ОСОБЕННОСТИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В ПРОГРАММНОМ СРЕДСТВЕ DOCKER

Ярошенко А. Л., Лось Н. А., Пискун Г. А.
Научный руководитель: канд. техн. наук, доц. Пискун Г. А.
Белорусский государственный университет информатики и радиоэлектроники, Беларусь
E-mail: yarashenkabsuir@gmail.com

Аннотация — Рассмотрены основные уязвимости программного обеспечения Docker с использованием командной строки. Приведены возможные варианты решения для обеспечения безопасности.

1. Введение

Docker в значительной мере изменил подход к настройке серверов, поддержке и доставке приложений. Его поддерживают и используют такие компании, как eBay, PayPal, а также корпорация Google. Данное решение дает возможность разработчикам изменять архитектуру своих программных продуктов, разбивая их на более мелкие компоненты (микросервисы), которые будут запускаться в изолированных контейнерах, что позволит достичь большего ускорения, параллелизации исполнения и надежности.

2. Основная часть

Docker — программное обеспечение, позволяющее автоматизировать процесс развертывания приложении, в основе которого лежит виртуализация на уровне операционной системы. Данный продукт предоставляется для всех основных операционных систем: Windows, Linux, MacOS.

Основными единицами данного инструмента являются образ и контейнер [1].

Если в контейнере производились какие-либо изменения, то полная его остановка приведет к исходному состоянию (образ). Поэтому, при необходимости, после выполнения некоторых действий в контейнере, есть возможность сохранить его состояние и использовать новый образ с уже сохраненным состоянием.

Каждый контейнер изначально предназначен для виртуализации одного процесса. Однако, используя bash-скрипты, эту возможность можно расширить.

Основным отличием данного типа виртуализации с помощью Docker является его полная изоляция от окружающего мира и от основной операционной системы. Таким образом, все необходимые процессы происходят в фоновом режиме.

Однако, помимо всего этого существуют некоторые проблемы, связанные с безопасностью.

Одной из проблем безопасности является изначальное наследование различных дистрибутивов и его запуск под пользователем root. Это означает, что злоумышленник сможет зайти в запущенный контейнер без пароля и произвести необходимые ему манипуляции беспрепятственно. Кроме того, запуск контейнера под пользователем root сопровождается полным отсутствием разграничения прав для домашней директории. Для данной проблемы можно предложить два решения:

— добавление пароля к пользователю root с использованием passwd (добавление парольного доступа к пользователю). Однако, данное решение требует описание дополнительной инструкции и не решает вторую часть проблемы – разграничение доступа к домашним директориям и процессам.

— создание нового независимого пользователя (например, Ubuntu) с собственной домашней директорией с использованием passwd или без. Однако, данное решение включает в себя использование большего количества инструкции, нежели первое решение, однако оно решает данную проблему.

В отличие от современных систем виртуализации, Docker имеет свой собственный публичный и приватный репозиторий образов, включающий как официальные сборки, так и пользовательские, с которыми необходимо быть осторожными. При этом, можно воспользоваться как уже готовым публичным репозиторием Docker Hub (за приватные репозитории взымается отдельная плата), а также развернуть свой собственный репозиторий под названием Docker Trusted Registry на собственном сервере и использовать приватные репозитории бесплатно (однако, требуется лицензия). Оба типа репозитория используют сканирование каждого образа контейнера при помощи встроенного Docker's Security Scanning Service. Данная опция позволяет сканировать каждый образ и проверить его на основные уязвимости по базе данных CVE (Common Vulnerabilities and Exposures), которые впоследствии можно исправить и сделать образ Docker более безопасным.

Docker позволяет автоматизировать процесс развертывания приложений и ускорить процесс доставки в несколько раз. Однако, чем больше приложений, тем большие последствия могут возникнуть (например, при DDoS-атаке). Чтобы была возможность узнать, что произошло с тем или иным контейнером, необходимо вести определенные записи в журнале. Для этого используются логи, которые включают в себя как выходные данные, так и ошибки на уровне запущенного процесса.

3. Заключение

Таким образом, были выявлены некоторые уязвимости в программном продукте Docker, а также представлены методы обеспечения безопасности.

Выявлена возможность проектирования более гибкой архитектуры путем разбиения больших программных продуктов на более мелкие компоненты при использовании Docker.

4. Список литературы

[1] Моуэт, Э. Использование Docker / Э. Моуэт. — O'Reilly Media, 2017. — 354 с.
[2] Вольф, Э. Continuous Delivery. Практика непрерывных апдейтов / Э. Вольф. — СПб. : Питер, 2017. — 320 с.

SAFETY FEATURES IN THE DOCKER SOFTWARE

Yarashenka A. L., Los N. A., Piskun G. A.
Scientific adviser: Piskun G. A.
Belarusian State University of Informatics and Radioelectronics, Belarus

Abstract — The main vulnerabilities of the Docker software using the command line are discussed. Possible solutions for security solutions are given.

Библиотека БГУИР