УДК 004.056

В.О. Козачок
В.В. Лукічов

МЕТОД ПІДВИЩЕННЯ ЗАХИЩЕНОСТІ DOCKER-КОНТЕЙНЕРІВ

Вінницький національний технічний університет

Анотація

Здійснено дослідження існуючих методів захисту Docker контейнерів та розроблено власний метод підвищення захищеності Docker-контейнерів.

Ключові слова: *Docker, контейнери, аналіз конфігурації Docker контейнерів, підвищення захищеності Docker контейнерів, автоматизація виправлення помилок.*

Abstract

Investigation of existing methods of protecting Docker containers and develops its own method of increasing the security of Docker containers.

Keywords*: Docker, containers, analysis of Docker container configuration, increasing the security of Docker containers, automating error correction.*

Вступ

В епоху швидкого технологічного розвитку повсюдне поширення технологій контейнеризації зробило революцію в розробці, розгортанні та масштабуванні додатків. Docker, провідний лідер в управлінні контейнерами, надає потужну основу для інкапсуляції додатків та їх залежностей. Однак, з повсюдним впровадженням контейнерних рішень, проблеми з безпекою стають все більш актуальними. Забезпечення відмовостійкої та безпечної конфігурації контейнерів Docker стає першочерговим завданням, що вимагає надійних рішень для автоматизації.

Починаючи з вивчення основних функцій безпеки Docker і Ansible, це дослідження формулює їх індивідуальний внесок у захист контейнерних середовищ. Згодом фокус зміщується на точки інтеграції, де Ansible розширює можливості безпеки Docker, автоматизуючи конфігурацію засобів контролю безпеки, управління доступом і перевірку відповідності.

Результати дослідження

Docker-контейнери – це легкі, портативні та самодостатні одиниці, які інкапсулюють програму та її залежності, включаючи бібліотеки, середовище виконання та системні інструменти. Ці контейнери забезпечують узгоджене та відтворюване середовище в різних обчислювальних середовищах, що дозволяє розробникам безперешкодно створювати, постачати та запускати додатки на різних платформах. До ключових особливостей контейнерів Docker належать ізоляція, переносимість, ефективність, керування версіями, реєстр образів, оркестрування [1].

В сучасному світі доводиться впроваджувати множину засобів, для забезпечення інфраструктури. Використання контейнерів хоч і зменшує кількість вразливостей з одного боку, проте дає зловмисникам нові вектори атак, які можна експлуатувати. Починаючи від рівня серверного обладнання та операційної системи, закінчуючи двигуном контейнерів, засобом управління контейнерами, середою виконання і власне самим контейнером рис. 1.

Контейнери використовують віртуалізацію на рівні операційної системи, щоб ізолювати програми від базової інфраструктури. Ця ізоляція гарантує, що програми працюють стабільно, незалежно від хост-середовища [2].

Docker контейнери пакують додатки та їхні залежності в єдиний стандартизований блок, це забезпечує їх переносимість. Тобто контейнер може працювати на будь-якій системі, що підтримує Docker, що полегшує переміщення додатків між середовищами розробки, тестування та виробництва.

Контейнери використовують ядро операційної системи хоста, зменшуючи накладні витрати, пов'язані з традиційною віртуалізацією, що забезпечує ефективність. Це призводить до більш швидкого запуску та більш ефективного використання ресурсів.

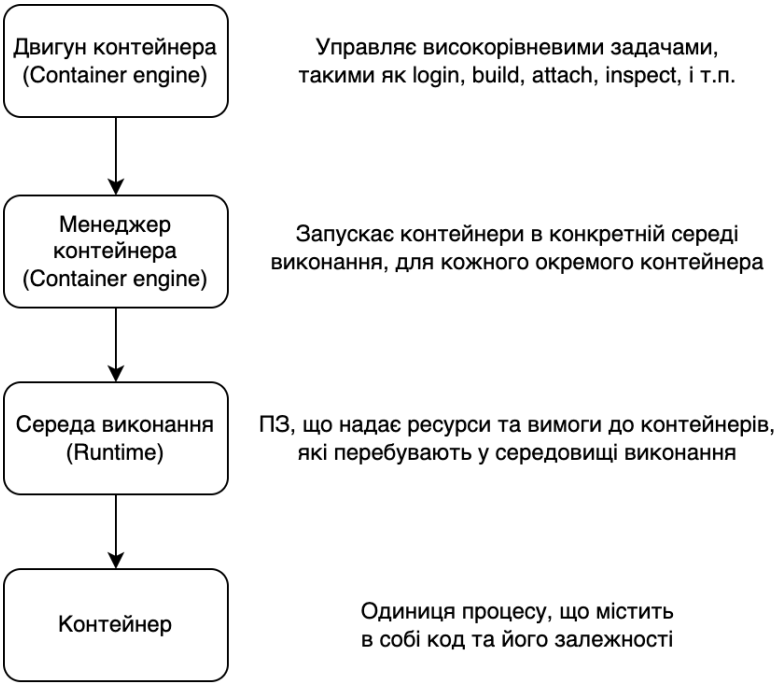

Рисунок 1 – Рівні середовища docker

Докерні контейнери, надаючи численні переваги, також створюють унікальні проблеми з безпекою. Розуміння цих загальних загроз безпеці має важливе значення для ефективного захисту контейнерних середовищ. Перед тим як ми розглянемо вразливості контейнерів, буде доречно згадати, що у 2020 році команда з кібербезпеки Aqua Security виявила новий метод, за допомогою якого зловмисники створювали шкідливі образи безпосередньо на неправильно налаштованих хостах. Представлені дослідницькі групи виступають за "динамічне сканування", яке було проведено в рамках цього дослідження, щоб виявити приховані загрози, які пропускаються при статичному аналізі вразливостей. Аналіз Prevasio, який проводився на 800 машинах протягом місяця, показав, що контейнери, запущені з багатьох підозрілих зображень, завантажували і виконували шкідливе програмне забезпечення рис. 2. Дослідники запустили антивірус Clam проти шкідливого програмного забезпечення, виявленого під час виконання, та сканер вразливостей Trivy від Aqua Security.

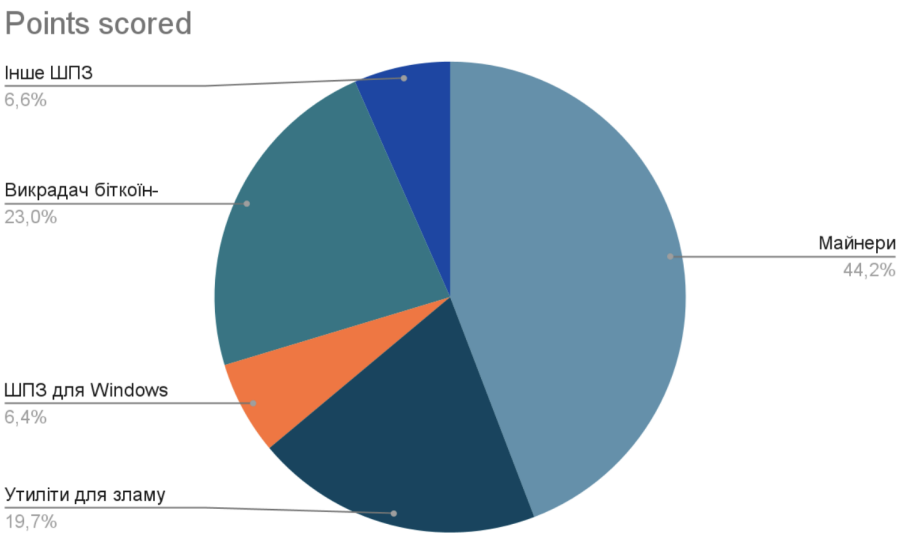

Рисунок 2 – Діаграма розподілення типів ШПЗ у образах Docker-контейнерів

Наразі відсутнє рішення, яке оцінювало б Docker-контейнери з системи хоста без вказання IP адрес, відкриття SSH портів, підкладання сертифікатів і створення повноцінного звіту. Також відсутні інструменти для оцінки системи Ubuntu 20.04 з новими критеріями захисту.

Пропонується покращити існуюче рішення для сканування віртуальних машин та створити його модуль для аналізу та оцінки docker контейнерів на базі операційної системи Ubuntu, що вирішить задачу складної конфігурації контейнерів, зменшить кількість ручної роботи, створюватиме документ з оцінкою кожного окремого контейнера, хоста, та навіть групи хостів.

Новий метод для підвищення безпеки у Docker контейнерах:

1. Виявлення Docker контейнера.
2. Збір інформації про Docker контейнер.
3. Створення Docker SSH сервер контейнеру для підключення.
4. Підключення до необхідного Docker контейнеру.
5. Аналіз Docker контейнера за критеріями CIS Ubuntu Linux 20.04 Benchmark [3].
6. Виправлення конфігурацій, що не відповідають критеріям.
   6.1. Вимкнути програми, що не використовуються.
   6.2. Зміна конфігурації мережі.
   6.3. Відключити пристрої та мережеві протоколи, що не використовуються.
7. Надсилаємо звіт адміністратору системи.

Даний метод можна реалізувати за допомогою різних мов програмування. Проте було вирішено використовувати наступні інструменти:

- Ansible – інструмент для сканування та виправлення конфігурації.
- Bash скрипт – набір команд командної оболонки, що збиратимуть метадані про контейнери, запускатимуть Ansible та розгортатимуть Docker SSH Server.
- Docker + Docker SSH Server.

Першим кроком є встановлення змінних оточення, таких як `SCAN_DIR`, що вказує каталог `container-ansible-scan`, в якому буде виконано сканування Ansible. Наступна зміна, це `ANSIBLE_CONFIGURATION`, що містить конфігурацію плейбука Ansible у форматі YAML. Вона визначає плейбук з назвою "CIS", націлений на хости з іменами "myhosts" з певними ролями, зокрема "CIS-Ubuntu-20.04-Ansible".

Наступним кроком необхідно зібрати інформацію про запущені Docker-контейнери, а найголовніше зібрати їх IP-адреси. Отримані IP-адреси запущених контейнерів Docker зберігаються у зміну `DOCKER_HOSTS_IPS`.

Наступним кроком треба підготувати Ansible для виконання. Створюється вказаний каталог `$SCAN_DIR` і записується конфігурація Ansible `$ANSIBLE_CONFIGURATION` у файл плейбука з назвою `playbook.yaml` у цьому каталозі. Клонується певний репозиторій GitHub `CIS-Ubuntu-20.04-Ansible`, якщо його не існує у поточному каталозі. Перевіряється версію `CIS_v1.1.0` ролі Ansible.

Наступним кроком є перебирання кожної IP-адреси хостів Docker у `DOCKER_HOSTS_IPS`. Для кожного хоста отримується ім'я контейнера, пов'язане з IP-адресою за допомогою команд Docker. Запускається контейнер Docker з ім'ям `docker-server-ssh` для роботи в якості SSH-сервера. Отримується IP-адреса контейнера SSH-сервера `SSH_SERVER_IP`. Створюється файл інвентаризації Ansible `hosts` з IP-адресою SSH-сервера. Запускається цикл для виконання сканування безпеки різних розділів (від 1 до 6) плейбука, створюючи звіти для кожного розділу. Після успішного сканування контейнер `docker-server-ssh` зупиняється.

Висновки

Дослідження було зосереджено на розробці системного підходу до посилення безпеки контейнерів Docker, використовуючи автоматизацію Ansible і дотримуючись Центру інтернет-безпеки CIS Ubuntu Benchmark версії 20.04.

Дослідження включало ретельне вивчення існуючих заходів безпеки Docker, виявлення потенційних вразливостей та оцінку загального стану безпеки контейнерних розгортань. Цей аналіз заклав основу для розробки методу усунення прогалин у безпеці та підвищення надійності контейнерів Docker.

Основний метод зосереджений на інтеграції Ansible, потужного інструменту автоматизації, та CIS Ubuntu Benchmark v20.04. Плейбуки Ansible були використані для автоматизації

впровадження засобів контролю безпеки, дотримання найкращих практик та коригування конфігурації в контейнерах Docker. Також було використано bash для автоматизації процесів. CIS Ubuntu Benchmark v20.04 містить повний набір рекомендацій і тестів, які слугують орієнтиром для встановлення безпечної базової конфігурації.

Таким чином, було проаналізовано ландшафт безпеки Docker, але й запропоновано практичну метод для підвищення безпеки контейнерів за допомогою інтеграції автоматизації Ansible та CIS Ubuntu Benchmark v20.04.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Security for Containers and Kubernetes. Luigi Aversa. (дата звернення: 11.12.2023).*
2. *NIST SP 800-190. Application Container Security Guide. URL: https://csrc.nist.gov/pubs/sp/800/190/final (дата звернення: 12.12.2023).*
3. *CIS Benchmark for Ubuntu 20.04. URL: https://www.cisecurity.org/benchmark/docker (дата звернення: 14.12.2023).*

Козачок Вячеслав Олександрович - студент групи 1БС-22м, факультет інформаційних технологій, спеціальність 125 Кібербезпека Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем, Вінницький національний технічний університет, Вінниця, e-mail: viacheslav.kozachok@gmail.com.

Лукічов Віталій Володимирович — к-т техн. наук, доцент, доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця, e-mail: lukichov.vitalyi@vntu.edu.ua.

Viacheslav Oleksandrovych Kozachok - a student of the 1BS-22m group, Faculty of Information Technologies, majoring in 125 Cybersecurity, Educational and Professional Program - Information and Communication Systems Security, Vinnytsia National Technical University, Vinnytsia, email: viacheslav.kozachok@gmail.com.

Vitalii Volodymyrovych Lukichov - Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Information Security, Vinnytsia National Technical University, Vinnytsia, email: lukichov.vitalyi@vntu.edu.ua.