ТЕХНИЧЕСКИЕ НАУКИ

УДК 004.056.53

Моделирование угроз и разработка политик безопасности для контейнеризированных приложений

Я.Н. Тисейко, О.А. Сафарьян
Донской государственный технический университет, г. Ростов–на–Дону, Российская Федерация

Аннотация
Проблема обеспечения информационной безопасности микросервисных архитектур обусловлена неэффективностью классической защиты сетевого периметра против специфических уязвимостей контейнеризированных сред. Цель статьи — проанализировать механику современных кибератак и разработать комплексную архитектуру эшелонированной защиты контейнеризированных приложений. Объектом исследования выступают локальные инфраструктуры, функционирующие под управлением системы Docker. Последовательность работы включает декомпозицию рисков по методологии Process for Attack Simulation and Threat Analysis (PASTA), детальный анализ векторов компрометации, а также внедрение механизмов нулевого доверия и методов ядерного поведенческого мониторинга на базе технологии Modern extended Berkeley Packet Filter (eBPF). В результате доказано, что размещение логики мониторинга в ядре операционной системы Linux совместно с асинхронным контуром реагирования позволяет прерывать цепочку атаки за время менее 20 мс. Практическая значимость работы заключается в формировании научно обоснованных рекомендаций по составлению декларативных правил безопасности для надёжной изоляции рабочих нагрузок и предотвращения побега из контейнера.

Ключевые слова: кибербезопасность контейнеров, микросервисы, архитектура нулевого доверия, eBPF-мониторинг, моделирование угроз, побег из контейнера

Для цитирования. Тисейко Я.Н., Сафарьян О.А. Моделирование угроз и разработка политик безопасности для контейнеризированных приложений. *Молодой исследователь Дона*. 2026;11(4):69–74.

Threat Modeling and Security Policy Development for Containerized Applications

Yaroslav N. Tiseiko Y, Olga A. Safaryan
Don State Technical University, Rostov-on-Don, Russian Federation.

Abstract
The problem of ensuring information security in microservice architectures stems from the inefficiency of conventional network perimeter defense in protecting specific vulnerabilities inherent to the containerized environments. The aim of the article is to analyse the mechanisms of modern cyber attacks and develop a comprehensive Defence-in-Depth architecture for the containerized applications. The objects of the study were local development environments run inside Docker container. The study plan included risk decomposition according to the Process for Attack Simulation and Threat Analysis (PASTA) methodology, detailed analysis of compromise vectors, as well as implementation of the zero trust mechanisms and kernel-level behavioral monitoring methods based on Extended Berkeley Packet Filter (eBPF) technology. As a result, it has been proved that placing monitoring logic in the Linux operating system kernel in conjunction with the asynchronous response loop allows aborting the attack chain in less than 20 ms. The practical significance of the study lies in the formulation of scientifically-based recommendations for drawing up declarative security policies for robust isolation of workloads and container escape prevention.

Keywords: container security, microservices, zero-trust architecture, eBPF monitoring, threat modeling, container escape

For Citation. Tiseiko YaN, Safaryan OA. Threat Modeling and Security Policy Development for Containerised Applications. *Young Researcher of Don*. 2026;11(4):69–74.

© Тисейко Я.Н., Сафарьян О.А., 2026

Введение. Современная индустрия программного обеспечения неразрывно связана с технологиями контейнеризации, обеспечивающими высокую скорость развёртывания и оптимальное использование вычислительных ресурсов. Однако переход к контейнерам породил фундаментальную проблему: в отличие от аппаратной виртуализации, контейнеры совместно используют единое ядро хостовой операционной системы Linux [1]. Данная архитектурная особенность делает классические методы защиты периметра сети критически уязвимыми перед внутренними угрозами. Актуальность обусловлена тем, что существующие подходы к безопасности зачастую не поспевают за развитием векторов атак. В предыдущих исследованиях, посвящённых защите контейнеров, основной упор делался либо на статическое сканирование образов до их запуска [2], либо на изоляцию на уровне сетевых портов [3]. Нерешённой оставалась проблема выявления динамических аномалий в реальном времени и противодействия атакам на цепочки поставок, которые обходят статические анализаторы. Цель статьи — провести комплексный анализ механики возникновения уязвимостей в локальных контейнерных средах, разработать и оценить эффективность многоуровневой архитектуры защиты на базе поведенческого анализа. Для достижения цели применялась методика структурного риск-моделирования PASTA, технология расширенного фильтра пакетов Беркли для перехвата системных вызовов и концепция управления политиками безопасности как кодом. Краткие результаты исследования подтверждают выдвинутую гипотезу: предложенный подход позволяет блокировать попытки несанкционированного доступа к системным директориям (побег из контейнера) без создания узких мест в производительности. Вклад данного исследования в науку состоит в эмпирическом обосновании эффективности eBPF-фильтрации, а практическая ценность заключается в возможности быстрого развёртывания предложенного лабораторного стенда для аудита безопасности существующих систем.

Основная часть. Объектом исследования в данной работе выступает изолированная программная среда на базе технологии локальной контейнеризации Docker.

Исследование выполнялось в следующей последовательности:

– организация лабораторного стенда с уязвимыми базовыми образами;
– декомпозиция архитектуры приложения и моделирование угроз;
– анализ механики ключевых векторов атак;
– разработка набора декларативных правил для eBPF-сенсоров;
– сравнительный анализ эффективности системы до и после применения эшелонированной обороны.

Выбор методов исследования строго обоснован спецификой объекта. Для выявления уязвимостей была выбрана методология PASTA. В отличие от популярного фреймворка STRIDE [4], ориентированного исключительно на технические компоненты, PASTA выстраивает связь между критическими рисками и механизмами защиты. Для динамического мониторинга выбрана технология eBPF, поскольку она позволяет безопасно выполнять программы-зонды внутри ядра Linux, перехватывая любые действия контейнера на уровне пространства ядра [5]. Кроме того, данная технология эффективно детектирует тактики вертикального и горизонтального перемещения по скомпрометированному периметру [6].

Опираясь на стандарт NIST SP 800-190 [7], а также аналитику матрицы тактик киберпреступников MITRE ATT&CK для контейнерных сред [8], в ходе моделирования были выделены и детально проанализированы три ключевых вектора компрометации. Согласно матрице MITRE ATT&CK и отчётам исследователей «Лаборатории Касперского» [9], механика атаки на цепочку поставок базируется на внедрении вредоносного кода в базовые образы или транзитивные зависимости ещё на этапе сборки. Злоумышленник использует тайпосквоттинг или взламывает репозитории сторонних разработчиков на публичных площадках. В результате контейнер поступает на сервер уже заражённым, успешно обходя классические сетевые экраны. Тактика горизонтального перемещения описывает развитие атаки внутри скомпрометированного периметра. Получив первоначальный доступ к наименее защищённому контейнеру, атакующий сканирует внутреннюю виртуальную сеть Docker. Целью в данном случае является нахождение смежных микросервисов со слабыми паролями либо перехват незашифрованного трафика между узлами с использованием доверительных отношений внутри кластера. Наиболее критическим вектором является побег из контейнера, детально задокументированный в Национальной базе данных уязвимостей США. Показательным примером служит критическая уязвимость CVE-2024-21626 с оценкой по шкале CVSS, равной 8.6 баллов, в популярной среде выполнения runC [10]. Эксплуатируя логические изъяны обработки файловых дескрипторов, вредоносный процесс обманывает механизмы монтирования файловых систем. Это позволяет ему выйти за рамки пространств имён и получить права записи в системные директории хоста, что ведёт к полному перехвату управления физическим сервером.

Учитывая многообразие и постоянную эволюцию описанных векторов атак, традиционная защита на уровне сетевых экранов и статического анализа образов оказывается недостаточной. В связи с этим целью практической части данного исследования стала реализация механизма Active Runtime Security. Экспериментальный стенд спроектирован не для имитации конкретных CVE, а для перехвата универсальных поведенческих паттернов злоумышленника, которые неизбежно проявляются на финальных этапах реализации вышеописанных угроз.

Стенд развёрнут на базе операционной системы Fedora 42 с версией ядра Linux 6.14, в которой изначально присутствует поддержка технологии Modern eBPF и среды контейнеризации Docker Engine. В качестве сенсора и анализатора событий используется система поведенческого мониторинга Falco, сконфигурированная для работы с современным eBPF-драйвером. Данный инструмент установлен непосредственно на целевом хосте, что гарантирует независимый и защищённый перехват системных вызовов всех функционирующих контейнеров в режиме реального времени, исключая возможность вмешательства со стороны скомпрометированных изолированных сред.

Рассмотрим созданные правила для детектирования атак, направленных на контейнеризированные приложения (рис. 1).

```
1  - rule: Kill Container Shell
2    desc: Мгновенная блокировка любого шелла в контейнере
3    condition: container.id != host and proc.name = bash and evt.type = execve
4    output: "CRITICAL: shell detected container_id=%container.id"
5    priority: CRITICAL
6
7  - rule: Kill Network Utilities
8    desc: Блокировка запуска сетевых утилит в контейнере
9    condition: container.id != host and spawned_process and proc.name in (curl, wget, nmap, nc)
10   output: "CRITICAL: Network utility spawned! container_id=%container.id"
11   priority: CRITICAL
12
13 - rule: Kill Sensitive File Read
14   desc: Блокировка попытки чтения файла shadow
15   condition: open_read and container.id != host and fd.name = /etc/shadow
16   output: "CRITICAL: Sensitive file read! container_id=%container.id"
17   priority: CRITICAL
18
19 - rule: Kill System Dir Write
20   desc: Блокировка записи в системные папки
21   condition: >
22     open_write and container.id != host
23     and fd.directory in (/bin, /sbin, /usr/bin, /usr/sbin)
24   output: "CRITICAL: System directory modified! container_id=%container.id"
25   priority: CRITICAL
```

Рис. 1. Созданные правила

С целью комплексной защиты среды исполнения разработан набор из четырёх политик поведенческого анализа, охватывающих основные векторы атак на контейнерную инфраструктуру. Первое правило предотвращает получение интерактивного доступа, строго блокируя системные вызовы, ведущие к запуску командной оболочки bash внутри изолированной среды. Второе правило пресекает несанкционированную сетевую активность путём запрета исполнения утилит curl, wget, nmap и nc. Третье правило защищает систему от кражи учётных данных, отслеживая и прерывая любые попытки чтения критичного файла /etc/shadow. Четвёртое правило исключает возможность закрепления злоумышленника, запрещая операции модификации и записи в системные критичные директории. При нарушении любого из этих условий сенсор генерирует событие наивысшего приоритета с обязательной передачей идентификатора скомпрометированного контейнера, что инициирует его немедленную автоматическую ликвидацию.

В ходе эксперимента стенд подвергся четырём типам воздействий, имитирующим действия злоумышленника после успешного пробития внешнего периметра.

В случае атаки интерактивного доступа целью злоумышленника является получение оболочки командной строки для разведки системы, выполнения скриптов и закрепления в контейнере. Для детектирования eBPF-сенсор перехватывает системный вызов execve. Движок Falco сопоставляет параметры вызова: процесс bash запущен внутри пространства имён контейнера, а не на хосте. В результате скрипт-респондер получает идентификатор контейнера и выполняет docker kill. Задержка составляет менее 20 мс, поэтому злоумышленник не успевает ввести ни одной команды после появления приглашения, и сессия обрывается.

При атаке, направленной на загрузку вредоносного программного обеспечения (ВПО), целью злоумышленника выступает установка ВПО либо организация канала связи с сервером Command & Control. Детектирование реализуется путём предотвращения инициализации исходящего соединения из контейнера, в котором развёрнут Nginx или иной базовый веб-сервис, с помощью утилит curl либо wget. eBPF-сенсор перехватывает запуск указанных бинарных файлов через execve. Блокировка срабатывает до того, как утилита успевает выполнить разрешение DNS-имени и установить TCP-рукопожатие, что позволяет предотвратить загрузку файла за доли секунды.

При атаке, нацеленной на кражу учётных данных, целью нарушителя является извлечение хешей паролей пользователей для их последующего взлома и повышения привилегий. Детектирование осуществляется путём перехвата системных вызовов openat и read. Правило Falco настроено на мониторинг любых попыток доступа к критичным файлам со стороны процессов, не относящихся к доверенным системным демонам. В результате злоумышленник успевает увидеть содержимое файла на экране, поскольку выполнение cat происходит быстрее, чем цикл реагирования. Однако контейнер немедленно уничтожается, и нарушитель лишается возможности

использовать полученные данные для развития атаки в пределах данного узла. Подобный эффект является допустимым для асинхронных систем предотвращения вторжений.

При атаке, направленной на закрепление в системе, целью злоумышленника выступает модификация системных файлов или размещение ВПО в директориях, входящих в переменную $PATH, для обеспечения автоматического запуска. Детектирование реализуется путём перехвата вызовов openat с флагами записи в защищаемые системные директории — /bin, /sbin, /usr/bin. Попытка создания или изменения файла прерывается средствами изоляции контейнера; файловая система хоста при этом остаётся в безопасности.

Рассмотрим результат работы правил (рис. 2).

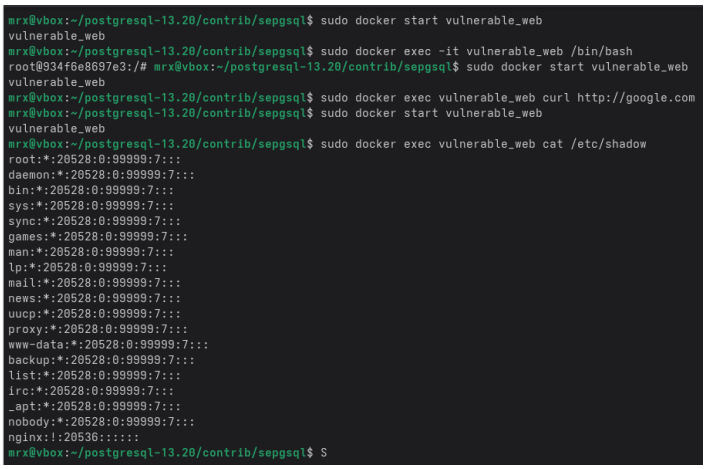

Рис. 2. Результат работы правил

Результаты испытаний подтвердили высокую эффективность внедрённого eBPF-контура активной защиты при обработке целевых векторов атак. При попытке злоумышленника получить интерактивный доступ к оболочке контейнера система мгновенно перехватила системный вызов и ликвидировала среду исполнения, не позволив атакующему ввести ни одной команды. Аналогичный механизм пресечения сработал при несанкционированном запуске сетевой утилиты curl, где вредоносный процесс был уничтожен до момента установки сетевого соединения. Особый научный интерес представляет реакция системы на попытку кражи учётных данных через утилиту cat. В данном сценарии зафиксирована классическая ситуация гонки для асинхронных систем безопасности: атомарная операция вывода сработала быстрее цикла реагирования, что позволило однократно прочитать целевой файл /etc/shadow.

В таблице 1 представлены результаты экспериментальной проверки системы активной защиты контейнеров.

Таблица 1

Сравнительный анализ эффективности методов защиты контейнерных сред

Вектор атаки	Имитируемое действие	Триггер детекции	Реакция системы	Результат и время отклика
Интерактивный доступ	Запуск оболочки в контейнере	Процесс bash вне хоста	Генерация алерта, передача ID в скрипт изоляции	Успешно. Контейнер прерван до вывода приглашения ввода (< 20 мс)
Сетевая разведка/загрузка ВПО	Использование утилит передачи данных	Процесс curl/ nmap	Инициализация команды docker kill целевого узла	Успешно. Соединение прервано на этапе разрешения DNS - имени
Кража учетных данных	Чтение критичных системных файлов	Чтение несистемным процессом	Фиксация аномалии, инициирование уничтожения среды	Изоляция контейнера (2 мс)
Закрепление в системе	Запись в системные директории	Системный вызов openat с флагами записи в защищаемый $PATH	Блокировка файловой операции и ликвидация контейнера	Успешно. Файловая система не скомпрометирова (2 мс)

Реализация стенда на базе eBPF-инструментария продемонстрировала высокую инженерную гибкость. Был разработан комплекс правил в формате YAML, описывающих недопустимое поведение. Экспериментальные результаты подтвердили, что данный подход детерминированно пресекает попытки эксплуатации известных уязвимостей сред выполнения на этапе обращения к ядру.

Заключение. Подводя итог проведённому исследованию, можно утверждать, что высокая скорость развёртывания локальных микросервисов неизбежно сопровождается критическим расширением поверхности атаки. Разделяемое ядро Linux представляет собой главное слабое звено контейнеризации. В ходе работы доказано, что реактивные базовые настройки Docker не способны предотвращать современные угрозы.

Полученные результаты полностью согласуются с целью, обозначенной во введении. Автором успешно проанализированы механизмы актуальных кибератак и разработана комплексная модель защиты, легко реализуемая на практике. Обобщая изложенное, можно сделать вывод: обеспечение отказоустойчивости требует обязательного внедрения эшелонированной обороны. Практические рекомендации сводятся к отказу от привилегий root внутри контейнеров, внедрению строгой верификации образов и развёртыванию систем eBPF-мониторинга со специализированными наборами правил.

Результаты исследования подтверждают выдвинутую гипотезу: предложенный подход позволяет эффективно блокировать попытки несанкционированного доступа к системным директориям без создания узких мест в производительности.

Главным направлением дальнейших исследований в данной области выступает автоматизация процесса генерации eBPF-политик на основе профилирования штатного поведения микросервисов на этапе тестирования.

Прогнозируя развитие ситуации в ближайшие 3–5 лет, следует ожидать смещения фокуса киберпреступников в сторону сложных логических атак против сборочных конвейеров. В этих условиях технологии глубокого eBPF-мониторинга в сочетании с подходом Policy-as-Code станут обязательным индустриальным стандартом защиты любых контейнеризированных систем, вытесняя устаревшие механизмы изоляции.

Список литературы

1. Ищанова С.Г. Обзор и экономическое обоснование методов защиты Docker контейнеров от Escape-атак. Вызовы современности и стратегии развития общества в условиях новой реальности. В: *Труды XXXVI Международной научно-практической конференции. Москва, 2025*. 268–272 с.

2. Гальцев П.Н. Обзор уязвимостей и методов защиты контейнеров. *Научный аспект.* 2024;48(4):6374–6377. URL: https://na-journal.ru/4-2024-informacionnye-tekhnologii/10957-obzor-uyazvimostei-i-metodov-zashchity-konteinerov (дата обращения: 29.03.2026).

3. Фомин Ю.С., Алтынбаев А.Ф., Тарануха А.В. Целостность, доступность, конфиденциальность виртуализации и контейнеризации. *Парадигма*. 2025;(5-5):173–182.

4. Курсков Е.М. Защита программного обеспечения от несанкционированного использования. В: Труды *Международной научно-практической конференции по компьютерной и информационной безопасности (INFSEC 2023). Екатеринбург,* 2023. С. 51–56.

5. Мукминов А.Ф., Порсев И.С. Построение эшелонированной защиты с помощью технологии контейнеризации средств сетевой безопасности. В: Труды конференции «*Радиоэлектронные устройства и системы для инфокоммуникационных технологий». Москва,* 2024. С. 394–398.

6. Селиверстов В.В., Корчагин С.А. Требования к защите приложений, основанных на технологии контейнеризации. В: *Труды II Всероссийской научно-практической конференции «Теория и практика обеспечения информационной безопасности». Москва*; 2022. 96–104 с.

7. Souppaya M, Morello J, Scarfone K. *Application Container Security Guide. NIST Special Publication 800-190.* Gaithersburg: National Institute of Standards and Technology; 2017. 61 p. URL: https://www.hsdl.org/c/abstract/?docid=805077 (дата обращения: 29.03.2026).

8. *MITRE ATT&CK. Matrix for Enterprise. Containers Matrix.* URL: https://attack.mitre.org/matrices/enterprise/containers (дата обращения: 29.03.2026).

9. *Ландшафт киберугроз: атаки на цепочки поставок и компрометация open-source репозиториев. Securelist by Kaspersky.* URL: https://securelist.ru (дата обращения: 29.03.2026).

10. *CVE-2024-21626 Detail. National Vulnerability Database (NVD).* URL: https://nvd.nist.gov/vuln/detail/CVE-2024-21626 (дата обращения: 29.03.2026).

Об авторах:

Ярослав Николаевич Тисейко, студент кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), yaroslav.tiseiko@yandex.ru

Ольга Александровна Сафарьян, кандидат технических наук, доцент, заведующий кафедры «Кибербезопасность информационных систем» Донского государственного технического университета (344003, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1), osafaryan@donstu.ru

Конфликт интересов: **авторы заявляют об отсутствии конфликта интересов.**

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the Authors:

Yaroslav N. Tiseiko, Student of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), yaroslav.tiseiko@yandex.tu

Olga A. Safaryan, Cand.Sci. (Engineering), Associate Professor, Head of the Cybersecurity of Information Systems Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, Russian Federation), osafaryan@donstu.ru

Conflict of Interest Statement: **the authors declare no conflict of interest.**

All authors have read and approved the final manuscript.