Claude Sonnet 4.5. Також за допомогою даної моделі визначено такі параметри, як можливість захисту TLS (is_ssl) та використання бінарного протоколу (is_binary – наприклад, для HTTP буде значення false, оскільки основна інформація запитів має текстовий формат). Приклад прив'язки зв'язку до сервісу зображено на рис. 4. Залежність визначається за допомогою параметру "is_dependency".

Parameter	Value
name	pri_service_api
type	api_app
exposed	false
cacheable	true
replicable	false
extensible	true
ondemand	false
location	private
remote_role	client
ports	3000/tcp
Parameter	**Value**
is_ssl	false
is_binary	true
is_dependency	true

Рис. 4. Параметри пов'язаного сервісу та характер зв'язку

Для визначення патернів пріоритетності для створення правил мережевого екрана WAF залучено допомогу трьох незалежних експертів із конфігурування обчислювальних хмар – DevOps-інженерів. Для визначення патернів пріоритетності введено два показники пріоритетності: пріоритет доступності та пріоритет покриття мережевим екраном WAF. Для зручності експертів введено 10-ти бальну шкалу оцінювання кожного показника для кожного окремого сервісу (цілі числа від 0 до 10). Для цього для кожного сервісу LLM-моделлю підготовлено коротку анотацію.

Пріоритет доступності визначає наскільки важливою є безперешкодна доступність веб-сайту чи додатку для користувачів. Найбільший пріоритет передбачає доступність хмарного сервісу навіть в умовах DDoS-атак. А пріоритет покриття мережевим екраном WAF визначає критичність застосування додаткових заходів захисту для попередження несанкціонованого доступу чи експлуатації вразливостей. Для визначення цих показників створено анкетування для експертів.

Для визначення пріоритету доступності експерти мали дати ствердну або заперечну відповідь на наступні питання: "для роботи сервісу має бути дотримання SLA на рівні не менш ніж 99.95%" (2 бали); "сервіс має працювати цілодобово" (2 бали); "сервіс має бути доступним для користувачів в умовах DDoS-атаки" (2 бали); "сервіс має бути доступним для індексування пошуковими ботами" (2 бали); "сервіс має бути доступним для соціальних мереж та месенджерів" (1 бал); "сервіс має бути доступним для користувачів без додаткових перевірок" (1 бал).

Для визначення пріоритету покриття мережевим екраном WAF експертам було поставлено такі питання: "для доступу до сервісу користувач має використовувати VPN-з'єднання" (2 бали); "для доступу до сервісу користувач має пройти звичайну автентифікацію" (2 бали); "для доступу до сервісу користувач має пройти двофакторну автентифікацію" (1 бал); "доступ до сервісів відкриває доступ до інших сервісів передбачених безпосередніми зв'язками" (1 бал); "доступ до сервісів відкриває доступ до інших сервісів поза безпосередніми зв'язками" (1 бал); "доступ дозволено лише технічним співробітникам" (1 бали); "для окремих методів REST API сервісу передбачений додатковий захист" (1 бал); "існує можливість порушення роботи сервісу у разі важких запитів" (1 бал).

У процесі проведення опитування сервіси були розподілені між різними експертами у довільному порядку. Опитувальник виводив коротку анотацію хмарного сервісу, його технічні параметри (рис. 3), а також перелік сервісів, з якими він має зв'язок (рис. 4). За результатами опитування визначено рейтинг пріоритетів доступності та покриття мережевим екраном, який наведено у рис. 5 та таблиці 1.

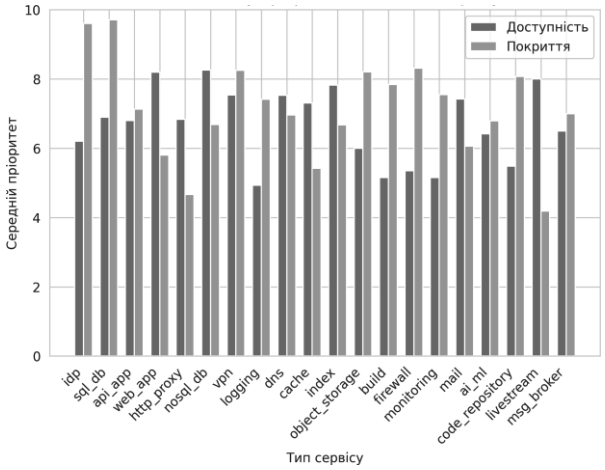

Рис. 5. Середні пріоритети доступності та покриття за типом сервісів

У всіх сервісах присутній булевий атрибут "exposed", який визначає наявність зовнішнього доступу. Спосіб надання зовнішнього доступу до сервісу визначався у масиві "network" кожної відповідної частини хмари. Існувало 4 типи забезпечення зовнішнього доступу до сервісів (рис. 6): реверсивний проксі з WAF-захистом (waf_protected), реверсивний проксі без WAF (proxy_exposed), доступ за допомогою правил переадресації портів NAT (nat_exposed), а також відсутність зовнішнього доступу (no_network_rules).

Реверсивний проксі-сервер використовується переважно для доступу до сервісів, доступ до який працює за протоколом HTTP/HTTPS. Для класичних протоколів, таких як LDAP, DNS та SMTP реверсивний проксі складно застосувати, тому доступ до них найчастіше забезпечується через переадресацію портів NAT.

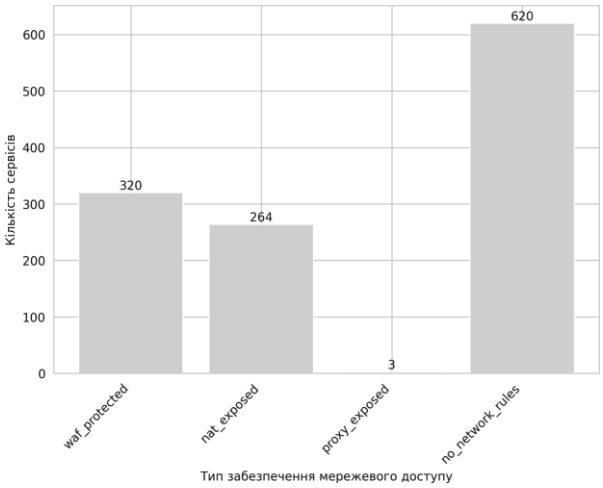

Рис. 6. Розподіл сервісів за типом забезпечення мережевого доступу

Таблиця 1

Пріоритети доступності та покриття за типом сервісів

	Пріоритети	
Тип сервісу	Доступність	Покриття мережевим екраном
web_app	8.2	5.8
api_app	6.8	7.1
sql_db	6.9	9.7
nosql_db	8.3	6.7
http_proxy	6.8	4.7
msg_broker	6.5	7.0
livestream	8.0	4.2
code_repository	5.5	8.1
logging	4.9	7.4
index	7.8	6.7
object_storage	6.0	8.2
cache	7.3	5.4
build	5.2	7.8
ai_ml	6.4	6.8
mail	7.4	6.1
dns	7.5	7.0
idp	6.2	9.6
firewall	5.3	8.3
monitoring	5.2	7.5
vpn	7.5	8.2

Пріоритети експертів також визначені за типами мережевого доступу та зображені на рис. 7 і наведені у таблиці 2.

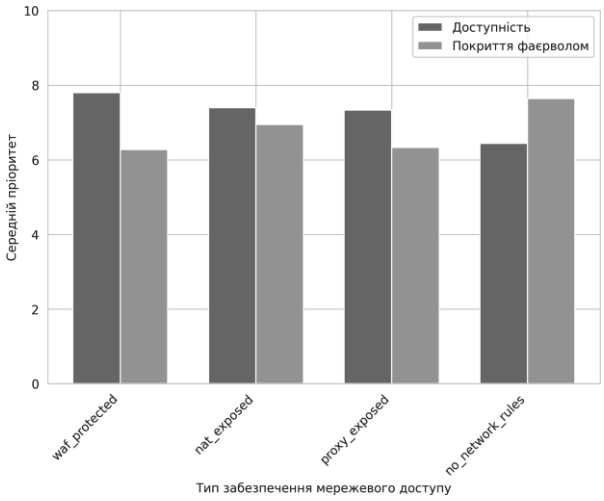

Рис. 7. Пріоритети за типом забезпечення мережевого доступу

Крім типу мережевого доступу на рішення експертів могли впливати також інші атрибути, зокрема, розширюваність, реплікація, запуск на вимогу та підтримка кешування (рис. 8).

Таблиця 2

Пріоритети за типом забезпечення мережевого доступу

	Пріоритети	
Зв'язок	Доступність	Покриття мережевим екраном
waf_protected	6.4	7.6
nat_exposed	7.4	6.9
proxy_exposed	7.3	6.3
no_network_rules	7.8	6.3

Хоча горизонтальне масштабування сервісу чи запуск на вимогу дають значну гнучкість, але для них також має бути налаштоване автоматичне створення правил мережевого екрана. Це актуально не лише коли до сервісів можливий доступ зовні, а й тоді коли до них здійснюється опосередкований доступ через інші сервіси. В анкеті експертів демонструвались зв'язки із сусідніми сервісами, що теж могло вплинути на їхні рішення.

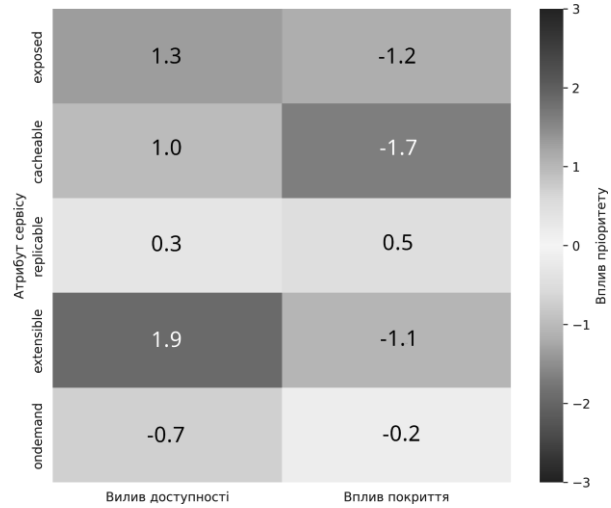

Рис. 8. Теплова карта впливу атрибутів сервісів на пріоритети експертів

Зв'язок між рішеннями експертів та станом атрибутів сервісів також наведено у вигляді теплової карти на рисунку 9.

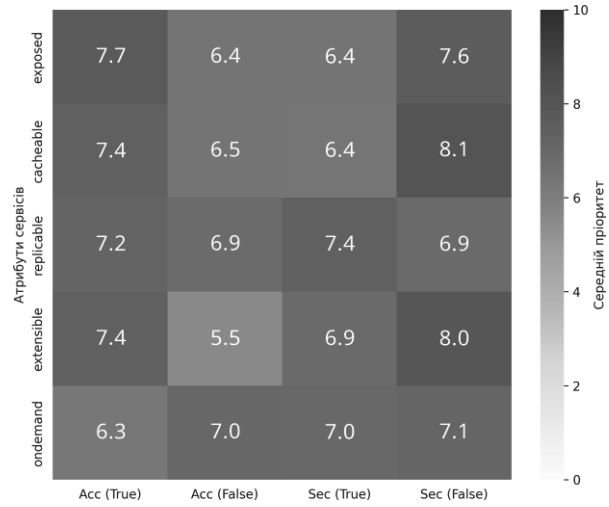

Рис. 9. Теплова карта середніх пріоритетів за станом атрибутів

У визначенні пріоритетів експертам важливо звертати увагу на пов'язані сервіси та на характер зв'язків. Як можна побачити з рисунку 10 та з таблиці 3, експерти виділили високими пріоритетами сильні залежні зв'язки між сервісами додатків та сервісами-контейнерами даних, такі як бази даних, об'єктні сховища та кеші даних. Це зумовлено високим рівнем зв'язності між ними [2].

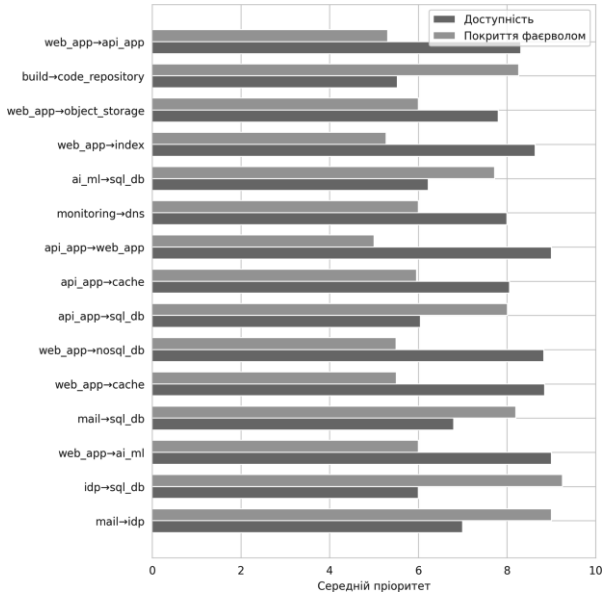

Рис. 10. Пріоритети зв’язків за типами сервісів

Хоча зв’язність у інженерії програмного забезпечення застосовується переважно до програмних компонентів [1], і у хмарних обчисленнях зв'язність стосується переважно зв’язків між сервісами-додатками, але в контексті даного дослідження зв’язність застосовується і до сервісів-контейнерів даних.

Таблиця 3

Залежності зв’язків між типами сервісів із пріоритетами сервісів

		Пріоритети	
Зв’язок		Доступність	Покриття мережевим екраном
1	web_app->api_app	8.3	5.3
2	build->code_repository	5.5	8.3
3	web_app->object_storage	7.8	6.0
4	web_app->index	8.6	5.3
5	ai_ml->sql_db	6.2	7.7
6	monitoring->dns	8.0	6.0
7	api_app->web_app	9.0	5.0
8	api_app->cache	8.1	6.0
9	api_app->sql_db	6.0	8.0
10	web_app->nosql_db	8.8	5.5
11	web_app->cache	8.8	5.5
12	mail->sql_db	6.8	8.2
13	web_app->nosql_db	8.8	5.5
14	idp->sql_db	6.0	9.2
15	mail->idp	7.0	9.0

Патерни пріоритетності мережевого екрана WAF, створені на основі пріоритетів доступності та покриття мережевим екраном, являють собою метаправила, описані у вигляді директив у форматі JSON. У подальшому їх можна використати для надання інструкцій великим мовним моделям, використовувати для моделювання мережевих екранів у хмарах, а також для навчання моделей штучного інтелекту.

Висновки

У статті запропоновано два показники для створення патернів пріоритетності мережевого екрана WAF – пріоритет доступності та пріоритет покриття мережевим екраном. У дослідженні на основі раніше зібраних даних [8] проведено опитування експертів. На основі анкетних даних розраховано ці показники, та з їхньою допомогою зібрано дані про пріоритетність типів хмарних сервісів, їхніх атрибутів та зв’язків.

У подальшому планується перевірити ефективність роботи патернів пріоритетності мережевого екрана WAF, провести моделювання роботи зібраних хмарних конфігурацій у хмарному середовищі та з’ясувати, які види моделювання найкращі для роботи мережевих екранів WAF у гібридному хмарному середовищі.

Застосування ШІ. За допомогою LLM-моделей OpenAI GPT-4.1, OpenAI GPT-5 mini та DeepSeek-v3.1 із текстових даних сформовано уніфіковані описові файли JSON з гібридними хмарними конфігураціями. Моделі визначили типові атрибути хмарних сервісів. Доповнення забраклих зв’язків між сервісами виконано за допомогою LLM-моделі Claude Sonnet 4.5.

Література

1. Prasetyo S. E., Haeruddin H., Ariesryo K. Website Security System from Denial of Service attacks, SQL Injection, Cross Site Scripting using Web Application Firewall.

Antivirus: Jurnal Ilmiah Teknik Informatika. 2024. Т. 18, № 1. С. 27–36. DOI: 10.35457/antivirus.v18i1.3339

2. Малініч І. П., Іванчук Я. В. Особливості розміщення мікросервісів систем управління навчанням у гібридних хмарах. *Системні технології*. 2025. № 3(158). С. 157–170. DOI: 10.34185/1562-9945-3-158-2025-16
3. Малініч І. П., Іванчук Я. В. Показники оцінки мережевої доступності та зв'язності інформаційних систем у обчислювальних хмарах. *Інформаційні системи та технології: результати і перспективи: матеріали 2-ої Міжнародної науково-практичної конференції, 5 березня 2025 р., Київ, Україна*. Київ: ФІТ КНУТШ, 2025. С. 248–251.
4. Beigi-Mohammadi N., Shtern M., Litoiu M. Adaptive load management of web applications on software defined infrastructure. IEEE Transactions on Network and Service Management. 2020. Vol. 17, No. 1. P. 488–502. DOI: 10.1109/TNSM.2019.2948969.
5. Малярчук І. І., Смолинець М. А. Гібридні хмарні рішення як шлях до балансу між контролем і гнучкістю в діяльності сучасних ІТ підприємств. *Актуальні питання економічних наук*. 2025. № 10. DOI: 10.5281/zenodo.15292588.
6. Коваленко А., Ляшенко О., Ярошевич Р. Порівняльний аналіз організації хмарної інфраструктури. *Advanced Information Systems*. 2021. Т. 5, № 2. С. 108–113. DOI: 10.20998/2522-9052.2021.2.15.
7. Downey B. Building a Kubernetes Hybrid Cloud with Terraform [Електронний ресурс]. ITNEXT. Режим доступу: https://itnext.io/building-a-kubernetes-hybrid-cloud-with-terraform-fe15164b35fb (дата звернення: 09.06.2025).
8. Малініч І. П., Іванчук Я. В. Збір та обробка конфігурацій гібридних хмар. *Матеріали XVIII міжнародної науково-практичної конференції "Інформаційні технології і автоматизація – 2025", 30–31 жовтня 2025 р., Одеса*. Одеса: Видавництво ОНТУ, 2025. С. 847–850.
9. Ajay Kamal G., Subha A. Object Detection Using Vertex AI AutoML. *International Journal of Innovative Research in Science Engineering and Technology*. 2025. Т. 14, № 4. С. 9304–9312.
10. Yarovyi A., Kudriavtsev D., Baraban S., Ozeranskyi V., Krylyk L., Smolarz A., Karnakova G. Information technology in creating intelligent chatbots. *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments. 2019*. Art. no. 1117627. DOI: 10.1117/12.2537415.
11. Yarovii A., Kudriavtsev D., Olena P. Improving the accuracy of text message recognition with an intelligent chatbot information system. *Proc. 2020 IEEE 15th Int. Conf. Comput. Sci. Inf. Technol. (CSIT), Zbarazh, Ukraine. 2020*. С. 76–79. DOI: 10.1109/CSIT49958.2020.9322036.

References

1. Prasetyo, S.E., Haeruddin, H. & Ariesryo, K., 2024. Website Security System from Denial of Service attacks, SQL Injection, Cross Site Scripting using Web Application Firewall. *Antivirus: Jurnal Ilmiah Teknik Informatika,* 18(1), pp.27–36. https://doi.org/10.35457/antivirus.v18i1.3339
2. Malinich, I.P. & Ivanchuk, Y.V., 2025. Features of microservice deployment in learning management systems in hybrid clouds. *Systemni Tekhnolohii*, 3(158), pp.157–170. [in Ukrainian] https://doi.org/10.34185/1562-9945-3-158-2025-16
3. Malinich, I.P. & Ivanchuk, Y.V., 2025. Indicators for assessing network availability and connectivity of information systems in cloud computing. *In: Information Systems and Technologies: Results and Prospects: Proceedings of the 2nd International Scientific and Practical Conference, 5 March 2025, Kyiv, Ukraine*. Kyiv: FIT KNUTSH, pp.248–251. [in Ukrainian]
4. Beigi-Mohammadi, N., Shtern, M. & Litoiu, M., 2020. Adaptive load management of web applications on software defined infrastructure. IEEE Transactions on Network and Service Management, 17(1), pp.488–502. https://doi.org/10.1109/TNSM.2019.2948969
5. Maliarchuk, I.I. & Smolynets, M.A., 2025. Hybrid cloud solutions as a way to balance control and flexibility in the activities of modern IT enterprises. [in Ukrainian]. *Aktualni Pytannia Ekonomichnykh Nauk*, 10. https://doi.org/10.5281/zenodo.15292588
6. Kovalenko, A., Lyashenko, O. & Yaroshevych, R., 2021. Comparative analysis of cloud infrastructure organization. *Advanced Information Systems*, 5(2), pp.108–113. [in Ukrainian].

https://doi.org/10.20998/2522-9052.2021.2.15
7. Downey, B., 2025. Building a Kubernetes Hybrid Cloud with Terraform [online]. *ITNEXT*. Available at: https://itnext.io/building-a-kubernetes-hybrid-cloud-with-terraform-fe15164b35fb [Accessed 09 June 2025].
8. Malinich, I.P. & Ivanchuk, Y.V., 2025. Collection and processing of hybrid cloud configurations. *Proceedings of the 18th International Scientific and Practical Conference "Information Technologies and Automation – 2025"*, 30–31 October 2025, Odesa. Odesa: ONTU Publishing House, pp. 847–850. [in Ukrainian].
9. Ajay Kamal, G. & Subha, A., 2025. Object Detection Using Vertex AI AutoML. *International Journal of Innovative Research in Science Engineering and Technology*, 14(4), pp.9304–9312.
10. Yarovyi, A., Kudriavtsev, D., Baraban, S., Ozeranskyi, V., Krylyk, L., Smolarz, A. & Karnakova, G., 2019. Information technology in creating intelligent chatbots. *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments*, Art. no. 1117627. https://doi.org/10.1117/12.2537415
11. Yarovii, A., Kudriavtsev, D. & Olena, P., 2020. Improving the accuracy of text message recognition with an intelligent chatbot information system. *Proc. 2020 IEEE 15th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Zbarazh, Ukraine, pp.76–79. https://doi.org/10.1109/CSIT49958.2020.9322036.

Одержано: 03.11.2025
Внутрішня рецензія отримана: 09.11.2025
Зовнішня рецензія отримана: 12.11.2025

Про авторів:

Малініч Ілля Павлович,
асистент кафедри комп'ютерних наук,
http://orcid.org/0000-0002-5862-3732

Іванчук Ярослав Володимирович,
доктор технічних наук,
професор кафедри комп'ютерних наук,
http://orcid.org/0000-0002-4775-6505

Місце роботи авторів:

Вінницький національний технічний університет,
тел. +38 0432 651903
E-mail: vntu@vntu.edu.ua
vntu.edu.ua