Инженерный вестник Дона, №6 (2024)
ivdon.ru/ru/magazine/archive/n6y2024/9291

Об увеличении поверхности атаки при переходе на микросервисную архитектуру

К.В. Шепелев

Финансовый университет при Правительстве Российской Федерации

Аннотация: Рассматривается сетевая составляющая перехода от монолитной архитектуры приложения к микросервисам. Представлена динамика увеличения поверхности атаки при переходе на микросервисную архитектуру на примере приложения, предназначенного для ведения централизованного учета серверного оборудования и фиксации ответственных за его эксплуатацию сотрудников. Предложены меры, направленные на повышение защищенности сетевой инфраструктуры систем оркестрации контейнеров.
Ключевые слова: информационная безопасность, микросервисная архитектура, безопасность контейнеризации, сетевые взаимодействия, сеть контейнеров, система оркестрации контейнеров, микросервис, контейнер, переход от монолита на микросервисы, увеличение поверхности атаки

На текущий момент сложно представить современную компанию без межсетевых экранов. Долгое время они обеспечивают безопасность сетевой инфраструктуры организаций, отражая атаки «нулевого дня» [1], предотвращая вторжения и ограничивая доступ. Но в связи с развитием тенденции перехода от монолитных к микросервисным приложениям и системам, старые подходы обеспечения сетевой безопасности [2] перестают быть актуальными. Современные межсетевые экраны не учитывают специфику работы микросервисных приложений, функционирующих внутри хостовых операционных систем, образующих системы оркестрации контейнеров [3].

Неприменимые к стандартным инфраструктурам угрозы информационной безопасности систем оркестрации контейнеров, описанные в работах [4, 5], подтверждают особенности функционирования таких систем.

В работе [6] рассмотрен процесс перехода с монолитной на микросервисную архитектуру. Такие изменения влекут за собой пересмотр

подходов обеспечения информационной безопасности, одной из важнейших областей которой является защита сетевых взаимодействий.

На рис. 1 представлена схема сетевых взаимодействий монолитного приложения, функционирующего на одном сервере. Приложение предназначено для ведения централизованного учета серверного оборудования и фиксации ответственных за его эксплуатацию сотрудников. Для получения данных об IP-адресе или FQDN-имени сервера система обращается к корпоративному DNS-серверу, для получения данных о сотруднике – к LDAP-каталогу, расположенному на контроллере домена организации. Обновление списка оборудования выполняется через proxy-сервер. Система авторизует пользователей по средствам протокола аутентификации Kerberos. Пользователь системы обращается к веб-интерфейсу приложения с использованием протокола HTTPS.

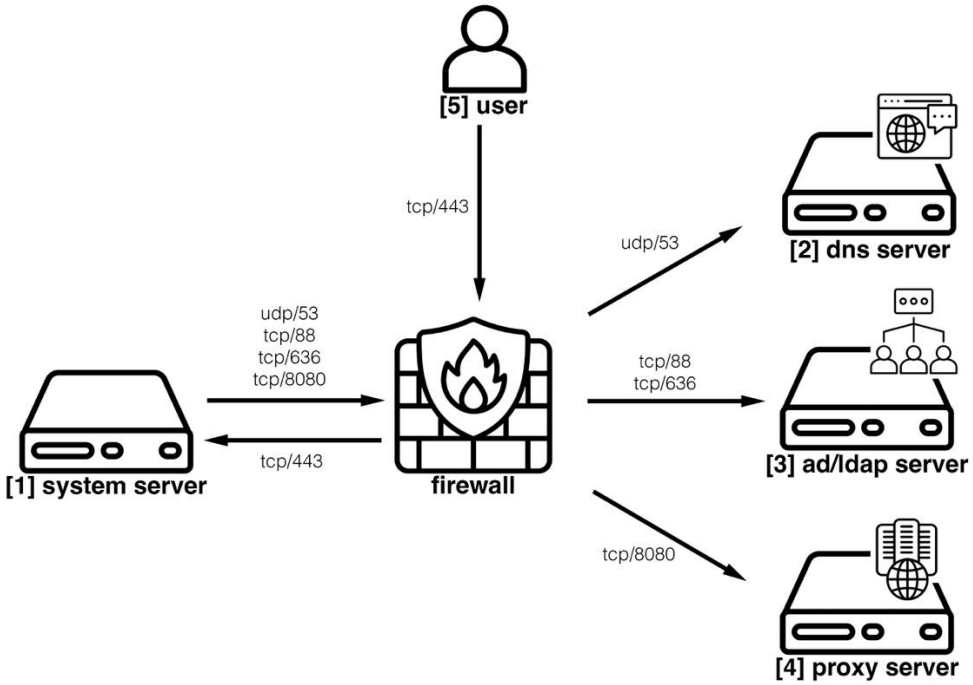

Рис. 1 – Сетевые взаимодействия монолитной системы

Для корректной работы такой системы требуется 5 правил межсетевого экранирования, последнее из которых блокирует весь сетевой трафик, не

подпадающий под правила выше. Список правил межсетевого экранирования для корректной работы монолитной системы представлен в таблице №1.

Таблица №1

Правила межсетевого экранирования для работы монолитной системы

№	Источник	Назначение	Порт	Действие
1	(1)	(2)	udp/53	Разрешено
2	(1)	(3)	tcp/88 tcp/636	Разрешено
3	(1)	(4)	tcp/8080	Разрешено
4	(5)	(1)	tcp/443	Разрешено
5	Все	Все	Любой	Запрещено

На рис. 2 приведена схема сетевых взаимодействий рассматриваемого приложения после его перевода на микросервисную архитектуру. Как видно на рис. 2, монолитное приложение было разбито на 6 контейнеров, каждый из которых выполняет строго определенную задачу: frontend – веб-интерфейс и вся логика приложения; dns-check – обращается в корпоративный DNS сервис для разрешения имен добавляемых серверов; user-auth – обращается к контроллеру домена для авторизации пользователей; user-check – обращается в LDAP-каталог за информацией о добавляемых пользователях; database – база данных; update – обновление списка оборудования.

В системе оркестрации контейнеров за исходящий и входящий трафик отвечают разные компоненты, поэтому на рис. 2 потоки из и в кластер разделены на две сущности (1.1) и (1.2) соответственно. Сущность (1.2) содержит меньшее количество IP-адресов, чем сущность (1.1), поскольку исходящий трафик может быть инициирован любым рабочим сервером из состава кластера оркестратора, в то время как входящий трафик обрабатывают выделенные под эту задачу серверы.

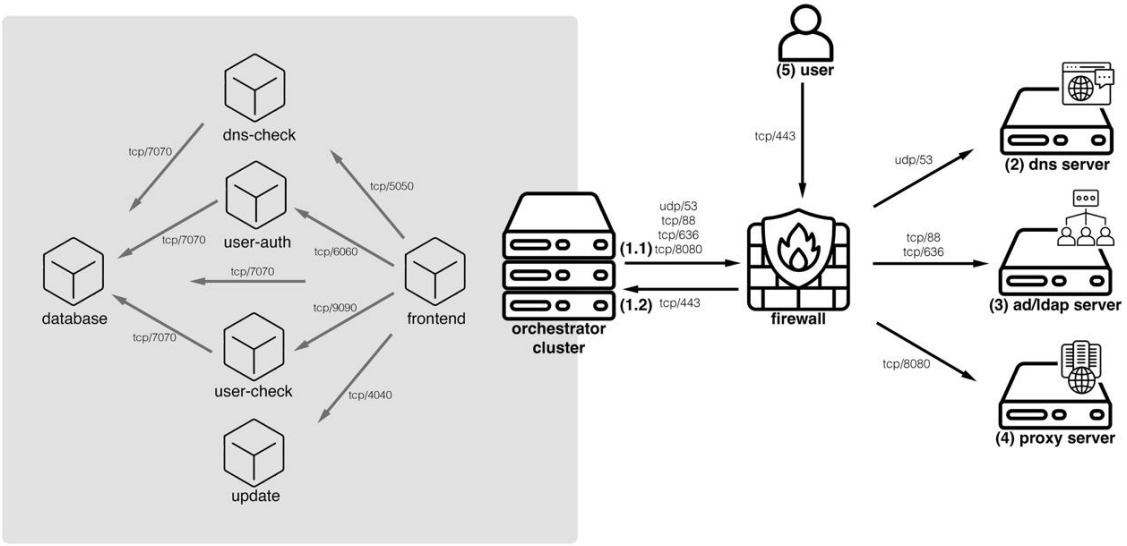

Рис. 2. – Сетевые взаимодействия системы после перехода на микросервисную архитектуру

В таблице № 2 приведены правила межсетевого экранирования для корректного функционирования микросервисного приложения.

Таблица № 2

Правила межсетевого экранирования для работы приложения на микросервисной архитектуре

№	Источник	Назначение	Порт	Действие
1	(1.1)	(2)	udp/53	Разрешено
2	(1.1)	(3)	tcp/88 tcp/636	Разрешено
3	(1.1)	(4)	tcp/8080	Разрешено
4	(5)	(1.2)	tcp/443	Разрешено
5	Все	Все	Любой	Запрещено

Общее число правил не изменилось, но под сущностью (1.1) на рис. 2 понимается уже не один IP-адрес сервера, указанного на рис. 1, а IP-адреса всех серверов кластера системы оркестрации контейнеров. Количество

промежуточных сетевых устройств увеличивается, что в свою очередь способствует расширению поверхности атаки.

Стоит отметить, что системы оркестрации контейнеров очень редко используются под одну конкретную систему или приложение. Так на рис. 3 к рассматриваемому ранее приложению для более эффективной утилизации ресурсов, выделенных под систему оркестрации контейнеров, добавляются два новых сервиса. Все три микросервисных приложения доступны через веб-интерфейс, но выполняют разные функции. Два новых сервиса будут отправлять электронную почту администратору, для этого необходимо организовать сетевой доступ к корпоративному почтовому серверу. При этом сервис № 2 не будет обращаться к LDAP-каталогу и ему не нужен доступ к proxy-серверу, а сервис № 3, в свою очередь, будет иметь локальную авторизацию, доступ к DNS-серверу и контроллеру домена для него будет избыточным.

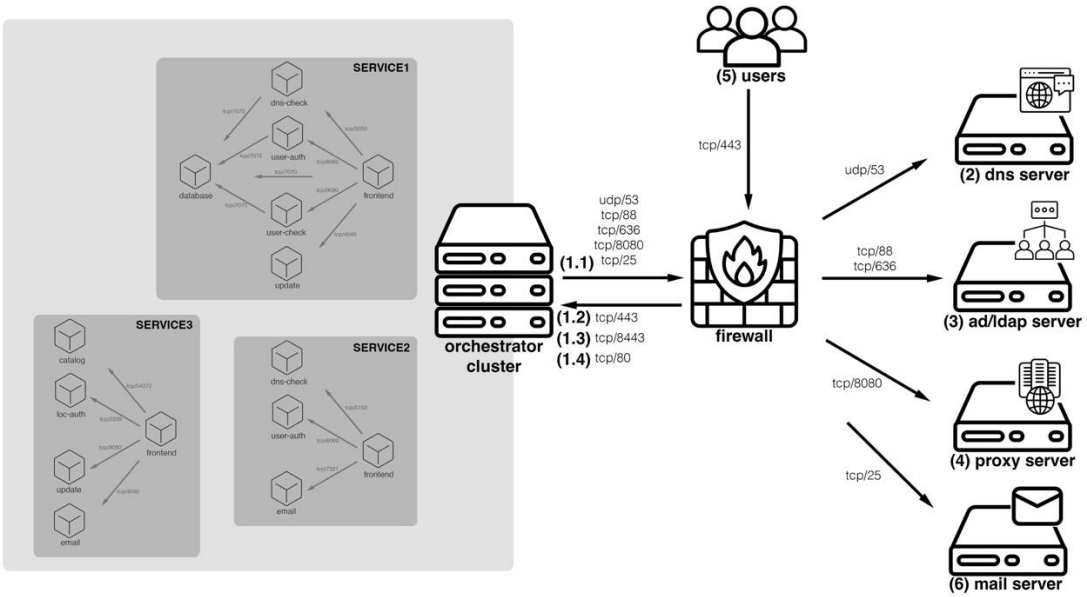

Рис. 3. – Использование системы оркестрации контейнеров различными приложениями с микросервисной архитектурой

Число доступных пользователям сервисов увеличивается, поэтому появляется возможность выдавать более гранулированный доступ к

конкретным веб-интерфейсам на основании информации о выделенных системой оркестрации под конкретные приложения портах. В таблице № 3 представлены правила межсетевого экранирования для корректной работы трех приложений, одновременно функционирующих в одной системе оркестрации контейнеров.

Таблица № 3

Правила межсетевого экранирования для работы приложений в системе оркестрации контейнеров

№	Источник	Назначение	Порт	Действие
1	(1.1)	(2)	udp/53	Разрешено
2	(1.1)	(3)	tcp/88 tcp/636	Разрешено
3	(1.1)	(4)	tcp/8080	Разрешено
4	(1.1)	(6)	tcp/25	Разрешено
5	(5)	(1.2)	tcp/443	Разрешено
6	(5)	(1.3)	tcp/8443	Разрешено
7	(5)	(1.4)	tcp/80	Разрешено
8	Все	Все	Любой	Запрещено

Переиспользование ресурсов влечет за собой увеличение количества и расширение уже существующих правил сетевых взаимодействий. Так, в сущности (5) увеличивается число конечных потребителей, также больше сервисов требуют публикации и у части из них появляется потребность в доступе к корпоративному почтовому серверу. К сожалению, существующие средства межсетевого экранирования не способны обеспечить защиту микросервисных приложений и фильтровать их исходящий трафик. Самый распространённый вариант, обеспечивающий максимальную отказоустойчивость в этом случае – открытие сетевого доступа от IP-адресов всех серверов кластера системы оркестрации контейнеров, отвечающих за

исходящий трафик, до всех корпоративных сервисов организации. Так у части приложений, функционирующих внутри системы оркестрации контейнеров, появляется избыточный неконтролируемый доступ.

Поскольку при микросервисном подходе все приложения разделяют между собой ресурсы одного кластера, не менее важно обеспечивать контроль их внутренних сетевых соединений. Ошибка в конфигурации или злонамеренные действия могут стать причиной несанкционированного доступа контейнера одного приложения к контейнеру другого. Чтобы исключить вектор атаки на микросервисы, основанный на сетевых соединениях контейнеров, важно на ранних этапах выявлять аномалии сетевого трафика внутри функционирующих систем оркестрации контейнеров.

Использование системы оркестрации контейнеров несколькими приложениями является самым распространённым вариантом, применяемым организациями, поскольку в этом случае достигается максимальный эффект от имеющихся ресурсов за счет их утилизации. Однако такой подход значительно расширяет поверхность атаки, поскольку у микросервисных приложений появляются потенциальные избыточные доступы к инфраструктуре и друг к другу. Чтобы избежать расширения площади атаки при развертывании системы оркестрации контейнеров на операционной системе с ядром Linux, можно воспользоваться enhanced Berkeley Packet Filter (eBPF), который подробно описан в работах [7, 8] или использовать встроенный механизм сетевых ограничений функционирующих контейнеров и их пространств имен, описанные в документации [9, 10]. Инструменты на базе eBPF в большинстве своем подходят для обнаружения атак, в то время как функции ограничения сетевых взаимодействий, встроенные в системы оркестрации, правильней относить к предотвращающим средствам, которые необходимо внедрять на ранних стадиях жизненного цикла систем.

Литература

1. Безродных О.А. Использование различных межсетевых экранов нового поколения для защиты корпоративной сети от сетевых атак // Инновации. Наука. Образование. 2022. № 50. С. 1693–1702.

2. Иконников С.Е., Ермакова А.Е. Построение защищенной локальной сети организации на основе межсетевого экранирования // Интеллектуальные транспортные системы: материалы Международной научно-практической конференции, Москва, 26 мая 2022 года. Москва: Российский университет транспорта. 2022. С. 187–192.

3. Мареев Н.А. Сравнение контейнерных оркестраторов для реализации микросервисной архитектуры // Технологии информационного общества: Сборник трудов XVII Международной отраслевой научно-технической конференции, Москва, 02–03 марта 2023 года. Москва: ООО «Издательский дом Медиа паблишер». 2023. С. 228–230.

4. Гурбатов Г.О., Паничев А.Д., Ушаков И.А. Обеспечение безопасности Kubernetes // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сборник научных статей: в 4-х томах, Санкт-Петербург, 24–25 февраля 2021 года / Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича. Том 1. 2021. С. 282–286.

5. Жилина А.А., Володин С.М. Обеспечение безопасности виртуальной инфраструктуры, построенной на базе микросервисной архитектуры с использованием оркестратора KUBERNETES // Информационная безопасность: вчера, сегодня, завтра: Сборник статей по материалам V Международной научно-практической конференции, Москва, 14 апреля 2022 года. Москва: Российский государственный гуманитарный университет. 2022. С. 66–72.

6. Луценко Д.Ю., Полякова Л.П. Разбиение монолитного приложения на микросервисы с использованием паттерна Strangler // Информационные технологии в управлении и экономике. 2021. № 3(24). С. 82–87.

7. Ларин Д.В., Гетьман А.И. Средства захвата и обработки высокоскоростного сетевого трафика // Труды Института системного программирования РАН. 2021. Т. 33. № 4. С. 49–68. DOI 10.15514/ISPRAS-2021-33(4)-4.

8. Hedam N. BPF - From a Programmer's Perspective, 2023. URL: researchgate.net/publication/349173667 – DOI 10.13140/RG.2.2.33688. 11529/4.

9. Apache Mesos Documentation // apache.org. URL: mesos.apache.org/documentation/latest/isolators/network-port-mapping/ (дата обращения: 15/05/2024).

10. Kubernetes Documentation // kubernetes.io. URL: kubernetes.io/docs/concepts/services-networking/network-policies/ (дата обращения: 10/05/2024).

References

1. Bezrodnyh O.A. Innovacii. Nauka. Obrazovanie. 2022, № 50, pp. 1693–1702.

2. Ikonnikov S.E., Ermakova A.E. Intellektual'nye transportnye sistemy: materialy Mezhdunarodnoj nauchno-prakticheskoj konferencii, Moskva, 26 maya 2022 goda. Moskva: Rossijskij universitet transporta. 2022, pp. 187–192.

3. Mareev N.A. Tekhnologii informacionnogo obshchestva: Sbornik trudov XVII Mezhdunarodnoj otraslevoj nauchno-tekhnicheskoj konferencii, Moskva, 02–03 marta 2023 goda. Moskva: OOO "Izdatel'skij dom Media pablisher". 2023, pp. 228–230.

4. Gurbatov G.O., Panichev A.D., Ushakov I.A. Aktual'nye problemy infotelekommunikacij v nauke i obrazovanii: sbornik nauchnyh statej: v 4-h tomah, Sankt-Peterburg, 24–25 fevralya 2021 goda. Sankt-Peterburgskij gosudarstvennyj

universitet telekommunikacij im. prof. M.A. Bonch-Bruevicha. Vol. 1, 2021, pp. 282–286.

5. Zhilina A.A., Volodin S.M. Informacionnaya bezopasnost': vchera, segodnya, zavtra: Sbornik statej po materialam V Mezhdunarodnoj nauchno-prakticheskoj konferencii, Moskva, 14 aprelya 2022 goda. Moskva: Rossijskij gosudarstvennyj gumanitarnyj universitet. 2022, pp. 66–72.

6. Lutsenko D.Yu., Polyakova L.P. Informacionnye tekhnologii v upravlenii i ekonomike. 2021, № 3(24), pp. 82–87.

7. Larin D.V., Get'man A.I. Trudy Instituta sistemnogo programmirovaniya RAN. 2021, vol. 33, № 4, pp. 49–68. DOI 10.15514/ISPRAS-2021-33(4)-4.

8. Hedam N. BPF - From a Programmer’s Perspective, 2023. URL: researchgate.net/publication/349173667 DOI 10.13140/RG.2.2.33688. 11529/4.

9. Apache Mesos Documentation URL: mesos.apache.org/ documentation/latest/isolators/network-port-mapping/ (date assessed: 15/05/2024).

10. Kubernetes Documentation URL: kubernetes.io/docs/ concepts/services-networking/network-policies/ (date assessed: 10/05/2024).

Дата поступления: 17.04.2024

Дата публикации: 30.05.2024