

Научная статья
DOI 10.66424/2071-8217-2026-3-4
УДК 004.056

МОДЕЛЬ РИСК-ОРИЕНТИРОВАННОГО ПЛАНИРОВАНИЯ В КОНТЕЙНЕРНЫХ ОРКЕСТРАТОРАХ С УЧЕТОМ ГОРИЗОНТАЛЬНОГО ПЕРЕМЕЩЕНИЯ АТАКУЮЩЕГО

В. А. Веденский*

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ [*sci@ibks.spbstu.ru](mailto:sci@ibks.spbstu.ru)

ДЛЯ ЦИТИРОВАНИЯ

Веденский В. А. Модель
риск-ориентированного
планирования в контейнерных
оркестраторах с учетом
горизонтального перемещения
атакующего // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 60–71.
DOI: 10.66424/2071-8217-2026-3-4

ПОСТУПИЛА 02.07.2026

ПРИНЯТА 28.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Веденский В. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Стандартные планировщики контейнерных оркестраторов ориентируются на ресурсные метрики и политики распространения, игнорируя риски безопасности, связанные с горизонтальным перемещением между контейнерами и узлами кластера. Представлена модель риск-ориентированного размещения подов, которая учитывает три канала горизонтального перемещения атакующего: общие системные вызовы на узле (риск побега из контейнера), расхождение ролевой модели управления доступом (RBAC) и сетевые взаимодействия между подами. Вводятся вероятностные оценки компрометации контейнера как функции безопасности узла. Предложен жадный алгоритм последовательного назначения контейнеров, пригодный для использования в реальных планировщиках. Проведен сравнительный анализ с основными подходами к защите контейнерных сред. Результаты исследования могут быть использованы при разработке расширений планировщика Kubernetes, интегрирующих безопасность на уровне размещения.

КЛЮЧЕВЫЕ СЛОВА

Kubernetes, планирование размещения, безопасность контейнеров, горизонтальное перемещение

Original article
DOI 10.66424/2071-8217-2026-3-4

A MODEL OF RISK-AWARE SCHEDULING IN CONTAINER ORCHESTRATORS CONSIDERING THE ATTACKER'S HORIZONTAL MOVEMENT

V. A. Vedenskii*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ [*sci@ibks.spbstu.ru](mailto:sci@ibks.spbstu.ru)

FOR CITATION

Vedenskii V. A. A model of risk-aware scheduling in container orchestrators considering the attacker's horizontal movement. *Problems of information security. Computer systems*. 2026. No. 3, pp. 60–71.
DOI: 10.66424/2071-8217-2026-3-4 (In Russian)

RECEIVED 02.07.2026

ACCEPTED 28.07.2026

PUBLICATION 13.10.2026

ABSTRACT

Common container orchestrator schedulers focus on resource metrics and distribution policies, ignoring the security risks associated with lateral movement between containers and cluster nodes. This paper presents a risk-based pod placement model that accounts for three channels of lateral movement: shared system calls on a node (risk of container escape), role-based access control mismatches, and network interactions between pods. Probabilistic estimates of container compromise are introduced as a function of node security. A greedy algorithm for sequential container assignment suitable for use in real-world schedulers is proposed. A comparative analysis with leading approaches to securing container environments is provided. The results of this study can be applied for the development of Kubernetes scheduler extensions that integrate placement-level security.

KEYWORDS

Kubernetes, scheduling, container security, lateral movement

1. ВВЕДЕНИЕ

Kubernetes – широко известная платформа оркестрации с открытым исходным кодом, применяемая для автоматизации развертывания, масштабирования и управления контейнеризированными приложениями. Данная платформа широко используется предприятиями и поставщиками облачных услуг [1]. Однако использование контейнеризированных приложений несет в себе определенные риски информационной безопасности.

Контейнеры менее изолированы от других контейнеров и узлов, чем виртуальные машины, что обусловлено использованием общего ядра операционной системы. Данная особенность работы контейнеризированных приложений порождает угрозу побега из контейнера и способствует горизонтальному перемещению атакующего между приложениями и узлами в кластере. Свободное размещение частей приложения на разных узлах кластера [2, 3] создает излишнюю сетевую связность узлов, что в случае компрометации контейнеров или узлов также способствует горизонтальному перемещению атакующего. Сама инфраструктура платформы оркестрации создает риски информационной безопасности при ненадлежащей конфигурации или компрометации узла, контейнера. Например, расхождение

ролевой модели управления доступом (RBAC) может расширить полномочия атакующего [4]. При компрометации узла у нарушителя могут появляться привилегии всех контейнеров, размещенных на узле, так как RBAC основан на использовании токенов, доступ к которым можно получить с узла. Пример реализации горизонтального перемещения представлен на рисунке.

Данные риски можно уменьшить, используя безопасно-ориентированное размещение контейнеров в кластере, минимизируя связи узлов через сетевую связность контейнеров, контейнеров – через узел, а также контролируя распределение привилегий RBAC по кластеру.

Базовый планировщик Kubernetes распределяет поды (под – единица планирования рабочей нагрузки Kubernetes, может включать в себя от одного до нескольких контейнеров) по узлам кластера, оптимизируя ресурсные метрики: загрузку CPU, потребление памяти, сетевую задержку. Однако безопасность приложений и кластера при этом не рассматривается как целевая функция. Встроенные механизмы управления размещением подов позволяют задавать статические правила, но не оценивают критичность совместного размещения конкретных подов на конкретных узлах, что оставляет риски горизонтального перемещения.

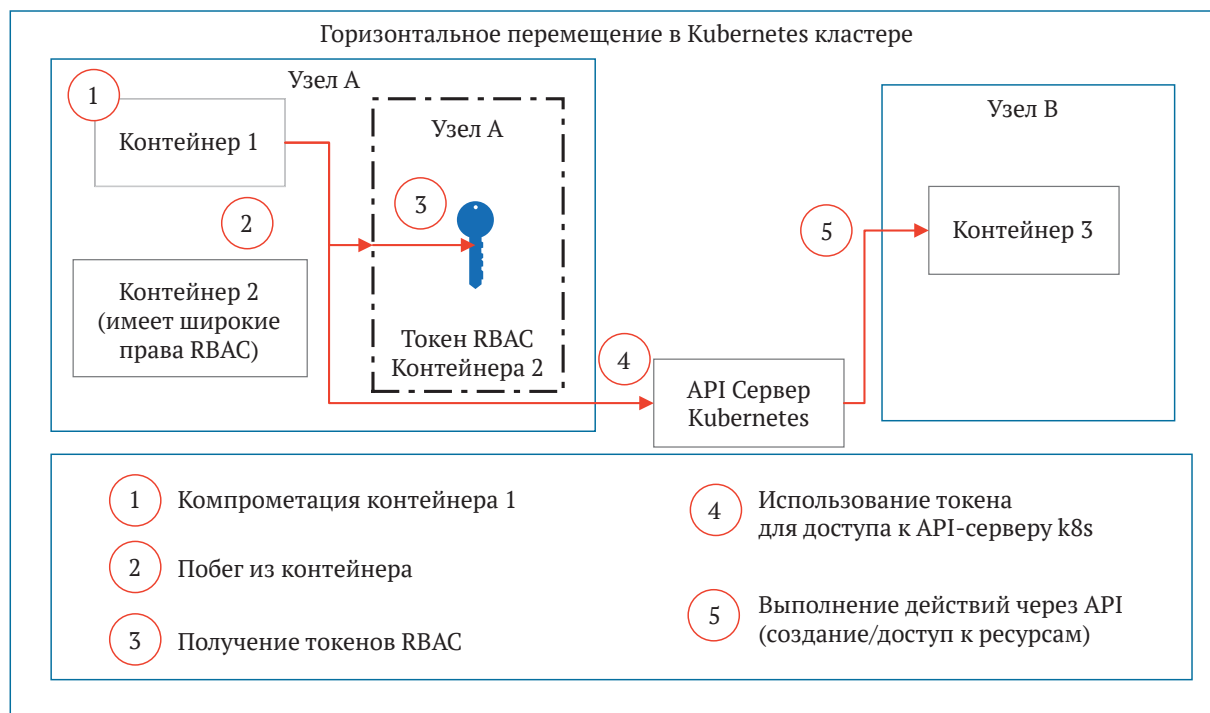


Схема реализации горизонтального перемещения в RBAC

Scheme for implementing horizontal movement in RBAC

Исследования [5, 6] подчеркивают, что, несмотря на многочисленные исследования по безопасности контейнеров, интеграция безопасности в процесс планирования остается все еще нерешенной проблемой. В анализе адаптивных методов планирования [5] отмечено, что ни одна из стратегий не учитывает безопасность как фактор. В исследовании [6] систематизированы угрозы (побеги из контейнера, атаки на образы, компрометация оркестратора) и существующие контрмеры, но отмечено отсутствие решений, проактивно влияющих на размещение контейнеров.

В этой связи поставлена задача разработать метод планирования размещения подов по узлам кластера, учитывающий все три канала горизонтального размещения.

2. АНАЛИЗ ИЗВЕСТНЫХ ПОДХОДОВ

Методы, основанные на применении политик безопасности [6–8], связаны с декларируанием общих политик безопас-

ности в кластере, например, с ограничением запуска привилегированных подов, использованием пользователя с привилегиями администратора, разделением кластера на зоны и сетевую изоляцию подов и т.д. Применение данных мер является обязательным в кластере, так как они обеспечивают базовую защиту от атак, связанных с неверной конфигурацией, что существенно снижает пространство угроз. Однако применение только данных мер не является исчерпывающим, риски побега за пределы контейнера и горизонтального перемещения остаются. Также недостатком данных методов является сложная ручная конфигурация политик.

Методы обеспечения безопасности на основе планирования размещения нацелены на снижение потенциального риска горизонтального перемещения и побега из контейнера и подразделяются на три группы по принципу своей работы:

1. Планирование на основе анализа критичности приложения. Основная идея состоит в том, чтобы ранжировать приложения по степени уязвимости к атакам

совместного размещения и расположить их в кластере таким образом, чтобы минимизировать данный риск. Оценка приложений может проводиться на основе анализа профиля системных вызовов доступных приложений. Веса критичности системных вызовов могут назначаться администраторам системы [9] или на основе баллов CVSS системных вызовов [10, 11]. Также оценка уязвимости приложения может проводиться на основе экспертного заключения при анализе исходного кода приложения или других внешних источников [12]. Отдельно стоит отметить, что в работах [11, 12] при планировании размещения также оценивается и сетевое взаимодействие подов. Вместе с тем данный подход к размещению подов не учитывает возможности горизонтального перехода по каналу распространения привилегий.

2. Планирование на основе анализа сетевого взаимодействия. Метод состоит в том, чтобы автоматизировать процесс сегментации сети и изоляции подов и узлов за счет интент-ориентированного планирования, где администратор кластера задает только высокоуровневые требования, а планировщик (Diktyo) совместно с менеджером безопасности (GrassHopper) динамически определяют состав сетевых сегментов, назначают узлы в эти сегменты и настраивают группы безопасности на уровне виртуальных машин без ручного вмешательства. Развитие данного подхода идет по нескольким направлениям. Так, в работе [13] предложен первый рабочий прототип, где безопасность обеспечивалась реактивно (блокировка неразрешенного трафика), а сегментация оставалась статической. В работе [14] развивают этот подход до проактивной и декларативной модели – через Segmentation CRD и веб-интерфейс для администратора, позволяющий динамически переопределять сегменты в реальном времени. Однако все эти решения не учитывают угрозы горизонтального перемещения через узел по средствам побега из контейнера и распространения привилегий RBAC.

3. Планирование на основе анализа ролевой модели. Основная идея данного подхода заключается в том, что стандартные

планировщики Kubernetes, оптимизирующие качество обслуживания (QoS) и загрузку ресурсов, не учитывают RBAC-привилегии подов, что приводит к совместному размещению на одном узле подов с разными уровнями доступа. Это создает риски повышения привилегий, горизонтального перемещения при компрометации узла через побег из контейнера. Для количественной оценки и минимизации данных рисков в исследовании [15] предложена метрика Extraneous Risk Privileges (ERP) для оценки разнородности и критичности привилегий на узлах кластера. Пусть на узле n совместно работают поды с наборами привилегий $\{priv_i^j\}$, объединенный вектор риска узла определяется по формуле $PRIV^j = \bigcup_i priv_i^j$.

Для конкретного пода p_i дополнительные привилегии, которые может получить атакующий после компрометации узла n_j (например, через побег из контейнера), вычисляются по формуле $E_i^j = PRIV^j \oplus priv_i^j$. Итоговая оценка ERP для пода p_i на узле n_j оценивается по формуле $ERP_i^j = WE_i^j$, где W – весовой вектор оценки важности ресурсов кластера. Суммарный риск узла и кластера выражается по формулам $ERP^j = \sum_i ERP_i^j$ и $ERP = \sum_j ERP^j$ соответственно. Однако данный подход при планировании размещения не учитывает сетевую связность подов.

Группа методов обеспечения безопасности через альтернативную среду запуска контейнеров реализует изоляцию контейнеров на уровне узла за счет изоляции системных ресурсов и процессов хвостовой операционной системы друг от друга, чтобы предотвратить побег из контейнера даже в случае эксплуатации уязвимостей [16–18]. Kata Containers запускает каждый контейнер в облегченной виртуальной машине (Lightweight VM) с собственным ядром, что обеспечивает аппаратный уровень изоляции. gVisor реализует пользовательское ядро на Go, перехватывающее системные вызовы. TZ-Container использует ARM TrustZone для аппаратной изоляции. Vpfbox реализует точное ограничение процессов через eBPF, GyroidOS

сокращает атакуемую поверхность ядра, эмулируя избыточные системные вызовы через бинарную перезапись ELF или `sessomp notify`. Данные решения работают на уровне изоляции отдельных контейнеров, не затрагивая их размещение на узлах. Они являются альтернативой или дополнением к планировщику, но никак с ним не интегрированы. Они решают проблему «как защитить контейнер от побега из контейнера», но не решают проблему «как разместить контейнеры, чтобы минимизировать горизонтальное перемещение между ними».

Методы, реализующие обнаружение аномалий и атак, анализируют поведение подов и выявляют в нем аномалии, свидетельствующие о компрометации или проведении атаки. В данную группу можно отнести:

1. Мониторинг системных вызовов контейнеров на узле с помощью eBPF-зондов [19], при котором собираются многомерные данные (сетевые, файловые, системные вызовы) для обнаружения побегов из контейнера. Альтернативный подход [20] позволяет отслеживать процессы и контролировать доступ к файлам узла и доступным для записи и чтения слоям образов, используя модификацию ядерных функций.

2. Нейросетевые детекторы аномалий анализируют распределение системных вызовов во времени с помощью методов машинного обучения. Например, детектор на базе LSTM [21] фиксирует отклонение предсказанного распределения от реального и сигнализирует об атаке. Дополнительно нейросеть с вложением слов предсказывает следующий путь в файловой системе, что позволяет выявлять вредоносное программное обеспечение. Более сложный детектор ContainerGuard [22] использует ансамбль вариационных автокодировщиков (VAE) для многомерных временных рядов аппаратных и программных событий (кэш-промахи, страничные ошибки). VAE обучается только на нормальных образцах; аномалия определяется по низкой вероятности реконструкции, достигая AUC99,17 % при обнаружении атак.

3. Автоматное и реактивное обнаружение. Для интерпретируемости предложено

обучение вероятностных детерминированных конечных автоматов (PDFA) на данных, полученных из трафика [23]. Автомат вычисляет вероятность наблюдаемой последовательности; низкая вероятность указывает на аномалию, достигая сбалансированной точности 99,2 %. Реализация подобных методов представлена в инструменте `μDetector` [24], который автоматизирует сбор системных вызовов в Kubernetes и KubeEdge с помощью Sysdig-зондов и передает их во внешний модуль IDS. Эксперименты с нагрузкой свыше 100 тыс. системных вызовов показали, что потребление ресурсов рабочими узлами не превышает 10 %.

Методы фильтрации последовательностей в реальном времени могут противодействовать угрозам побега из контейнера, горизонтальному перемещению через узел и сетевому взаимодействию подов. Однако суть данных методов заключается в выявлении атак непосредственно в момент их исполнения, в отличие от методов безопасного расположения, которые минимизируют риски проведения для данных атак еще до развертывания приложения. Индивидуальная фильтрация системных вызовов не останавливает атаки, использующие только разрешенные вызовы. Например, в работе [25] представлена система NIMOS, которая из 106 эксплойтов привилегий Linux извлекает повторяющиеся паттерны последовательностей (*n*-граммы) длиной до 65. Показано, что 40 из 58 эксплойтов, не блокируемых стандартным `sessomp Docker`, могут быть нейтрализованы фильтрацией последовательностей длиной ≥ 8 .

Исходя из проведенного анализа существующих решений, составлен перечень требований к разрабатываемому новому методу защиты:

- охват всех каналов горизонтального перемещения атакующего, т.е. учет рисков, связанных с общим ядром (системные вызовы), расхождением RBAC-привилегий и сетевыми соединениями между подами;
- количественная оценка рисков – возможность вводить измеримую целевую функцию, позволяющую сравнивать различные варианты размещения;

- учет ресурсных ограничений, т.е. удовлетворение ограничений по CPU, памяти и другим ресурсам узлов;

- метод должен работать за полиномиальное время, приемлемое для использования в реальном планировщике Kubernetes;

- возможность настраивать приоритеты – обеспечение гибкой настройки весов между различными каналами риска и равномерностью загрузки;

- совместимость с существующими механизмами – метод не должен заменять политики безопасности или системы обнаружения, а дополнять их.

Перечисленные требования легли в основу разработанной математической модели риск-ориентированного размещения подов, которая учитывает три канала горизонтального перемещения атакующего.

3. МОДЕЛЬ РИСК-ОРИЕНТИРОВАННОГО ПЛАНИРОВАНИЯ ПОДОВ В ПЛАТФОРМЕ ОРКЕСТРАЦИИ KUBERNETES

Пусть заданы множества узлов $N = \{n_1, \dots, n_m\}$ с ресурсной емкостью Ω_j , m – количество узлов в кластере, подов $P = \{p_1, \dots, p_k\}$ с запросами ресурсов ω_i , k – количество подов в кластере.

Матрица размещения пода p_i на узел n_j , $X = [x_{ij}]$, x_{ij} определяется согласно формуле $x_{ij} = \begin{cases} 1, & \text{под } p_i \text{ назначен на узел } n_j \\ 0, & \text{иначе} \end{cases}$.

Размещение должно удовлетворять ограничениям:

- каждый под в единственном экземпляре расположен на единственном узле:

$$\sum_i x_{ij} = 1, \forall i;$$

- соответствие ресурсных ограничений узлов и запросов подов:

$$\sum_i x_{ij} \omega_i \leq \Omega_j, \forall j.$$

Сетевое взаимодействие между подами описывается как $A = [a_{ik}]$ – матрица

сетевой связности, a_{ik} определяется по формуле

$$a_{ik} = \begin{cases} 1, & \text{существует связь от } p_i \text{ к } p_j \\ 0, & \text{связи нет} \end{cases}$$

Пусть под p_i развернут на узле n_j и под p_k на узле n_l обмениваются трафиком $a_{ik} > 0$. Тогда риск горизонтального перемещения между узлами n_j, n_l рассчитывается по формуле

$$R_{net}(n_j, n_l) = \sum_{\substack{i: x_{ij}=1 \\ k: x_{kl}=1}} a_{ik}.$$

Обозначим S как глобальное множество всех системных вызовов, поддерживаемых ядром узлов кластера. Каждому вызову $s \in S$ сопоставим числовую оценку критичности $danger(s) \geq 0$, отражающую его потенциальную опасность для контейнерной изоляции, оценка проводится на основе анализа CVE, при эксплуатации которых используются данные системные вызовы [10].

Каждый под p_i характеризуется подмножеством системных вызовов $S_i \in S$, которые он может использовать (определяется статическим анализом образа или динамической трассировкой системных вызовов). Тогда объединенный профиль на узле n_j будет рассчитываться по формуле

$$S^j = \bigcup_i S_i^j,$$

где S_i^j – профиль системных вызовов пода p_i , расположенного на узле n_j .

Риск узла по каналу системных вызовов определяется как сумма критичностей всех вызовов, присутствующих на узле:

$$R_{sys}(n_j) = \sum_{s \in S^j} danger(s).$$

Каждому поду и узлу соответствует оценка ERP , рассчитанная по методике из работы [15].

Для оценки эффективности распределения подов с точки зрения эксплуатации ресурсов будет использована дисперсия η [12], рассчитанная по формуле

$$\eta = \frac{1}{|N|} \sum_j (u_j - \bar{u})^2, \quad u_j = \frac{1}{\Omega_j} \sum_i x_{ij} \omega_i, \quad \bar{u} = \frac{1}{|N|} \sum_j u_j.$$

Тогда целевая функция представляет собой линейную комбинацию рисков и дисбаланса при заданных ограничениях:

$$\min_X F(X) = \alpha \sum_j R_{sys}(n_j) + \beta ERP + \delta \sum_j \sum_l R_{net}(n_j, n_l) + \gamma \eta,$$

где веса $\alpha \beta \gamma \delta \geq 0$ задают приоритет между различными каналами атак и ресурсной равномерностью. Их выбор может осуществляться на основе экспертного мнения или методами многокритериальной оптимизации.

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Поскольку задача вычисления указанной целевой функции является сложной, содержит нелинейные комбинаторные зависимости и на практике труднорешаемая для кластеров среднего и большого размера, предлагается использовать жадный алгоритм последовательного назначения подов, основанный на ранжировании подов по критичности и выборе узла с минимальным приращением целевой функции. Такой подход можно применить в задаче планирования Kubernetes благодаря своей вычислительной эффективности и приемлемому качеству на практике.

Описание алгоритма. Используется жадная стратегия последовательного назначения подов с выбором узла, дающего минимальное приращение целевой функции:

1. Ранжирование подов по уровню критичности:

$$\text{criticality}(p_i) = c_1 \sum_k a_{ik} + c_2 \sum_{s \in S_i} \text{danger}(s) + c_3 (ERP_i),$$

где c_1, c_2, c_3 – веса, задающие значимость риска.

2. Процедура размещения.

2.1. Инициализация $x_{ij} = 0$; для каждого узла хранятся текущие объединенные профили вызовов S^j , индикаторы привилегий ERP^j и остаточная емкость Ω_j .

2.2. Для очередного пода p_i и каждого допустимого узла n_j вычисляется приращение ΔF .

2.3. Под назначается на узел, при котором ΔF минимально.

2.4. Обновляется состояние узла.

Веса опасности $\text{danger}(s)$ основаны на экспертных оценках. Жадный алгоритм не гарантирует нахождение глобального оптимума, хотя его оперативность делает его применимым в практических планировщиках.

В таблице сравниваются предложенные модели с известными работами.

Политики безопасности являются базовым уровнем защиты. Они проактивны, не требуют изменений планировщика и накладывают минимальные вычислительные затраты. Однако эти меры носят статический и декларативный характер, так как администратор задает общие правила, но не может автоматически оптимизировать совместное размещение подов на узлах. Кроме того, типовые политики не учитывают специфику системных вызовов каждого пода или тонкие различия в RBAC-привилегиях. Предложенная модель дополняет политики, позволяя динамически адаптировать размещение к конкретным профилям подов, сохраняя при этом возможность применения общих политик безопасности.

Существующие методы планирования размещения демонстрируют преимущества проактивного управления рисками. Однако каждый из них фокусируется лишь на одном-двух каналах горизонтального перемещения. Например, Sysched минимизирует риск через перемещения через узел, RBAClock – через расхождение RBAC-привилегий, Diktyo – через сетевую изоляцию. Однако ни один из указанных подходов не рассматривает все три канала совместно. Построенная модель впервые объединяет все три вектора атак в единой целевой функции, позволяя находить компромиссные решения, когда минимизация риска по одному каналу не увеличивает риск по-другому.

Альтернативные среды запуска контейнеров решают проблему изоляции на уровне узла, предотвращая побег из контейнера.

Сравнение характеристик подходов по обеспечению безопасности контейнеров и платформ оркестрации

Comparison of the characteristics of approaches to ensuring the security of containers and orchestration platforms

Характеристика	Политики безопасности	Планирование размещения	Альтернативная среда	Обнаружение аномалий	Предложенный метод
Учет горизонтального перемещения через системные вызовы	Частично (запрет опасных вызовов)	+	+	+	+
Учет латерального перемещения через RBAC	–	+	–	Частично (опосредованно)	+
Учет латерального перемещения через сеть	Частично (сетевые политики)	+	–	Частично (анализ сетевого поведения)	+
Одновременный учет всех трех каналов	–	–	–	–	+
Проактивность/реактивность	Проактивная	Проактивная	Проактивная	Реактивная	Проактивная

Они проактивны, не требуют модификации планировщика, но ценой существенного повышения накладных расходов (виртуализация, эмуляция системных вызовов). Предложенный подход может применяться совместно с любой средой запуска, дополнительно снижая риски за счет оптимизации размещения. Например, даже при использовании gVisor, размещение подов с сильно различающимися привилегиями на одном узле остается нежелательным, и наша модель это предотвращает.

Существующие системы обнаружения аномалий и атак работают реактивно – они выявляют аномальное поведение в процессе выполнения атаки. Такие решения не предотвращают начальную компрометацию и не снижают вероятность успеха атаки до ее начала. Проактивное планирование размещения, напротив, уменьшает поверхность атаки на этапе развертывания: атакующий получает меньше возможностей для латерального перемещения, даже если ему удалось скомпрометировать один под. Таким образом, оба подхода взаимно дополняют друг друга. Планировщик минимизирует ущерб, а система обнару-

жения перехватывает оставшиеся угрозы, работая в реальном времени.

Таким образом, предложенная модель занимает свое уникальное место среди существующих методов защиты, объединяя все три основных канала латерального перемещения (системные вызовы, RBAC-привилегии, сетевые взаимодействия) в рамках единой оптимизационной задачи размещения подов. Модель не заменяет, а дополняет политики безопасности и системы обнаружения, обеспечивая проактивное управление рисками на уровне оркестратора. Перспективным направлением дальнейших исследований является комбинирование предложенного планировщика с реактивными детекторами для построения комплексной системы безопасности Kubernetes.

5. ЗАКЛЮЧЕНИЕ

Предложена новая модель риск-ориентированного планирования размещения подов в контейнерных оркестраторах,

которая в отличие от существующих подходов учитывает все каналы латерального перемещения атакующего: через системные вызовы, расхождение RBAC-привилегий (метрика ERP) и сетевые взаимодействия между подами. Для каждого канала введены количественные оценки риска на уровне узла и кластера, объединенные в единую целевую функцию с возможностью задания приоритетов с помощью весовых коэффициентов. Разработан жадный алгоритм последовательного назначения подов, осно-

ванный на ранжировании по интегральной критичности и выборе узла с минимальным приращением целевой функции.

Проведенный сравнительный анализ с основными подходами показывает, что предложенная модель занимает уникальную нишу. Она является первой, объединяющей три канала латерального перемещения в единой оптимизационной задаче размещения, и может дополнять как политики безопасности, так и реактивные системы обнаружения.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Flexera Releases 2021 State of the Cloud Report. URL: <https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report> (дата обращения: 14.05.2026).
2. **Shringarputale S., McDaniel P., Butler K., La Porta T.** Co-residency Attacks on Containers are Real // CCSW 2020 – Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop. 2020. P. 53–66. DOI: 10.1145/3411495.3421357.
3. **Xing Gao, Zhongshu Gu, Kayaalp M. et al.** ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds // Proceedings – 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, 26–29 June 2017, Denver, CO, USA. 2017. P. 237–248.
4. **Nanzi Yang, Wenbo Shen, Jinku Li et al.** Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications // CCS 2023 – Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023. Vol. 1. P. 3048–3062. DOI: 10.1145/3576915.3623121.
5. **El Kafhali S.** A Survey of Adaptive Scheduling Techniques, Goals, and Challenges in Kubernetes // Archives of Computational Methods in Engineering. 2026. Vol. 33. P. 6047–6070.
6. **Bagnato R., Notarianni L., Sabatini A., Vol- lero L.** Security Challenges and Solutions in Containerized Environments: A Comprehensive Review // Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR). 2025. P. 720–724.
7. **Senjab K., Abbas S., Ahmed N. et al.** A survey of Kubernetes scheduling algorithms // Journal of Cloud Computing. 2023. Vol. 12. № 1. P. 87. DOI: 10.1186/s13677-023-00471-1.
8. **Ahmad I., AlFailakawi M. Gh., AlMuta- wa A., Alsalman L.** Container scheduling techniques: A Survey and assessment // Journal of King Saud University – Computer and Information Sciences. 2022. Vol. 34. № 7. P. 3934–3947.
9. US11921885B2 USA. Security risk-aware scheduling on container-based clouds / Le M. V., Md S. Ahmed, H. T. Jamjoom. 2021.
10. **Le M. V., Ahmed S., Williams D., Jam- joom H.** Securing Container-based Clouds with Syscall-aware Scheduling // Proceedings of the ACM Conference on Computer and Communications Security, 10–14 July 2023, Melbourne, VIC, Australia. 2023. P. 812–826. DOI: 10.1145/3579856.3582835.
11. **Zhou D., Chen H., Cheng G.** A Security Containers Placement Algorithm Based on

- DQN for Microservices to Defend Against Co-Resident Threat // 2023 8th International Conference on Computer and Communication Systems (ICCCS), 21–23 April 2023, Guangzhou, China. 2023. P. 683–688.
12. **Li M., Hu H., Liu W.** A Secure Container Placement Algorithm Based on Microservice Invocation Criticality // 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT), 26–28 April 2024, Jilin, China. 2024. P. 234–240.
 13. **Santos J., Truyen E., Baumann C. et al.** Towards Intent-Based Scheduling for Performance and Security in Edge-to-Cloud Networks // 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), 11–14 March 2024, Paris, France. 2024. P. 222–227.
 14. **Boutih K., Djelouah M., Agoulmine N.** Next Step Towards Intent-based Scheduling for Performance and Security in Edge-to-Cloud Networks // 12th International Workshop on ADVANCES in ICT Infrastructures and Services (ADVANCE 2025), June 2025, Nice, France. 2025. P. 98–102. DOI: 10.48545/advance2025-shortpapers-4.5.
 15. **Qingwang Chen, Ru Tan, Xinyu Liu et al.** RBAClock: Contain RBAC Permissions through Secure Scheduling // 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 19–22 October 2025, Gold Coast, Australia. 2025. P. 950–965. DOI: 10.1109/RAID67961.2025.00072.
 16. **Zhichao Hua, Yang Yu, Jinyu Gu et al.** TZ-Container: protecting container from untrusted OS with ARM TrustZone // Science China Information Sciences. 2021. Vol. 64. № 9. DOI: 10.1007/s11432-019-2707-6.
 17. **Findlay W., Somayaji A., Barrera D.** Bpf-box: Simple Precise Process Confinement with eBPF // Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW 2020). 2020. P. 91–103.
 18. **Wruck F., Sarafov V., Jakobsmeier F., Weiß M.** GyroidOS: Packaging Linux with a Minimal Surface // Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS 2022). 2022. P. 87–96.
 19. **Tao Zheng, Yunxiang Qiu, Yundan Zheng et al.** Enhancing TinyML-Based Container Escape Detectors with Systemcall Semantic Association in UAVs Networks // IEEE Internet of Things Journal. 2024. Vol. 11. № 12. P. 21158–21169.
 20. **Kehe Wu, Dan Yang, Xue Gao et al.** Process based container escape monitoring and resource isolation scheme // 2022 13th International Conference on Information and Communication Systems (ICICS), 21–23 June 2022, Irbid, Jordan. 2022. P. 54–58.
 21. **Gantikow H., Zohner T., Reich C.** Container anomaly detection using neural networks analyzing system calls // 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 11–13 March 2020, Västerås, Sweden. 2020. P. 408–412.
 22. **Wang Yu., Wang Q., Chen X. et al.** ContainerGuard: A Real-Time Attack Detection System in Container-Based Big Data Platform // IEEE Transactions on Industrial Informatics. 2022. Vol. 18. № 5. P. 3327–3336.
 23. **Cao C., Blaise A., Verwer S., Rebecchi F.** Learning State Machines to Monitor and Detect Anomalies on a Kubernetes Cluster // ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security. 2022. Art. 117. P. 1–9.
 24. **Flora J., Teixeira M., Antunes N.** µDetector: Automated Intrusion Detection for Microservices // 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023. P. 748–752.
 25. **Song S., Suneja S., Le M. V., Tak B.** On the Value of Sequence-Based System Call Filtering for Container Security // IEEE International Conference on Cloud Computing. 2023. P. 296–307.

REFERENCES

1. Flexera Releases 2021 State of the Cloud Report. URL: <https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report> (accessed: 14.05.2026).
2. **Shringarputale S., McDaniel P., Butler K., La Porta T.** Co-residency Attacks on Containers are Real. CCSW 2020 – Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop. 2020, pp. 53–66. DOI: 10.1145/3411495.3421357.
3. **Xing Gao, Zhongshu Gu, Kayaalp M. et al.** ContainerLeaks: Emerging Security Threats

- of Information Leakages in Container Clouds. Proceedings – 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, 26–29 June 2017, Denver, CO, USA. 2017, pp. 237–248.
4. **Nanzi Yang, Wenbo Shen, Jinku Li et al.** Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications. CCS 2023 – Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023. Vol. 1, pp. 3048–3062. DOI: 10.1145/3576915.3623121.
 5. **El Kafhali S.** A Survey of Adaptive Scheduling Techniques, Goals, and Challenges in Kubernetes. *Archives of Computational Methods in Engineering*. 2026. Vol. 33. P. 6047–6070.
 6. **Bagnato R., Notarianni L., Sabatini A., Volero L.** Security Challenges and Solutions in Containerized Environments: A Comprehensive Review. Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR). 2025, pp. 720–724.
 7. **Senjab K., Abbas S., Ahmed N. et al.** A survey of Kubernetes scheduling algorithms. *Journal of Cloud Computing*. 2023. Vol. 12. No. 1, pp. 87. DOI: 10.1186/s13677-023-00471-1.
 8. **Ahmad I., AlFailakawi M. Gh., AlMutawa A., Alsalman L.** Container scheduling techniques: A Survey and assessment. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34. No. 7, pp. 3934–3947.
 9. **Le M. V., Ahmed Md S., Jamjoom H. T.** US11921885B2 USA. Security risk-aware scheduling on container-based clouds. 2021.
 10. **Le M. V., Ahmed S., Williams D., Jamjoom H.** Securing Container-based Clouds with Syscall-aware Scheduling. Proceedings of the ACM Conference on Computer and Communications Security, 10–14 July 2023, Melbourne, VIC, Australia. 2023, pp. 812–826. DOI: 10.1145/3579856.3582835.
 11. **Zhou D., Chen H., Cheng G.** A Security Containers Placement Algorithm Based on DQN for Microservices to Defend Against Co-Resident Threat. 2023 8th International Conference on Computer and Communication Systems (ICCCS), 21–23 April 2023, Guangzhou, China. 2023, pp. 683–688.
 12. **Li M., Hu H., Liu W.** A Secure Container Placement Algorithm Based on Microservice Invocation Criticality. 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT), 26–28 April 2024, Jilin, China. 2024, pp. 234–240.
 13. **Santos J., Truyen E., Baumann C. et al.** Towards Intent-Based Scheduling for Performance and Security in Edge-to-Cloud Networks. 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), 11–14 March 2024, Paris, France. 2024, pp. 222–227.
 14. **Boutih K., Djelouah M., Agoulmine N.** Next Step Towards Intent-based Scheduling for Performance and Security in Edge-to-Cloud Networks. 12th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2025), June 2025, Nice, France. 2025, pp. 98–102. DOI: 10.48545/advance2025-shortpapers-4_5.
 15. **Qingwang Chen, Ru Tan, Xinyu Liu et al.** RBAClock: Contain RBAC Permissions through Secure Scheduling. 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 19–22 October 2025, Gold Coast, Australia. 2025, pp. 950–965. DOI: 10.1109/RAID67961.2025.00072.
 16. **Zhichao Hua, Yang Yu, Jinyu Gu et al.** TZ-Container: protecting container from untrusted OS with ARM TrustZone. *Science China Information Sciences*. 2021. Vol. 64. No. 9. DOI: 10.1007/s11432-019-2707-6.
 17. **Findlay W., Somayaji A., Barrera D.** Bpfbbox: Simple Precise Process Confinement with eBPF. Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW 2020). 2020, pp. 91–103.
 18. **Wruck F., Sarafov V., Jakobsmeier F., Weiß M.** GyroidOS: Packaging Linux with a Minimal Surface. Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS 2022). 2022, pp. 87–96.
 19. **Tao Zheng, Yunxiang Qiu, Yundan Zheng et al.** Enhancing TinyML-Based Container Escape Detectors with Systemcall Semantic Association in UAVs Networks. *IEEE Internet of Things Journal*. 2024. Vol. 11. No. 12, pp. 21158–21169.
 20. **Kehe Wu, Dan Yang, Xue Gao et al.** Process based container escape monitoring and resource isolation scheme. 2022 13th International Conference on Information and Communication Systems (ICICS), 21–23 June 2022, Irbid, Jordan. 2022, pp. 54–58.

21. **Gantikow H., Zohner T., Reich C.** Container anomaly detection using neural networks analyzing system calls. 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 11–13 March 2020, Västerås, Sweden. 2020, pp. 408–412.
22. **Wang Yu., Wang Q., Chen X. et al.** ContainerGuard: A Real-Time Attack Detection System in Container-Based Big Data Platform. *IEEE Transactions on Industrial Informatics*. 2022. Vol. 18. No. 5, pp. 3327–3336.
23. **Cao C., Blaise A., Verwer S., Rebecchi F.** Learning State Machines to Monitor and Detect Anomalies on a Kubernetes Cluster. ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security. 2022. Art. 117, pp. 1–9.
24. **Flora J., Teixeira M., Antunes N.** μ Detector: Automated Intrusion Detection for Microservices. 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023, pp. 748–752.
25. **Song S., Suneja S., Le M. V., Tak B.** On the Value of Sequence-Based System Call Filtering for Container Security. IEEE International Conference on Cloud Computing. 2023, pp. 296–307.

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

ВЕДЕНСКИЙ Виктор Алексеевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: sci@ibks.spbstu.ru
ORCID: 0009-0003-4569-2258

VEDENSKII Viktor A. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29