Пивоварова У.А. Уязвимости контейнеров: риски, примеры и методы защиты // Международный журнал информационных технологий и энергоэффективности.– 2024. – Т. 9 № 12(50) с. 71–74

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056

УЯЗВИМОСТИ КОНТЕЙНЕРОВ: РИСКИ, ПРИМЕРЫ И МЕТОДЫ ЗАЩИТЫ

Пивоварова У.А.
ФГБОУ ВО САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФЕССОРА М. А. БОНЧ-БРУЕВИЧА, Санкт-Петербург, Россия (193232, г. Санкт-Петербург, просп. Большевиков, 22, корп. 1), e-mail: pivovarova.ulyana2017@yandex.ru

С ростом популярности контейнеров и технологий контейнеризации, таких как Docker и Kubernetes, киберпреступники стали активно искать уязвимости в этих средах. Контейнерные уязвимости представляют серьёзный риск для безопасности, так как могут привести к выполнению вредоносного кода и несанкционированному доступу к данным. В статье рассматриваются ключевые типы уязвимостей контейнеров, примеры реальных атак, а также способы защиты, такие как обновление образов, настройка контроля доступа и регулярное сканирование уязвимостей.

Ключевые слова: Уязвимости контейнеров, контейнеризация, Docker, Kubernetes, безопасность, контроль доступа, сканирование уязвимостей.

CONTAINER VULNERABILITIES: RISKS, EXAMPLES, AND PROTECTION METHODS

Pivovarova U.A.
ST. PETERSBURG STATE UNIVERSITY OF TELECOMMUNICATIONS NAMED AFTER PROFESSOR M. A. BONCH-BRUEVICH, St. Petersburg, Russia (193232, St. Petersburg, ave. Bolshevikov, 22, bldg. 1), e-mail: pivovarova.ulyana2017@yandex.ru

As the popularity of containers and containerization technologies like Docker and Kubernetes grows, cybercriminals are actively seeking vulnerabilities within these environments. Container vulnerabilities pose serious security risks, potentially leading to malicious code execution and unauthorized data access. The article covers key types of container vulnerabilities, examples of real attacks, and protection methods such as updating images, configuring access controls, and regular vulnerability scanning.

Keywords: Container vulnerabilities, containerization, Docker, Kubernetes, security, access control, vulnerability scanning.

Введение

В последние годы технологии контейнеризации, такие как Docker и Kubernetes, завоевали огромную популярность в мире разработки и эксплуатации приложений. Эти технологии позволяют разворачивать и управлять приложениями в независимых контейнерах, что повышает гибкость, масштабируемость и эффективность использования ресурсов. Однако контейнеризация имеет и оборотную сторону — она открывает новые векторы атак для киберпреступников, стремящихся использовать уязвимости в контейнерах и связанных инфраструктурах.

Контейнерные уязвимости могут привести к серьёзным последствиям, включая утечку данных, выполнение вредоносного кода, повышение привилегий и распространение атак в

пределах корпоративной сети. Поскольку контейнеры часто используются для развертывания критически важных приложений, такие уязвимости могут серьёзно подорвать безопасность организации. В этой статье мы рассмотрим основные виды уязвимостей, примеры реальных атак, связанные с ними риски и рекомендуемые методы защиты.

Уязвимости контейнеров

Контейнерные уязвимости могут возникать на разных уровнях архитектуры контейнеризации: от уязвимых образов контейнеров и неправильно настроенных конфигураций до недостатков в безопасности контейнерных оркестраторов, таких как Kubernetes. Одной из самых распространённых уязвимостей контейнеров является использование ненадёжных или устаревших образов, в которых содержатся уязвимые библиотеки и зависимости. По мере добавления новых слоёв и приложений в контейнеры, уязвимости могут накапливаться, создавая риски для безопасности всех систем, зависящих от таких контейнеров. Даже если контейнер безопасен при развёртывании, его зависимости могут устареть со временем и стать мишенью для атак[1].

Также одним из распространённых видов уязвимостей является ошибка настройки прав доступа и изоляции контейнеров. Контейнеры должны быть полностью изолированы друг от друга и от основной операционной системы, но в случае неправильно настроенной системы злоумышленник может выйти за пределы контейнера и получить доступ к другим контейнерам или даже к хост-системе. Этот тип уязвимости, известный как "контейнерный побег", даёт атакующему возможность расширить атаку за пределы одного контейнера, что может привести к компрометации всей сети или инфраструктуры[2].

Реальным примером уязвимости контейнеров стала уязвимость в Kubernetes CVE-2018-1002105, которая позволяла злоумышленникам отправлять запросы напрямую на серверные API, обходя стандартные проверки доступа. Это позволило атакующим получить привилегированный доступ к кластерам Kubernetes, в том числе к данным и конфиденциальной информации внутри кластера. Подобные атаки демонстрируют важность регулярного обновления и настройки доступа в Kubernetes и Docker, чтобы избежать использования устаревших версий с известными уязвимостями[3].

Сканирование контейнеров и их образов на наличие уязвимостей стало одной из критически важных практик в процессе разработки. Инструменты, такие как Docker Security Scanning, Clair и Trivy, помогают идентифицировать и устранять уязвимые зависимости, прежде чем контейнеры попадут в продакшен. Регулярное сканирование контейнерных образов на этапе сборки позволяет разработчикам быстро выявлять и исправлять потенциальные уязвимости до развертывания[4].

Помимо регулярного сканирования, важной мерой защиты является применение минимизации привилегий. В идеале каждый контейнер должен выполнять только минимально необходимые функции, а его конфигурация должна исключать возможность выполнения команд с привилегиями на уровне суперпользователя. Для этого рекомендуется использовать Docker-контейнеры с минимальным набором прав доступа и отказаться от выполнения контейнеров от имени "root". Более того, использование "модуля безопасности контейнера" (AppArmor, SELinux) может значительно уменьшить риски, связанные с доступом к хост-системе.

Ещё одна важная мера — сегментация сети для контейнеров. Контейнеры не должны иметь неограниченный доступ к ресурсам сети и другим контейнерам, если это не требуется для их работы. Сегментация и настройка сетевых политик позволяет ограничить взаимодействие контейнеров и тем самым минимизировать возможности атак, при которых злоумышленники могут получить доступ к незащищённым данным или сервисам внутри контейнерной инфраструктуры. Kubernetes Network Policies и Calico помогают ограничить сетевые взаимодействия между контейнерами, создавая дополнительные уровни защиты[5].

Также важно учитывать необходимость регулярного обновления и патчей для контейнерных систем. Обновления образов, Docker и Kubernetes являются ключевыми для предотвращения эксплуатации известных уязвимостей. Поскольку уязвимости, как правило, обнаруживаются и исправляются довольно быстро, оперативное применение обновлений минимизирует риск атак на основе известных эксплойтов.

Заключение

Контейнерные технологии, такие как Docker и Kubernetes, кардинально изменили подход к разработке и развертыванию приложений, предложив гибкие и масштабируемые решения для современных ИТ-систем. Однако вместе с их преимуществами появились и новые угрозы безопасности. Уязвимости контейнеров, такие как использование ненадёжных образов, ошибки в настройке привилегий и конфигураций, а также недостаточная изоляция, создают серьёзные риски для данных и инфраструктуры организаций.

Эффективная защита контейнерных сред требует комплексного подхода: регулярного сканирования образов и контейнеров, минимизации привилегий, настройки сетевой изоляции и своевременного обновления всех компонентов. В условиях, когда контейнеризация становится стандартом в мире ИТ, меры безопасности должны занимать центральное место в процессе разработки и эксплуатации контейнеров.

Список литературы

1. Котенко И. В. и др. Модель человеко-машинного взаимодействия на основе сенсорных экранов для мониторинга безопасности компьютерных сетей. – 2018.
2. Миняев А. А. Метод оценки эффективности системы защиты информации территориально-распределенных информационных систем персональных данных //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020). – 2020. – С. 716-719.
3. Леснова Е. М., Пестов И. Е. Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных //Региональная информатика и информационная безопасность. – 2018. – С. 236-240.
4. Горбань С. А., Красов А. В., Цветков А. Ю. Оценка эффективности механизмов контроля правами доступа в ОС Linux //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). – 2023. – С. 345-348.
5. Волкогонов В. Н. и др. Применение физически неклонируемых функций для выполнения аутентификации в среде интернета вещей //Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.

References

1. Kotenko I. V. et al. A human-machine interaction model based on touchscreens for monitoring the security of computer networks. – 2018.
2. Minyaev A. A. Method of evaluating the effectiveness of the information protection system of geographically distributed personal data information systems //Actual problems of infotelecommunications in science and education (APINO 2020). – 2020. – pp. 716-719.
3. Lesnova E. M., Pestov I. E. Development of a method for detecting and correcting errors for a distributed information network based on big data //Regional informatics and information security. – 2018. – pp. 236-240.
4. Gorban S. A., Krasov A.V., Tsvetkov A. Yu. Assessment of the effectiveness of access rights control mechanisms in Linux OS //Actual problems of infotelecommunications in science and education (APINO 2023). – 2023. – pp. 345-348.
5. Volkogonov V. N. et al. The use of physically non-cloned functions to perform authentication in the Internet of Things environment //Current problems of infotelecommunications in science and education. - 2021. – pp. 409-414.