DOI 10.20535/2411-1031.2024.12.2.316257
УДК 004.075

ІВАН ГОРНІЙЧУК,
МИХАЙЛО ШЕЛЕЛЬО,
АРТЕМ МИКИТЮК,
ВОЛОДИМИР ОНІЩЕНКО

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОРКЕСТРАЦІЇ ВІРТУАЛЬНОГО СЕРЕДОВИЩА КІБЕРПОЛІГОНУ НАВЧАЛЬНОГО СИТУАЦІЙНОГО ЦЕНТРУ З КІБЕРБЕЗПЕКИ

Стрімкий розвиток інформаційних технологій та постійне зростання складності кіберзагроз створюють нові виклики у сфері кібербезпеки, що потребують сучасних та інноваційних підходів до підготовки фахівців. Ефективне навчання вимагає реалістичних і безпечних платформ, таких як інтерактивні симульовані середовища, які дозволяють моделювати різноманітні сценарії атак і захисту. Кіберполігони стали основним інструментом для розвитку навичок у сфері кібербезпеки, однак створення таких платформ є складним і ресурсомістким процесом. Оптимізація управління віртуальними ресурсами є критичною умовою для забезпечення гнучкості та масштабованості таких середовищ.

Метою роботи є підвищення ефективності управління віртуальними середовищами для тренувань у кібербезпеці шляхом вдосконалення процесу оркестрації віртуальних ресурсів кіберполігону. У роботі наведено визначення кіберполігону як інтерактивної симульованої платформи, яка може включати фізичне та віртуальне обладнання для створення навчальних середовищ, максимально наближених до реальних умов. Виокремлено основні типи таких платформ та основні категорії їх користувачів. Проаналізовано сучасні підходи до оркестрації віртуальних середовищ, такі як ручна конфігурація, написання сценаріїв, інфраструктура як код, контейнеризація та хмарна оркестрація. Запропоновано інформаційну технологію оркестрації віртуального середовища кіберполігону навчального ситуаційного центру з кібербезпеки, визначено функціональні вимоги та фізичну структуру модуля оркестрації.

У ході роботи розроблено програмну реалізацію запропонованої інформаційної технології у вигляді модуля оркестрації, що автоматизує процеси розгортання, конфігурації, управління та масштабування віртуальних ресурсів. Розробка базується на технологіях Python 3, Flask 3, Docker та JWT, що дозволяє забезпечити безпечну автентифікацію, ефективне керування контейнерами та інтеграцію через RESTful API. Проведене тестування підтвердило відповідність модуля критеріям масштабованості, гнучкості та безпеки.

Новизна роботи полягає у розробці інформаційної технології оркестрації віртуального середовища кіберполігону, яка враховує специфічні потреби організацій та мінімізує залежність від комерційних рішень, а також забезпечує інтеграцію сучасних технологій для підвищення ефективності навчальних платформ.

Ключові слова: кібербезпека, розподілені обчислювальні системи, оркестрація, кіберполігон, віртуальне середовище, контейнеризація, віртуалізація.

Постановка проблеми. Стрімкий розвиток інформаційних технологій та постійне ускладнення кіберзагроз ставлять перед організаціями нові виклики у сфері кібербезпеки. Ефективна підготовка фахівців з кібербезпеки потребує використання реалістичних та безпечних навчальних середовищ, які дозволяють моделювати різноманітні сценарії атак та відпрацьовувати стратегії захисту. Кіберполігони стали невід'ємним інструментом для тренувань та тестування навичок фахівців у цій галузі. Сьогодні існує багато готових рішень, розроблених різними компаніями, які використовують різноманітні технології та підходи до

оркестрації і управління віртуальними середовищами [1]–[5]. Однак кожна організація прагне мати власне середовище, адаптоване під її специфічні потреби та вимоги. Створення власної платформи є складним та ресурсомістким процесом, що потребує значних фінансових та технічних ресурсів. Зважаючи на високу вартість та обмеженість ресурсів, виникає необхідність у оптимальному розподілі та ефективному управлінні віртуальними ресурсами навчального середовища.

Ефективна оркестрація віртуального середовища є ключовим фактором для забезпечення масштабованості, гнучкості та надійності системи. У цьому контексті доцільним є використання технологій контейнеризації, таких як Docker, Podman, LXC та ін. Вони забезпечують ізоляцію додатків, швидке розгортання та економію ресурсів [4]–[9]. Таким чином, актуальною є розробка інформаційної технології оркестрації віртуального середовища кіберполігону та її програмна реалізація, який підвищить ефективність управління ресурсами та забезпечить гнучкість у налаштуванні навчальних сценаріїв для навчання.

Аналіз останніх досліджень і публікацій. Кіберполігони є ключовими платформами для моделювання реалістичних сценаріїв атак та захисту у сфері кібербезпеки, де оркестрація віртуального середовища забезпечує автоматизоване розгортання і управління ресурсами. Більшість комерційних рішень для оркестрації використовують загальні підходи, базовані на контейнеризації, віртуалізації та використанні хмарних платформ [3], [5]. Проте їх використання обмежує можливості організацій, оскільки залежність від постачальника знижує гнучкість і ускладнює адаптацію [10]–[13].

У науковій літературі бракує детальних методик, які б описували організацію процесів оркестрації віртуального середовища для навчальних платформ. Переважна більшість публікацій надає загальні рекомендації, зосереджуючись на теоретичних аспектах, і майже не висвітлює практичні підходи до реалізації комплексних рішень [5], [14]. Унаслідок цього організації змушені самостійно проводити дослідження та розробляти власні рішення, які забезпечують більшу гнучкість, ефективніше використання ресурсів та уникають залежності від сторонніх постачальників.

Метою роботи є підвищення ефективності управління віртуальними середовищами для тренувань у кібербезпеці шляхом вдосконалення процесу оркестрації віртуальних ресурсів кіберполігону.

Виклад основного матеріалу дослідження. Наразі в науковій літературі відсутнє загальноприйняте визначення кіберполігону, що ускладнює стандартизацію та уніфікацію підходів до його розробки й використання. Однак, проаналізувавши наявні джерела та визначивши фундаментальні принципи, можна запропонувати наступне визначення [1]–[3].

Кіберполігон – це інтерактивна симульована платформа, яка відтворює локальні мережі, системи, інструменти та додатки, підключені до симульованого Інтернет-середовища. Вона забезпечує безпечне та законне середовище для набуття практичних навичок у сфері кібербезпеки, реалізації та протидії мережевим атакам, виявлення вразливостей, розробки продуктів і тестування захисту. Така платформа може містити реальне апаратне та програмне забезпечення або їх комбінацію з віртуальними компонентами, імітуючи реальну інфраструктуру для тренування фахівців в умовах максимально наближених до реальних.

Такі системи мають широкий спектр потенційних користувачів, серед яких приватні й державні організації, фахівці з кібербезпеки, військові структури, оперативні центри безпеки (SOC) [15]–[18], викладачі, студенти, дослідники та організатори заходів. Це дозволяє застосовувати кіберполігони для різноманітних цілей, таких як тестування безпеки імітаційного середовища, оцінки та розвитку професійних компетенцій, навчання у сфері кібербезпеки, рекрутингу нових фахівців, а також проведення національних та міжнародних змагань.

Функціональні можливості таких середовищ визначають їх ефективність та застосовність у різних сценаріях. Однією з ключових можливостей є оркестрація, яка забезпечує автоматизовану конфігурацію, координацію та управління компонентів,

включаючи створення, модифікацію та видалення віртуальних машин, що підвищує зручність, масштабованість і ефективність платформи [2], [4]. Додаткові функції включають симуляцію Інтернет-сервісів, реалістичних атак та активності користувачів, дозволяючи відтворювати реальні мережеві умови та моделювати різні фази кібератак. Платформи забезпечують збір і аналіз даних для оцінки користувачів, відстеження їхнього прогресу та формування звітів, які визначають відповідність отриманих навичок професійним стандартам фахівців з кібербезпеки, а також надають інструменти для ефективного управління навчальними сценаріями [3].

Кіберполігони еволюціонували в чотири основні типи, кожен з яких має унікальні характеристики та можливості: симуляційні, накладні, емуляційні та гібридні полігони. Симуляційні полігони створюють синтетичне середовище, забезпечуючи швидку реконфігурацію й економічність, але мають обмеження щодо реалістичності. Накладні полігони функціонують на реальних мережах, забезпечуючи високу точність, проте вимагають значних ресурсів і несуть ризики для інфраструктури. Емуляційні полігони використовують виділену інфраструктуру для автентичного відтворення мережевих умов. Гібридні полігони поєднують елементи інших типів, створюючи гнучке середовище, адаптоване до специфічних завдань. Вибір відповідного типу залежить від потреб користувача або організації [5], [6].

Навчальні платформи різних типів базуються на ключових технічних компонентах, що забезпечують їх функціонування та ефективність. Основними технологічними складовими, які формують їхню основу, є система управління навчанням на полігоні (Range Learning Management System, RLMS), рівень оркестрації, базова інфраструктура, рівень віртуалізації та ізоляції, а також цільова інфраструктура.

Система RLMS є центральним компонентом більшості таких платформ. Вона поєднує стандартні функції системи управління навчанням (LMS) із додатковими можливостями, характерними для спеціалізованих середовищ. RLMS забезпечує взаємодію між двома основними сторонами: інструкторами та учнями [7]. Рівень оркестрації базується на вхідних даних від RLMS і інтегрує різні технологічні та сервісні компоненти системи, сприяючи динамічному масштабуванню за підтримки хмарної або фізичної інфраструктури [5].

Базова інфраструктура складається з мереж, серверів і сховищ та часто реалізується через програмно-визначену віртуальну інфраструктуру, що забезпечує гнучкість, масштабованість і економічну ефективність [7]. Рівень віртуалізації дозволяє зменшити залежність від фізичної інфраструктури завдяки гіпервізорам і програмно-визначеним рішенням, що створюють віртуальні середовища з ізоляцією ресурсів та спрощеним управлінням [5], [8]. Цільова інфраструктура відтворює реальні умови для тренувань, включаючи сервери, кінцеві точки, додатки та системи безпеки, моделюючи специфіку організаційної IT-інфраструктури [8].

Оркестрація віртуального середовища є ключовим фактором для створення реалістичних та складних сценаріїв у сфері кібербезпеки на кіберполігонах. Вона забезпечує автоматизацію процесів розгортання, конфігурації, управління та масштабування віртуалізованих ресурсів у мережевій інфраструктурі [1].

Виходячи з літературних джерел [2], [5], [9], оркестрацію віртуального середовища кіберполігонів можна визначити як автоматизований процес розгортання, конфігурації, управління та масштабування віртуалізованих ресурсів у мережевій інфраструктурі. Цей процес базується на використанні автоматизованих інструментів і сценаріїв для забезпечення ефективної та узгодженої роботи віртуальних машин, контейнерів та інших компонентів середовища, дозволяючи їх інтеграцію та взаємодію відповідно до попередньо визначених правил і вимог. Вона підтримує динамічний розподіл ресурсів, застосування політик і моніторинг для забезпечення оптимальної продуктивності та надійності віртуального середовища.

Розглянемо стандартні підходи до оркестрації віртуального середовища. Ручна конфігурація забезпечує точний контроль і адаптацію, але є трудомісткою, схильною до

помилок і неефективною в масштабі, що підходить лише для невеликих або специфічних середовищ. Написання сценаріїв (Scripting) автоматизує повторювані завдання, зменшуючи помилки, однак потребує регулярного оновлення, а зростання складності ускладнює управління. Інфраструктура як код (IaC) дозволяє декларативне визначення та управління інфраструктурою за допомогою таких інструментів, як Terraform, Ansible чи Puppet, забезпечуючи масштабованість і відтворюваність, але вимагає знань специфічних мов [2], [5]. Контейнеризація та мікросервіси використовують інструменти, як Kubernetes чи Docker Swarm, для автоматизації розгортання, масштабування і управління контейнерами, що підвищує ефективність і ізоляцію, однак вимагає додаткових налаштувань для збереження стану додатків і організації взаємодії між контейнерами. Хмарна оркестрація інтегрує сервіси хмарних платформ (AWS, Azure, Google Cloud) для гнучкого і масштабованого управління ресурсами, проте створює ризики залежності від постачальників послуг [2], [13].

Віртуалізація є основною технологією для реалізації кіберполігонів, забезпечуючи економічно ефективну імітацію складних систем. Традиційна віртуалізація використовує віртуальні машини (ВМ) під управлінням гіпервізорів типу 1 (bare-metal) або 2, надаючи гнучкість і контроль, але вимагаючи значних витрат на інфраструктуру і ліцензії [2], [14]. Контейнеризація ізолює додатки за допомогою спільного ядра операційної системи, підвищуючи ефективність використання ресурсів, але обмежена в моделюванні складних систем, таких як Windows з Active Directory. Хмарна віртуалізація, що базується на традиційній, пропонує вбудовані засоби оркестрації, масштабованість і динамічність конфігурацій. Публічні, приватні та гібридні рішення забезпечують різний баланс між контролем, масштабованістю та безпекою відповідно до потреб організацій [5], [14].

Впровадження кіберполігонів часто обмежено фінансовими ресурсами, що призводить до необхідності створювати власні рішення, адаптовані до специфічних потреб і завдань. Одним з них є розробка модуля оркестрації, який автоматизує процеси розгортання, конфігурації, управління та масштабування віртуалізованих ресурсів кіберполігону.

На основі викладеного матеріалу та проведеного аналізу запропоновано інформаційну технологію оркестрації віртуального середовища кіберполігону навчального ситуаційного центру з кібербезпеки. Її суть полягає у використанні підходу контейнеризації, в керованому середовищі.

Для управління ресурсами пропонується підхід, що базується на визначенні та управлінні доступними ресурсами, перевірці їх наявності, та прийнятті рішення про виконання або відмову у виконанні запитів на запуск завдань.

Нехай R множина доступних ресурсів:

$$R = \{r_1, r_2, ..., r_n\}, \quad (1)$$

де r_i – доступний ресурс, що характеризується вільною обчислювальною потужністю C_i, вільним обсягом пам'яті M_i та іншими параметрами H_i (мережеві інтерфейси, порти, тощо).

Запит на виконання завдання описується множиною:

$$Z = \{z_1, z_2, ..., z_m\}, \quad (2)$$

де z_j – запит ресурсів, що характеризується потребою у ресурсах, визначених через обчислювальну потужність $\hat{C}_j$, обсяг пам'яті $\hat{M}_j$, та інші необхідні параметри $\hat{H}_j$

Необхідно оптимально розподілити ресурси R між запитами Z для максимально ефективного їх використання, мінімізації витрат та забезпечення виконання завдань. При цьому існує обмеження: для кожного ресурсу r_i повинна виконуватись умова:

$$\begin{cases} C_i \geq \sum_{j=1}^{k} x_{ij}\hat{C}_j, \\ M_i \geq \sum_{j=1}^{k} x_{ij}\hat{M}_j,, \\ H_i \geq \sum_{j=1}^{k} x_{ij}\hat{H}_j \end{cases} \quad (3)$$

де x_{ij} – бінарна змінна, що позначає чи було виділено ресурси ($x_{ij}=1$, якщо ресурс r_i виділено для запиту z_j, в іншому випадку – $x_{ij}=0$).

Для обчислення значення x_{ij}, можливості виділення ресурсів необхідних для виконання запиту z_j, використовується функція $\phi(z_j)$:

$$x_{ij} = \phi(z_j),$$

$$\phi(z_j) = \begin{cases} 1, \ \textit{якщо } \exists r_i \in R : (C_i \geq \hat{C}_j, M_i \geq \hat{M}_j, H_i \geq \hat{H}_j), \\ 0, \ \textit{в іншому випадку}. \end{cases}, \quad (4)$$

Функція виділення ресурсів для виконання певного запиту виглядає наступним чином:

$$\theta(z_j) = \sum_{i=1}^{n} x_{ij}(\hat{C}_j + \hat{M}_j + \hat{H}_j), \quad (5)$$

Для оновлення доступних ресурсів використовуються такі функції:

$$C_i = C_i - \sum_{j=1}^{k} x_{ij}\hat{C}_j,$$

$$M_i = M_i - \sum_{j=1}^{k} x_{ij}\hat{M}_j,, \quad (6)$$

$$H_i = H_i - \sum_{j=1}^{k} x_{ij}\hat{H}_j.$$

Для вирішення проблеми ресурсної верифікація та конфігурації контейнерів сценаріїв на кіберполігоні навчального ситуаційного центру з кібербезпеки запропоновано використовувати файлу *image_info.json*. В файлі будуть визначатися основні параметри для запуску сценаріїв. Він виконує роль інструкції для модуля оркестрації щодо виділення ресурсів та налаштування контейнерів. Такий підхід дозволяє автоматизувати процеси та уникнути людських помилок при конфігуруванні середовищ. Далі наведено лістинг прикладу файлу *image_info.json*:

```
{
  “name”: “scenario_example”,
  “multiple_replica”: false,
  “container_resources”: {
    “cpu”: “1.0”,
    “memory”: “512m”
  },
  “ports”: [22, 40, 80],
  “port_definition_type”: “static”,
  “startup_type”: 1,
  “notes”: “Example setup for container.”
}
```

Цей файл містить ключові параметри для запуску контейнерів на кіберполігоні у форматі “ключ-значення”. Основними параметрами є:

– *name*: задає назва сценарію, необхідне для однозначної ідентифікації його конфігураційних файлів в загальній базі;

– *multiple_replica*: вказує необхідність запускати різні контейнери для різних користувачів, або ж можливість використання одного і того ж середовища;

– *container_resources*: описує обсяг ресурсів необхідний для запуску контейнера або їх сукупності, включає процесорні потужності та об’єм оперативної пам'яті;

– *ports*: задає список портів, які використовуються контейнером чи їх сукупністю та мусять бути перенаправлені через мережевий інтерфейс хостової машини;

– *port_definition_type*: визначає тип призначення портів – статичний для перенаправлення на мережевий інтерфейс хосту саме тих портів, що визначені цим конфігураційним файлом; динамічний – перенаправлення на будь-які доступні порти;

– *startup_type*: вказує метод запуску контейнера (*docker run*, *docker-compose* або складний сценарій через *Dockerfile.yml*);

– *notes*: задає додаткову інформацію необхідну для управління віртуальним середовищем сценарію.

Сформовано функціональні вимоги до модуля оркестрації, які охоплюють широкий спектр задач, пов’язаних із життєвим циклом контейнерів та інших віртуальних компонентів. Основні функції модуля повинні включати:

– *Запуск контейнера*: створення та запуск нового контейнера з виділенням необхідних обчислювальних ресурсів (процесора і пам'яті), перевіркою їх доступності.

– *Перезапуск контейнера*: повторний запуск існуючого контейнера для оновлення його стану або застосування нових конфігурацій.

– *Зупинка контейнера*: завершення роботи контейнера та звільнення ресурсів для оптимізації їх використання.

– *Отримання логів*: реєстрація подій у журналах системи та надання доступу до цих даних для моніторингу і діагностики роботи модулів та запущених контейнерів.

– *Перегляд інформації про ресурси*: отримання даних про поточне використання ресурсів контейнерами, що дозволяє аналізувати ефективність їх роботи.

Запропоновано фізичну структуру модуля оркестрації (рис. 1). Ця структура включає ключові модулі, які забезпечують функціонування та взаємодію віртуального середовища з контейнерними інфраструктурами та ресурсами.

Основні компоненти включають:

– Модуль REST API для приймання клієнтських запитів через HTTP;

– Модуль автентифікації для перевірки прав доступу та генерації токенів;

– Модуль логування для моніторингу подій і запитів;

– Модуль керування контейнерами для запуску, перезапуску та налаштування параметрів контейнерів через Docker API;

– Модуль керування портами для налаштування мережевих портів і маршрутизації;

– Модуль сховища образів для управління контейнерними образами;

– Модуль керування ресурсами для моніторингу та розподілу процесорного часу, пам'яті й дискового простору, забезпечуючи оптимальне використання ресурсів.

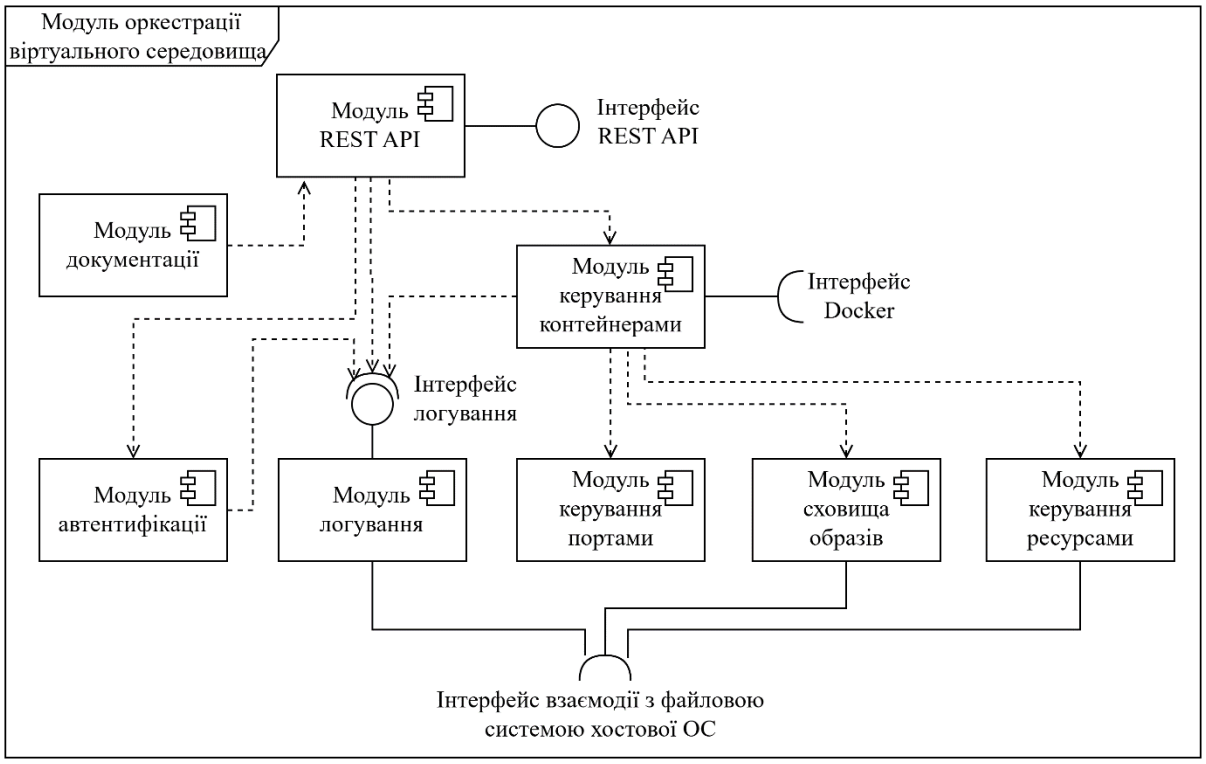

Рисунок 1 – Діаграма компонентів програмного модуля оркестрації віртуального середовища кіберполігону

Програмний модуль оркестрації повинен реалізувати ключові сценарії, які охоплюють напрямки його роботи. Їх можна класифікувати на три основні категорії:

– *Механізми доступу*: процеси авторизації користувача та оновлення токенів доступу, що забезпечують безпечний доступ до системи.

– *Управління контейнерами*: операції запуску, перезапуску та зупинки контейнерів за API-запитами, включаючи валідацію системних ресурсів та обробку запитів.

– *Процедури надання даних*: отримання логів та інформації про ресурси для моніторингу та управління системою.

Прийнято рішення використовувати архітектурний стиль REST (Representational State Transfer) , який є широко використовуваним підходом у розробці веб-сервісів. Цей метод забезпечує високу гнучкість, масштабованість і простоту інтеграції. Вибір RESTful API зумовлений його здатністю ефективно масштабуватись завдяки безстановій природі, що дозволяє працювати з великою кількістю клієнтів одночасно без необхідності збереження стану на сервері. Використання стандартних HTTP методів і форматів даних, таких як JSON, забезпечує легкість інтеграції з різними клієнтами та сервісами. Крім того, така архітектура сприяє легкості розширення функціональності та інтеграції нових ресурсів, не вимагаючи значних змін у структурі системи. Простота розробки і тестування завдяки уніфікованому інтерфейсу ресурсів знижує витрати на розробку та підтримку [19].

У ході дослідження розроблено програмну реалізацію запропонованої інформаційної технології з використанням інтеграції кількох ключових технологій та інструментів. Фреймворк Flask 3 було використано для розробки RESTful API, забезпечуючи простий та гнучкий доступ до функцій модуля. Python 3 слугує основною мовою для реалізації бізнес-логіки, тоді як Docker забезпечує контейнеризацію, дозволяючи ізольоване й незалежне виконання компонентів [20]. Для аутентифікації обрано JWT, що забезпечує захищений обмін даними. Така архітектура підтримує масштабованість, полегшує розширення системи та забезпечує надійну взаємодію компонентів, дозволяючи в майбутньому безперешкодне додавання нових функцій без значних змін архітектури.

Розроблений програмний модуль для оркестрації віртуального середовища кіберполігону забезпечує комплексне управління контейнерами, включаючи їх створення, перезапуск, видалення, доступ до логів та моніторинг доступних ресурсів. Ці функціональні можливості дозволяють ефективно керувати обчислювальними ресурсами, забезпечуючи гнучкість і контроль над навчальним середовищем. Виконання вищенаведених можливостей доступне при наявності дійсного токену доступу (access token), що виконує роль цифрового ключа для авторизації користувачів. Доступ до цих можливостей здійснюється через RESTful API за допомогою HTTP-запитів з використанням методів GET та POST.

Модуль оркестрації виконує роль центрального координатора системи управління віртуальним середовищем, інтегруючи різні компоненти та забезпечуючи повний життєвий цикл контейнерів від їх створення до видалення. На рис. 2 представлена схема функціонування програмного модуля оркестрації.

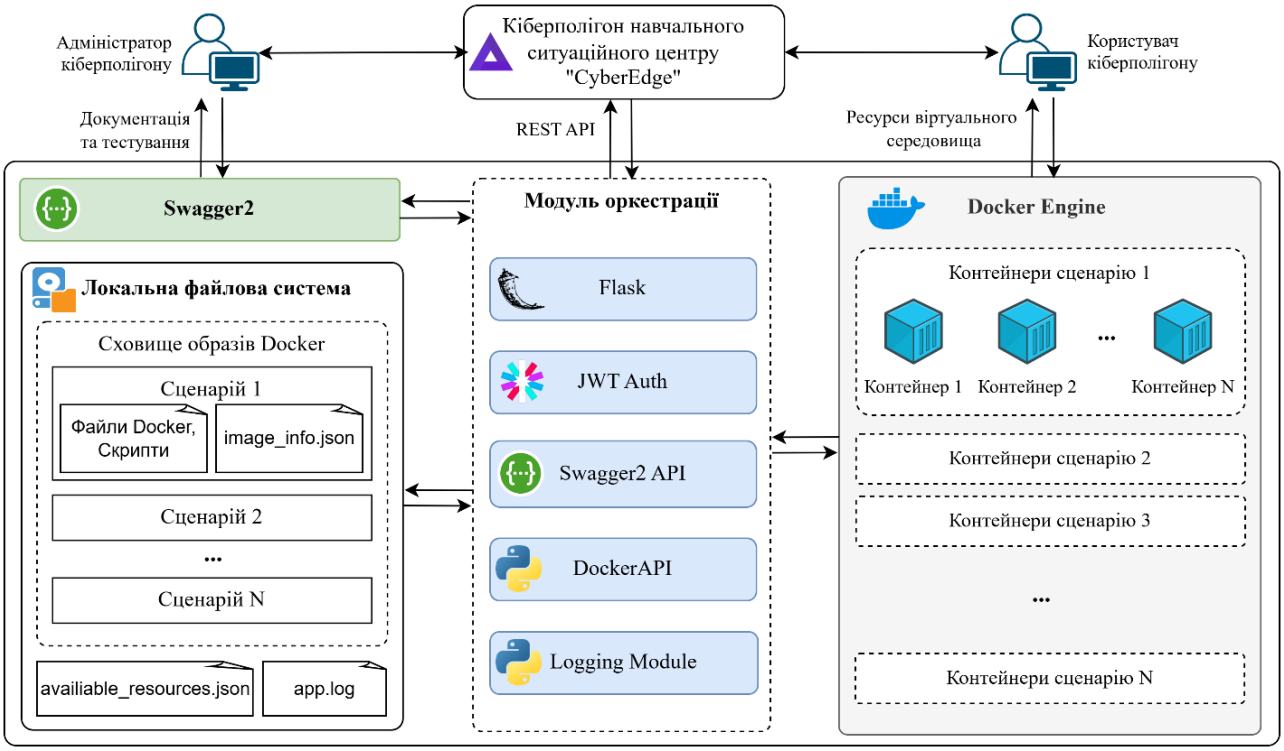

Рисунок 2 – Схема функціонування програмного модуля оркестрації

Кіберполігон навчального ситуаційного центру “CyberEdge”, попередньо автентифікувавшись та отримавши JWT-токен, звертається до модуля оркестрації через REST API із запитом на розгортання віртуального середовища сценарію. Модуль оркестрації здійснює перегляд сховища образів Docker, що знаходиться на локальній файловій системі хоста, на наявність необхідного сценарію, та завантажує вимоги до необхідного віртуального середовища з файлу *image_info.json*. Після цього здійснюється верифікація необхідних ресурсів (4), на основі інформації про їх доступність із файлу *available_resources.json*, їх виділення для розгортання середовища сценарію (5) та перерахунок залишку ресурсів із подальшою модифікацією відповідного файлу (6). Після чого модуль надає кіберполігону відповідь про успішність чи невдачу ініційованої операції, а той в свою чергу надає користувачам доступ до запущеної інфрастуктури сценарію.

За необхідності користувачі можуть ініціювати перезапуск або зупинку роботи віртуального середовища. Після чого кіберполігон звертається до модуля із запитом, відбувається відповідна операція, щодо управління контейнерним оточенням, та оновлення файлу доступних ресурсів.

Крім цього адміністраторам кіберполігону надається можливість відстеження запущених контейнерів, отримання інформації про доступні ресурси та перегляду логів роботи модуля. Додатково впроваджено систему документації та тестування REST API.

Демонстрація роботи розробленого програмного модуля. Для демонстрації роботи програмного модуля оркестрації віртуального середовища кіберполігону використано інтерфейс документації Swagger2, який забезпечує інтерактивний доступ до всіх кінцевих точок API (див. рис. 3).

Першим кроком є отримання JWT-токенів доступу, необхідних для виконання захищених API-запитів. Для цього надсилається запит на автентифікацію до інтерфейсу */auth/login* з відповідними даними користувача. Після успішної автентифікації сервер повертає відповідь, що містить *access token* для авторизації в системі та токен оновлення (*refresh token*), який дозволяє отримати новий токен доступу після закінчення терміну дії попереднього. Клієнт зберігає ці токени та використовує access-токен для авторизації подальших запитів. Відповідь сервера наведена на рисунку 4.

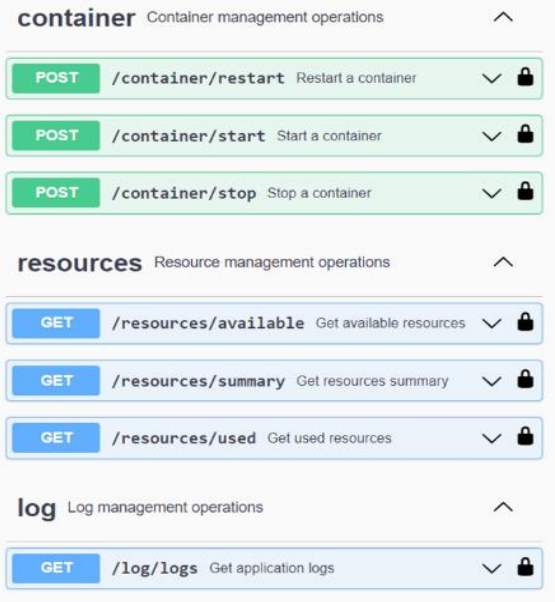

Рисунок 3 – Вигляд інтерфейсу Swagger2

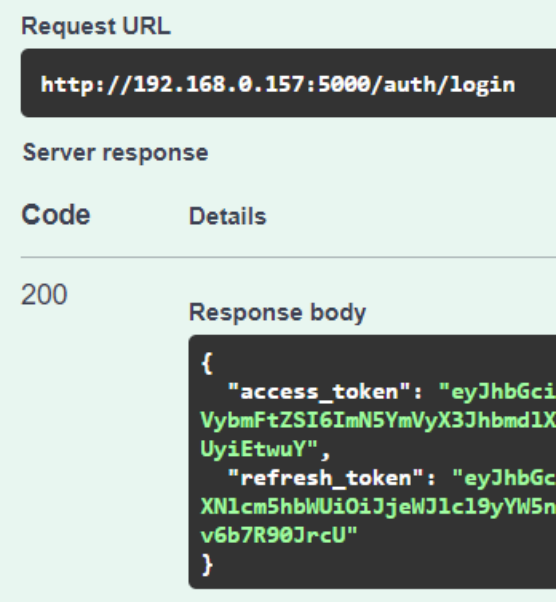

Рисунок 4 – Відповідь сервера після успішної автентифікації

З використанням отриманого токена можна виконувати захищені API-запити. Наприклад, для створення та запуску контейнера певного сценарію надсилається запит до інтерфейсу */container/start*. У тілі запиту передається назва сценарію для запуску (див. рисунок 5). У разі успішного виконання запиту сервер повертає відповідь із кодом 200 та JSON-даними, що містять інформацію для підключення до запущеного контейнера (IP-адресу та порт). Приклад відповіді показано на рисунку 6.

Підключившись до вказаної IP-адреси та порту в локальній мережі, користувач отримує доступ до запущеного контейнера і може з ним взаємодіяти.

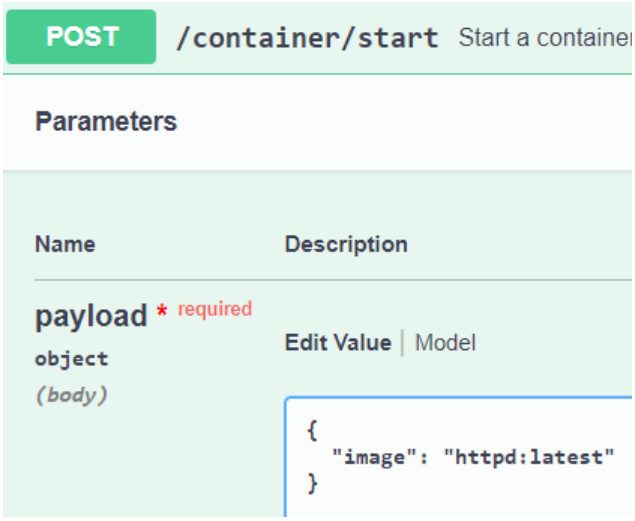

Рисунок 5 – Запит на запуск контейнера через інтерфейс */container/start*

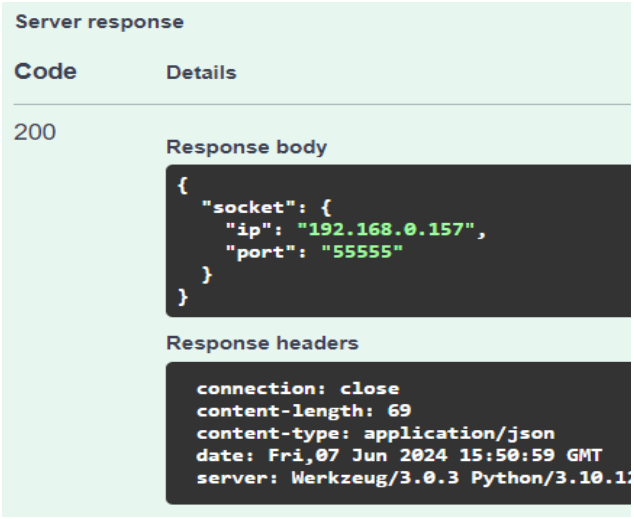

Рисунок 6 – Відповідь сервера після запуску контейнера

Висновки. В результаті проведеної роботи проаналізовано останні дослідження і публікації щодо підходів до оркестрації віртуальних середовищ кіберполігонів. Досліджено стандартні підходи до оркестрації, такі як ручна конфігурація, написання сценаріїв, інфраструктура як код, контейнеризація та мікросервіси, а також хмарна оркестрація. Запропоновано інформаційну технологію оркестрації віртуального середовища кіберполігону навчального ситуаційного центру з кібербезпеки, сформовано функціональні та структурні вимоги до програмного модуля оркестрації.

Розроблено програмну реалізацію запропонованої інформаційної технології з дотриманням висунутих вимог для автоматизації процесів розгортання, конфігурації, управління та масштабування віртуалізованих ресурсів. Модуль реалізовано з використанням технологій Docker, Python 3, Flask 3 та JWT, які забезпечують ефективне управління контейнерами, безпечну автентифікацію та взаємодію через RESTful API. Тестування підтвердило успішну інтеграцію з Docker, коректну роботу API та відповідність вимогам щодо масштабованості та безпеки.

Реалізований програмний модуль оркестрації віртуального середовища було інтегровано в кіберполігон “CyberEdge” навчального ситуаційного центру з питань кібербезпеки ІСЗЗІ КПІ ім. Ігоря Сікорського.

Отримано свідоцтво про реєстрацію авторських прав на розроблений в ході дослідження програмний застосунок [21].

Перспективи подальших досліджень полягають у розширенні функціональності програмного модуля оркестрації, покращені механізмів моніторингу та автоматизації, а також адаптації рішення для різних типів кіберполігонів з урахуванням специфічних потреб організацій у сфері кібербезпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] “Єгор Аушев запустив кіберполігон Unit Range для спеціалістів з безпеки”, *AIN.ua*. [Електронний ресурс]. Доступно: https://ain.ua/2023/06/14/v-ukrayini-zapustyly-kiberpoligon-unit-range/. Дата звернення: Серп. 4, 2024.

[2] “Understanding Cyber Ranges: From Hype to Reality”, *ECSO*. [Online]. Available: https://ecs-org.eu/ecso-uploads/2023/05/2020_SWG-5.1_paper_UnderstandingCyberRanges_final_v1.0-update.pdf. Accessed on: Aug. 4, 2024.

[3] "Cybersecurity Exercises for Training and Capability Development", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/. Accessed on: Sep. 8, 2024.
[4] "Cyber Range – what it is, what it is not and what it will be!", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/cyber-range-what-it-is-what-it-is-not-and-what-it-will-be/. Accessed on: Aug. 3, 2024.
[5] "The Cyber Range: a guide", *NIST*. [Online]. Available: https://www.nist.gov/system/files/documents/2023/09/29/The%20Cyber%20Range_A%20Guide.pdf. Accessed on: Sep. 11, 2024.
[6] G. Messina, "Types of cyber ranges compared: Simulations, overlays, emulations and hybrids", *Infosec*. [Online]. Available: https://www.infosecinstitute.com/resources/cyber-range/types-of-cyber-ranges-compared-simulations-overlays-emulations-and-hybrids/. Accessed on: Sep. 11, 2024.
[7] "Learn to select the right cyber range for you!", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/learn-to-select-the-right-cyber-range-for-you/. Accessed on: Sep. 11, 2024.
[8] M. Finio, and A. Downie, "What is a cyber range?", IBM. [Online]. Available: https://www.ibm.com/think/topics/cyber-range. Accessed on: Sep. 14, 2024.
[9] "Cyber Ranges Glossary", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/cyber-ranges-glossary/. Accessed on: Sep. 11, 2024.
[10] "Automation by PowerShell, Bash Script, and Python", *easillyy*. [Online]. Available: http://easillyy.com/automation-by-powershell-bash-script-and-python. Accessed on: Sep. 20, 2024.
[11] "Understanding Ansible, Terraform, Puppet, Chef, and Salt", *Red Hat*. [Online]. Available: https://www.redhat.com/en/topics/automation/understanding-ansible-vs-terraform-puppet-chef-and-salt. Accessed on: Sep. 8, 2024.
[12] V.R. Adkoli, "Cloud Orchestration with Infrastructure as Code (IaC): The Benefits", *Open Source For You*. [Online]. Available: https://www.opensourceforu.com/2023/10/cloud-orchestration-with-infrastructure-as-code-iac-the-benefits. Accessed on: Sep. 11, 2024.
[13] "Hybrid Cloud Orchestration: Making applications work together", *ITpedia*. [Online]. Available: https://en.itpedia.nl/2023/11/11/hybride-cloud-orkestratie-aws-azure-en-gcp-applicaties-samen-laten-werken. Accessed on: Sep. 11, 2024.
[14] "What Is Virtualization?", IBM. [Online]. Available: https://www.ibm.com/topics/virtualization. Accessed on: Oct. 1, 2024.
[15] I. Subach, A. Mykytiuk, and V. Kubrak, "Methodology of rational choice of security incident management system for building operational security center", in *Proc. 19th International Scientific and Practical Conference "Information Technologies and Security"* (*ITS 2019*), *CEUR Workshop Proceedings*, vol. 2577, pp. 11-20, 2019. doi: https://doi.org/10.5281/zenodo.7027782.
[16] I. Subach, D. Mogylevych, A. Mykytiuk, V. Kubrak, and O. Korotayev, "Design methodology of cyber security operational center", in *Proc. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-II-2021), CEUR Workshop Proceedings*, vol. 3187, pp. 79-88, 2021. doi: https://doi.org/10.5281/zenodo.7123828.
[17] "How to set up CSIRT and SOC", *European Union Agency for Cybersecurity*, December 10, 2020. [Online]. Available: https://www.enisa.europa.eu/publications/how-to-set-up-csirt-and-soc. Accessed on: Oct. 3, 2024.
[18] A. Zhylin, M. Khudyncev, and M. Litvinov, "Functional model of cybersecurity situation center", *Information Technology and Security*, vol. 6 (2), pp. 51-67, 2018. doi: https://doi.org/10.20535/2411-1031.2018.6.2.153490.
[19] S.J. Bigelow, and A.S. Gillis, "What is REST API (RESTful API)?", *TechTarget*. [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/RESTful-API Accessed on: Oct. 3, 2024.

[20] J. Haro, *Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More*. New York, USA: Manning Publ, 2023.

[21] М. Шелельо, І. Горнійчук, В. Оніщенко, А. Микитюк, та О. Шаповал, "Комп'ютерна програма "Програмний модуль оркестрації віртуального середовища кіберполігону навчального ситуаційного центру з кібербезпеки "CyberVE Orchestrator", *свід. про реєстр. авт. правана твір. № 130618*, Жовт. 14, 2024.

Стаття надійшла до редакції 03.11.2024.

REFERENCE

[1] "Egor Aushev launches Unit Range cyber range for security professionals", *AIN.ua*. [Online]. Available: https://ain.ua/2023/06/14/v-ukrayini-zapustyly-kiberpoligon-unit-range/. Accessed on: Aug. 4, 2024.

[2] "Understanding Cyber Ranges: From Hype to Reality", *ECSO*. [Online]. Available: https://ecs-org.eu/ecso-uploads/2023/05/2020_SWG-5.1_paper_UnderstandingCyberRanges_final_v1.0-update.pdf. Accessed on: Aug. 4, 2024.

[3] "Cybersecurity Exercises for Training and Capability Development", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/. Accessed on: Sep. 8, 2024.

[4] "Cyber Range – what it is, what it is not and what it will be!", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/cyber-range-what-it-is-what-it-is-not-and-what-it-will-be/. Accessed on: Aug. 3, 2024.

[5] "The Cyber Range: a guide", *NIST*. [Online]. Available: https://www.nist.gov/system/files/documents/2023/09/29/The%20Cyber%20Range_A%20Guide.pdf. Accessed on: Sep. 11, 2024.

[6] G. Messina, "Types of cyber ranges compared: Simulations, overlays, emulations and hybrids", *Infosec*. [Online]. Available: https://www.infosecinstitute.com/resources/cyber-range/types-of-cyber-ranges-compared-simulations-overlays-emulations-and-hybrids/. Accessed on: Sep. 11, 2024.

[7] "Learn to select the right cyber range for you!", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/learn-to-select-the-right-cyber-range-for-you/. Accessed on: Sep. 11, 2024.

[8] M. Finio, and A. Downie, "What is a cyber range?", IBM. [Online]. Available: https://www.ibm.com/think/topics/cyber-range. Accessed on: Sep. 14, 2024.

[9] "Cyber Ranges Glossary", *CYBER RANGES*. [Online]. Available: https://www.cyberranges.com/cyber-ranges-glossary/. Accessed on: Sep. 11, 2024.

[10] "Automation by PowerShell, Bash Script, and Python", *easillyy*. [Online]. Available: http://easillyy.com/automation-by-powershell-bash-script-and-python. Accessed on: Sep. 20, 2024.

[11] "Understanding Ansible, Terraform, Puppet, Chef, and Salt", *Red Hat*. [Online]. Available: https://www.redhat.com/en/topics/automation/understanding-ansible-vs-terraform-puppet-chef-and-salt. Accessed on: Sep. 8, 2024.

[12] V.R. Adkoli, "Cloud Orchestration with Infrastructure as Code (IaC): The Benefits", *Open Source For You*. [Online]. Available: https://www.opensourceforu.com/2023/10/cloud-orchestration-with-infrastructure-as-code-iac-the-benefits. Accessed on: Sep. 11, 2024.

[13] "Hybrid Cloud Orchestration: Making applications work together", *ITpedia*. [Online]. Available: https://en.itpedia.nl/2023/11/11/hybride-cloud-orkestratie-aws-azure-en-gcp-applicaties-samen-laten-werken. Accessed on: Sep. 11, 2024.

[14] "What Is Virtualization?", IBM. [Online]. Available: https://www.ibm.com/topics/virtualization. Accessed on: Oct. 1, 2024.

[15] I. Subach, A. Mykytiuk, and V. Kubrak, "Methodology of rational choice of security incident management system for building operational security center", in *Proc. 19th International*

Scientific and Practical Conference "Information Technologies and Security" (*ITS 2019*), *CEUR Workshop Proceedings*, vol. 2577, pp. 11-20, 2019. doi: https://doi.org/10.5281/zenodo.7027782.

[16] I. Subach, D. Mogylevych, A. Mykytiuk, V. Kubrak, and O. Korotayev, "Design methodology of cyber security operational center", in *Proc. Cybersecurity Providing in Information and Telecommunication Systems (CPITS-II-2021), CEUR Workshop Proceedings*, vol. 3187, pp. 79-88, 2021. doi: https://doi.org/10.5281/zenodo.7123828.

[17] "How to set up CSIRT and SOC", *European Union Agency for Cybersecurity*, December 10, 2020. [Online]. Available: https://www.enisa.europa.eu/publications/how-to-set-up-csirt-and-soc. Accessed on: Oct. 3, 2024.

[18] A. Zhylin, M. Khudyncev, and M. Litvinov, "Functional model of cybersecurity situation center", *Information Technology and Security,* vol. 6 (2), pp. 51-67, 2018. doi: https://doi.org/10.20535/2411-1031.2018.6.2.153490.

[19] S.J. Bigelow, and A.S. Gillis, "What is REST API (RESTful API)?", *TechTarget*. [Online]. Available: https://www.techtarget.com/searchapparchitecture/definition/RESTful-API Accessed on: Oct. 3, 2024.

[20] J. Haro, *Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More*. New York, USA: Manning Publ, 2023.

[21] M. Shelelo, I. Horniichuk, V. Onishchenko, A. Mykytiuk, and O. Shapoval, "Computer Program "Software Module for Orchestration of the Virtual Environment of the Cyber Range Training Situation Center with Cybersecurity "CyberVE Orchestrator", *cert. about the register author rights to the work no. 130618*, Oct. 14, 2024.

IVAN HORNIICHUK,
MYKHAILO SHELELO,
ARTEM MYKYTIUK,
VOLODYMYR ONISHCHENKO

INFORMATION TECHNOLOGY FOR ORCHESTRATION OF THE CYBERSECURITY TRAINING SITUATION CENTER CYBER RANGE VIRTUAL ENVIRONMENT

The rapid development of information technology and the ever-increasing complexity of cyber threats create new challenges in the field of cybersecurity that require modern and innovative approaches to training. Effective training requires realistic and secure platforms, such as interactive simulated environments that allow for the modelling of various attack and defence scenarios. Cyber ranges have become a key tool for developing cybersecurity skills, but creating such platforms is a complex and resource-intensive process. Optimising the management of virtual resources is a critical condition for ensuring the flexibility and scalability of such environments.

The aim of the paper is to improve the efficiency of managing virtual environments for cybersecurity training by improving the process of orchestrating virtual resources of a cyber training ground. The paper defines a cyber training ground as an interactive simulated platform that may include physical and virtual equipment to create training environments that are as close as possible to real-world conditions. The main types of such platforms and the main categories of their users are allocated. The modern approaches to the orchestration of virtual environments, such as manual configuration, scripting, infrastructure as code, containerisation and cloud orchestration, are analysed. An information technology for orchestrating the virtual environment of the cyber range training

situation center for cybersecurity has been proposed, with functional requirements and the physical structure of the orchestration module defined.

In the course of the work, a software implementation of the proposed information technology was developed in the form of an orchestration module, which automates the processes of deployment, configuration, management, and scaling of virtual resources. The development is based on Python 3, Flask 3, Docker, and JWT technologies, which allows for secure authentication, efficient container management, and integration via RESTful API. The testing confirmed that the module meets the criteria of scalability, flexibility, and security.

The novelty of the work lies in the development of information technology for orchestrating the virtual environment of a cyber training ground, which takes into account the specific needs of organisations and minimises dependence on commercial solutions, as well as integrates modern technologies to improve the efficiency of training platforms.

Keywords: cybersecurity, distributed computing systems, orchestration, cyber range, virtual environment, containerisation, virtualisation.

Горнійчук Іван Вікторович, доктор філософії, старший викладач кафедри кібербезпеки і застосування інформаційних систем і технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна. ORCID 0000-0001-6754-4764, horniychuk.ivan@gmail.com.

Шелельо Михайло Іванович, курсант, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна. ORCID 0009-0007-8066-5297, shelelomuxajlo@gmail.com.

Микитюк Артем В'ячеславович, доктор філософії, заступник завідувача кафедри кібербезпеки і застосування інформаційних систем і технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна. ORCID 0000-0002-8307-9978, mukuta8888@gmail.com.

Оніщенко Володимир Олександрович, старший викладач кафедри кібербезпеки і застосування інформаційних систем і технологій, Інститут спеціального зв'язку та захисту інформації Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна. ORCID 0009-0000-1355-9178, v.o.onishchenko@ukr.net.

Ivan Horniichuk, PhD in engineering, senior lecturer of the cybersecurity and application of information systems and technologies academic department, Institute of special communication and information protection of National technical university of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Mykhailo Shelelo, cadet, Institute of special communication and information protection of National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute", Kyiv, Ukraine.

Artem Mykytiuk, PhD in engineering, deputy of the head at the cybersecurity and application of information systems and technologies academic department, Institute of special communications and information protection of National technical university of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Volodymyr Onishchenko, senior lecturer of the cybersecurity and application of information systems and technologies academic department, Institute of special communication and information protection of National technical university of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.