https://doi.org/10.31891/2307-5732-2026-365-26
УДК 004.056

СІРЕНКО ОЛЕКСАНДР
Державний університет інтелектуальних технологій і зв'язку
https://orcid.org/0000-0001-9751-2544
emai: olexandr.sirenko@gmail.com

ПОВЕДІНКОВЕ ПРОФІЛЮВАННЯ КОНТЕЙНЕРІВ ДЛЯ ВИЯВЛЕННЯ КІБЕРАТАК НА ОСНОВІ СИСТЕМНИХ ВИКЛИКІВ

У статті досліджується можливість виявлення кібератак у контейнеризувати середовищах на основі поведінкового профілювання робочих навантажень. Для збору телеметрії системних викликів використано інструмент Falco з eBPF-движком, що дозволяє фіксувати події виконання процесів, мережеві з'єднання та файлові операції у контейнерах у режимі, наближеному до реального часу. Запропоновано підхід до формування поведінкових профілів контейнерів із використанням часової агрегації подій та rule-based класифікації. Проведено експериментальні дослідження для сценаріїв нормальної роботи сервісу та типових атак, зокрема інтерактивної оболонки, зворотної оболонки та емуляції криптомонета. Оцінку якості виявлення здійснено за метриками Precision та Recall. Отримані результати демонструють принципову придатність підходу, а також виявляють обмеження простих порогових правил, що обґрунтовує необхідність подальших досліджень.

Ключові слова: *контейнерна безпека, поведінкове профілювання, виявлення вторгнень.*

SIRENKO OLEKSANDR
State University of Intelligent Technologies and Telecommunications

BEHAVIORAL PROFILING OF CONTAINERS FOR CYBERATTACK DETECTION BASED ON SYSTEM CALLS

This paper investigates the feasibility of detecting cyberattacks in containerized environments based on behavioral profiling of container workloads. The growing adoption of container technologies in cloud-native infrastructures has increased the attack surface of modern systems and highlighted the need for effective runtime security mechanisms. In this context, behavioral analysis of system activity represents a promising direction for intrusion detection. System call telemetry is collected using the Falco security monitoring tool with an eBPF-based engine, which enables near real-time observation of process execution events, network connections, and file system operations inside containers with minimal performance overhead. A method for constructing behavioral profiles of containers is proposed, combining time-based aggregation of system events into fixed-length windows with a deterministic rule-based classification approach. This allows transforming low-level system call data into higher-level behavioral representations suitable for analysis. Experimental studies are conducted for scenarios representing normal service operation as well as typical attack patterns, including interactive shell execution, reverse shell activity, and cryptominer emulation. The quality of attack detection is evaluated using standard classification metrics, namely Precision and Recall. The obtained results demonstrate the fundamental applicability of the proposed approach for identifying anomalous container behavior. At the same time, they reveal significant limitations of simple threshold-based rules when applied in containerized environments, where different attack scenarios may exhibit overlapping behavioral characteristics. These findings substantiate the necessity of further research aimed at extending the feature set and incorporating more adaptive classification methods for improving detection accuracy in containerized systems.

Keywords: *container security, behavioral profiling, intrusion detection.*

Стаття надійшла до редакції / Received 04.03.2026
Прийнята до друку / Accepted 24.03.2026
Опубліковано / Published 28.05.2026

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Контейнеризація є однією з ключових технологій сучасних хмарних та агросервісних архітектур. Разом із перевагами у вигляді масштабованості та ізоляції, контейнерні середовища створюють нові вектори атак, зокрема шляхом компрометації контейнера та подальшого виконання шкідливого коду. Традиційні засоби захисту, орієнтовані на віртуальні машини або мережевий периметр, часто виявляються недостатньо ефективними для виявлення атак усередині контейнерів.

Одним із перспективних напрямків є поведінкове профілювання робочих навантажень на основі аналізу системних викликів та інших низькорівневих подій виконання. Такий підхід дозволяє виявляти аномалії, що не залежать від сигнатур конкретних атак.

Наукова новизна роботи полягає у розробці відтворюваного експериментального методу поведінкового профілювання контейнерів на основі агрегації системних викликів у часові вікна та у кількісному аналізі обмежень rule-based підходу в умовах контейнеризовано го середовища. На відміну від більшості існуючих робіт, орієнтованих на застосування методів машинного навчання, у даній роботі запропоновано прозорий baseline-підхід, який дозволяє формалізувати характерні поведінкові ознаки контейнерів та використовується як еталон для подальших порівняльних досліджень.

Аналіз досліджень та публікацій

Системні виклики є ключовою ланкою взаємодії між прикладним програмним забезпеченням та ядром операційної системи, що робить їх цінним джерелом інформації для задач виявлення вторгнень. Формування поведінкових профілів на основі syscall-послідовностей дозволяє відрізняти штатну активність контейнерів від підозрілої.

У роботі Б. Скориновича та Ю. Лаха продемонстровано ефективність застосування методів машинного навчання для аналізу системних викликів у контейнеризованих середовищах із використанням механізмів eBPF [1]. Отримані результати свідчать про досягнення точності виявлення понад 99 % за одночасного зниження частоти хибних спрацювань до рівня близько 2 %, без помітного негативного впливу на продуктивність системи, що є критично важливим для практичного впровадження таких рішень. Важливим напрямом розвитку є поєднання аналізу системних викликів з іншими джерелами телеметрії. Зокрема, у дослідженні Р. Піткара запропоновано комплексний підхід, який інтегрує журнали Kubernetes, виклики API та аналіз мережевого трафіку для виявлення відхилень від штатної поведінки [2]. Запропонована мультиджерельна модель демонструє кращі показники точності та повноти виявлення аномалій порівняно з традиційними rule-based засобами безпеки.

Останні наукові роботи зосереджені на адаптації методів поведінкового аналізу до специфіки контейнерних екосистем. Так, Хамза та співавтори запропонували модель виявлення об'ємних атак, орієнтовану на середовища Інтернету речей, яка може бути застосована і для захисту контейнеризованих платформ [3]. Підхід базується на контролі відповідності мережевої поведінки визначеним профілям та аналізі пакетів, що підвищує видимість аномальної активності. Поглиблений аналіз послідовностей системних викликів розглянуто в роботі Камалудділна, де досліджується використання прихованих марковських моделей і рекурентних нейронних мереж типу LSTM для виявлення шкідливого програмного забезпечення в контейнерних середовищах [4]. Отримані результати підтверджують доцільність застосування часових моделей для аналізу динамічної поведінки процесів.

Незважаючи на переваги поведінкового профілювання, його практична реалізація пов'язана з низкою технічних труднощів. Зокрема, у роботі Канелли та співавторів показано, що ручне створення Seccomp-фільтрів для різних контейнеризованих застосунків є складним і погано масштабується [5]. Автоматизація процесу генерації таких фільтрів розглядається як необхідна умова підвищення рівня безпеки без надмірного навантаження на розробників. Окремою проблемою є експлуатаційна складність впровадження систем виявлення аномалій у реальному часі. Дослідження Гаджбхіє та співавторів акцентує увагу на викликах, що виникають під час розгортання механізмів оперативного реагування у Kubernetes-кластерах, які характеризуються високою динамічністю та частими змінами конфігурації [6].

Подальший розвиток поведінкового профілювання у сфері безпеки контейнерів пов'язується з тіснішою інтеграцією алгоритмів машинного навчання з традиційними засобами захисту. У низці робіт відзначається потенціал використання edge-computing для обробки даних безпеки безпосередньо поблизу джерел їх генерації, що дозволяє зменшити затримки реагування та підвищити стійкість систем до складних атак [2]. Водночас актуальним залишається завдання безперервної адаптації поведінкових моделей до нових векторів атак. Консолідація результатів різних підходів до виявлення аномалій розглядається як перспективний шлях до побудови більш універсальних і надійних систем захисту [1], [4].

Поведінкове профілювання контейнерів на основі аналізу системних викликів є ефективним підходом до виявлення кібератак у сучасних розподілених середовищах. Аналіз сучасних досліджень показує тенденцію до використання мультиджерельних даних та методів машинного навчання, що дозволяє підвищити точність детекції та зменшити кількість хибних спрацювань [1], [2]. Разом із тим, залишаються відкритими питання автоматизації формування профілів безпеки та зниження експлуатаційної складності таких систем [5], [6], вирішення яких є ключовим для підвищення рівня захищеності контейнеризованих інфраструктур.

Таким чином, можна дійти висновку що питання формування цілісного поведінкового профілю та аналізу часової динаміки подій потребують подальшого дослідження, особливо в контексті простих і відтворюваних експериментів.

Формулювання цілей статті

Об'єктом даного дослідження є контейнеризовані програмні середовища, що використовуються для розгортання та виконання прикладних сервісів у сучасних хмарних та мікросервісних архітектурах. Предметом дослідження є поведінка контейнерів на рівні системних викликів операційної системи та можливість використання цієї поведінки для виявлення кібератак.

Метою даної роботи є дослідження можливостей використання системних викликів контейнерів для формування поведінкових профілів, а також розробка та експериментальна перевірка методу поведінкового профілювання контейнерів на їх основі з метою виявлення типових сценаріїв атак у контейнеризованому середовищі, зокрема запуску інтерактивної оболонки, зворотної оболонки та емуляції криптомайнера.

Виклад основного матеріалу

Задача виявлення атак у контейнерних середовищах суттєво відрізняється від аналогічних задач для традиційних операційних систем. Контейнери, як правило, виконують обмежений набір функцій, мають спрощену структуру процесів та не передбачають інтерактивної взаємодії з користувачем. У зв'язку з цим будь-які відхилення від типової моделі виконання можуть свідчити про потенційну компрометацію контейнера. Крім того, контейнери мають динамічний життєвий цикл: вони можуть часто перезапускатися, масштабуватися або створюватися автоматично. Це унеможливлює використання довготривалих поведінкових профілів, характерних для традиційних систем, та вимагає аналізу короткочасних інтервалів виконання.

Запропонований метод виявлення атак у контейнеризованому середовищі ґрунтується на аналізі системних викликів, що генеруються контейнером під час його виконання, та подальшій класифікації агрегованої поведінки у фіксованих часових інтервалах.

На першому етапі здійснюється збір телеметрії системних викликів за допомогою інструменту Falco з

eBPF-движком. У процесі моніторингу реєструються події виконання процесів, встановлення мережевих з'єднань та операції запису у файлову систему. Усі події зберігаються у вигляді журналу у форматі JSON Lines, де кожен запис відповідає окремій події.

На другому етапі потік подій агрегується у фіксовані часові вікна тривалістю 10 секунд. Кожне часове вікно розглядається як незалежний об'єкт аналізу, що дозволяє враховувати короткочасні сплески активності, характерні для атак у контейнерних середовищах.

На третьому етапі для кожного часового вікна виконується обчислення поведінкових ознак, які узагальнюють інтенсивність та різноманітність активності контейнера. До таких ознак належать: кількість запусків процесів, кількість мережевих з'єднань, кількість операцій запису у файлову систему, кількість унікальних виконуваних процесів, а також наявність стандартних сповіщень Falco про запуск інтерактивної оболонки.

На четвертому етапі здійснюється класифікація поведінки контейнера за допомогою детермінованого rule-based алгоритму. Для кожного часового вікна послідовно перевіряється набір експертно заданих умов, що відповідають визначеним сценаріям поведінки: нормальна робота сервісу, запуск інтерактивної оболонки, зворотна оболонка та емуляція криптомайнера. У разі одночасного виконання кількох умов застосовується фіксований пріоритет правил, що забезпечує однозначність рішення.

На завершальному етапі результати класифікації порівнюються з істинними мітками сценаріїв, сформованими на основі маркерних подій, після чого виконується оцінка якості виявлення за стандартними метриками багатокласової класифікації.

Запропонований алгоритм належить до класу behavior-based intrusion detection systems, не використовує сигнатур атак та не потребує попереднього навчання моделей. Це забезпечує відтворюваність експерименту та прозорість інтерпретації отриманих результатів, що є важливим для експериментальних досліджень у контейнеризованих середовищах.

Для збору телеметрії системних викликів використовувався інструмент Falco з eBPF-модулем, що забезпечує перехоплення подій на рівні ядра операційної системи без значного впливу на продуктивність контейнерів. Falco дозволяє реєструвати події виконання процесів, мережеві з'єднання та операції з файловою системою. У процесі експерименту реєструвалися такі типи подій:

1. Виконання процесів у контейнері (системний виклик execve).
2. Встановлення мережевих з'єднань (системний виклик connect).
3. Операції запису у файлову систему.
4. Стандартні сповіщення Falco про підозрілу активність.

Журнали подій зберігалися у форматі JSON Lines, де кожен рядок відповідає окремій події. Такий формат полегшує подальший аналіз та автоматичну обробку даних. Для мінімізації затримок запису подій внутрішня буферизація Falco була вимкнена, що дозволило фіксувати події у режимі, наближеному до реального часу. Як джерела даних використовувалися події запуску процесів, встановлення мережевих з'єднань, операції запису у файлову систему, а також стандартні сповіщення Falco про підозрілу активність.

Для переходу від окремих подій до аналізу поведінки контейнера застосовувалася агрегація подій у фіксовані часові вікна тривалістю 10 секунд. Кожне вікно розглядалося як незалежний об'єкт аналізу. Для експериментальної перевірки методу було реалізовано кілька сценаріїв поведінки контейнера: нормальна робота сервісу, запуск інтерактивної оболонки, зворотна оболонка та емуляція криптомайнера. Кожен сценарій запускався окремо та супроводжувався явною маркерною розміткою. Маркерні події створювалися шляхом виконання спеціальних команд у контейнері, що фіксувалися Falco та дозволяли точно визначити часові межі кожного сценарію. Такий підхід забезпечив коректне формування істинних міток класів для подальшої оцінки якості класифікації. Для кожного часового вікна обчислювався набір поведінкових ознак, що характеризують інтенсивність та різноманітність активності контейнера. До таких ознак належали:

1 Кількість запусків процесів.
2 Кількість мережевих з'єднань.
3 Кількість операцій запису у файлову систему.
4 Кількість унікальних виконуваних процесів.
5 Наявність сповіщень Falco про запуск інтерактивної оболонки.

Обрані ознаки є агрегованими та не залежать від конкретного прикладного коду контейнера, що дозволяє застосовувати метод до різних типів сервісів. Класифікація поведінки контейнера у межах запропонованого методу здійснюється на основі детермінованого rule-based підходу, в якому кожному класу поведінки відповідає набір логічних умов, сформованих експертним шляхом. Ознакою запуску інтерактивної оболонки вважається наявність у межах часового вікна подій виконання командної оболонки або стандартних сповіщень Falco, пов'язаних із shell-активністю. Сценарій зворотної оболонки визначається як поєднання shell-активності з мережевими з'єднаннями, ініційованими контейнером. Емуляція криптомайнера характеризується підвищеною інтенсивністю виконання процесів та файлових операцій у межах одного часового вікна за відсутності інтерактивної оболонки. За відсутності зазначених ознак поведінка контейнера відноситься до класу нормальної роботи сервісу.

У випадку одночасного виконання кількох умов застосовується фіксований пріоритет правил, у якому сценарії, пов'язані з shell-активністю, мають вищий пріоритет порівняно з іншими класами. Було реалізовано такі сценарії:

1. Нормальна робота сервісу - типове HTTP-навантаження на контейнер Nginx.
2. Інтерактивна оболонка - запуск shell усередині контейнера.
3. Зворотна оболонка (reverse shell) - ініціація вихідного мережевого з'єднання з контейнера.
4. Емуляція криптомайнера - створення тривалого CPU-навантаження.

Наявність інтерактивної оболонки розглядалася як основний індикатор підозрілої активності. Поєднання shell-активності з мережевими з'єднаннями інтерпретувалося як ознака зворотної оболонки. Підвищена інтенсивність виконання процесів або файлових операцій розглядалася як можливий індикатор криптомайнера. У разі відсутності зазначених ознак поведінка вважалася нормальною.

Метод належить до класу behavior-based intrusion detection systems та не використовує сигнатур атак або навчання моделей на великих обсягах даних. Рішення приймаються на основі експертно заданих правил, що дозволяє забезпечити відтворюваність експерименту та прозорість інтерпретації результатів. Ключовим елементом методу є агрегація системних викликів у часові вікна, що дозволяє враховувати не окремі події, а їх сукупність, яка відображає загальний характер активності контейнера.

Оцінка ефективності запропонованого методу здійснювалася за стандартними метриками багатокласової класифікації: Precision, Recall та F1-score. Дані метрики дозволяють окремо оцінити достовірність спрацьовувань, повноту виявлення та баланс між цими показниками. Додатково використовувалася матриця помилок (confusion matrix), яка дозволила проаналізувати характер та систематичність помилок класифікації.

На основі сформованих ознак застосовано rule-based підхід для класифікації часових вікон на чотири класи: normal, shell, reverse shell та miner. Таблиця 1 містить матрицю помилок класифікації поведінкових сценаріїв контейнерів, побудовану за результатами експерименту з використанням фіксованих часових вікон. Рядки таблиці відповідають істинним класам поведінки, тоді як стовпці відображають класи, визначені алгоритмом. Аналіз матриці помилок дозволяє оцінити характер і напрям помилок класифікації. Зокрема, спостерігається значна плутанина між сценаріями запуску інтерактивної оболонки та зворотної оболонки, що зумовлено подібністю їх поведінкових ознак на рівні системних викликів. Водночас нормальна поведінка сервісу не була коректно ідентифікована жодного разу, що свідчить про недостатню дискримінативність використаних правил для цього класу.

Таблиця 1

Матриця помилок класифікації поведінкових сценаріїв контейнерів

Істинний клас	normal	shell	reverse_shell	miner
normal	0	2	2	0
shell	0	1	1	0
reverse_shell	0	4	0	0
miner	0	1	1	0

Таблиця 2 узагальнює значення метрик Precision, Recall та F1-score для кожного з досліджуваних сценаріїв поведінки контейнерів. Значення метрики Recall демонструють здатність методу виявляти події певного класу серед усіх подій цього класу, тоді як Precision характеризує точність прийнятих рішень. Отримані результати показують, що найкраща чутливість досягається для сценарію запуску інтерактивної оболонки, для якого значення Recall є найвищим. Для інших сценаріїв, зокрема нормальної поведінки та емуляції криптомайнера, значення метрик є низькими або нульовими. Це підтверджує обмеженість rule-based підходу при використанні невеликого набору агрегованих ознак у контейнеризованому середовищі.

Таблиця 2

Показники якості класифікації поведінкових сценаріїв

Клас	Precision	Recall	F1	Support
normal	0.000	0.000	0.000	4
shell	0.125	0.500	0.200	2
reverse_shell	0.000	0.000	0.000	4
miner	0.000	0.000	0.000	2

Аналіз результатів класифікації показав наявність значної кількості хибнопозитивних спрацювань, зокрема у випадках нормальної поведінки контейнера, яка в усіх часових вікнах була віднесена до аномальних класів. Загальна частка хибнопозитивних рішень перевищує половину проаналізованих вікон, що свідчить про надмірну чутливість rule-based підходу при використанні обмеженого набору поведінкових ознак. Основним джерелом хибнопозитивних спрацювань є домінування правил, пов'язаних із shell-активністю, що призводить до перекласифікації короткочасних або службових подій у підозрілі сценарії.

На рис. 1 наведено теплову карту матриці помилок класифікації поведінкових сценаріїв контейнерів. Інтенсивність кольору відповідає кількості часових вікон, віднесених до відповідного класу. Така візуалізація дозволяє наочно оцінити структуру помилок та системні перекласифікації. З рисунка видно, що алгоритм демонструє вибіркову чутливість до окремих сценаріїв, зокрема до подій, пов'язаних із запуском інтерактивної

оболонки. Водночас відсутність виявлення нормальної поведінки підкреслює проблему перекриття поведінкових профілів у контейнеризованих сервісах, де більшість активності є короткотривалою та однотипною.

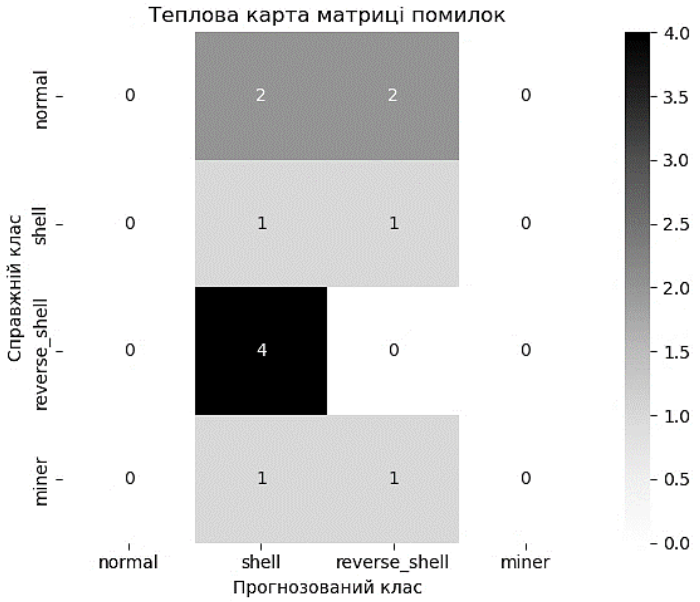

Рис. 1 - Теплова карта матриці помилок класифікації поведінкових сценаріїв контейнерів

Таким чином, отримані результати показали, що запропонований підхід дозволяє виявляти підозрілу активність у контейнерах, однак характеризується значною кількістю хибнопозитивних спрацювань. Зокрема, спостерігалася плутанина між сценаріями shell та reverse shell, а також складність коректного виявлення криптомайнера без урахування ресурсних метрик.

Висновки з даного дослідження і перспективи подальших досліджень у даному напрямі

Отримані в ході експерименту результати (таблиця 1, таблиця 2) вказують на суттєві обмеження детермінованого rule-based підходу при аналізі контейнеризувати робочих навантажень. Незважаючи на високу чутливість до сценарію shell, загальна точність системи виявилася незадовільною через кілька фундаментальних чинників:

1. Застосування фіксованих часових інтервалів тривалістю 10 секунд спричинило розрив контексту подій, що критично вплинуло на детекцію зворотної оболонки. Оскільки встановлення мережевого з'єднання та запуск процесів можуть відбуватися на межі сусідніх вікон, алгоритм не здатний пов'язати їх у єдиний ланцюжок. Це призвело до помилкової класифікації інцидентів як окремих подій та нульового показника Recall для даного сценарію.

2. Низька точність ідентифікації нормального трафіку свідчить про те, що штатне функціонування сервісу Nginx генерує активність, яка перекривається з ознаками атак. Системні виклики execve для службових скриптів або інтенсивний запис у міді-файли за своїми параметрами збігаються з пороговими значеннями для шкідливого програмного забезпечення. Це підтверджує, що проста агрегація подій без глибокого аналізу їхньої семантики є недостатньою для надійної дискримінації трафіку.

3. Обмеженість телеметрії системних викликів виявилася критичною при спробах детекції криптомемуарів, які переважно споживають ресурси процесора. Після етапу ініціалізації такі процеси здійснюють обчислення в просторі користувача без генерації аномальної кількості звернень до ядра. Оскільки метод базувався лише на інтенсивності подій Falco, він продемонстрував неспроможність виявляти загрози, що не створюють вираженого потоку системних викликів.

4. Жорстка ієрархія пріоритетів у rule-based логіці стала основною причиною високого рівня хибнопозитивних спрацювань. Будь-яка короткочасна активність, пов'язана із запуском оболонки для адміністративних цілей, автоматично призводила до класифікації всього часового вікна як атаки. Такий результат вказує на необхідність заміни лінійних правил на імовірнісні моделі, здатні гнучкіше враховувати контекст виконання операцій.

Разом із тим, отримані результати демонструють принципову можливість формування поведінкових профілів контейнерів на основі системних викликів та обґрунтовують доцільність подальших досліджень у цьому напрямку. Запропонований підхід дозволяє фіксувати підозрілу активність у контейнеризовано му середовищі та є придатним для подальшого розвитку. Отримані результати підтверджують, що поведінковий аналіз контейнерів має специфічні обмеження, пов'язані з короткочасністю атак, динамічністю життєвого циклу контейнерів та обмеженістю доступних ознак.

Подальші дослідження доцільно спрямувати на використання ковзних та адаптивних часових вікон, врахування часової залежності між подіями, інтеграцію ресурсних метрик та застосування методів машинного навчання.

References

1. Skorynovych B., Lakh Y. Assessing the potential of using artificial intelligence for intrusion detection in containerized environments // *Cybersecurity: Education, Science, Technique*. — 2025. — № 29. — DOI: 10.28925/2663-4023.2025.29.821.

2. Pitkar R. Enhancing Kubernetes security with AI: anomaly detection using logs, API calls, and network traffic // *International Journal of Software Engineering & Management*. — 2025. — Vol. 4. — P. 1–9. — DOI: 10.55041/isjem02746.

3. Hamza A. et al. Detecting volumetric attacks on IoT devices via behavioral compliance monitoring // *Proceedings of the ACM Conference*. — 2019. — P. 36–48. — DOI: 10.1145/3314148.3314352.

4. Kamaluddin S. Fine-grained behavioral analysis for malware detection in containerized environments // *American Journal of Computer Engineering*. — 2021. — Vol. 4. — P. 1–20. — DOI: 10.47672/ajce.2725.

5. Canella C. et al. Automating seccomp filter generation for Linux containers // *Proceedings of the ACM Conference*. — 2021. — P. 139–151. — DOI: 10.1145/3474123.3486762.

6. Gajbhiye B., Goel O. Managing vulnerabilities in containerized and Kubernetes environments // *Journal of Quantum Science and Technology*. — 2024. — P. 59–71. — DOI: 10.36676/jqst.v1.i2.16.