

УДК: 004.415.5:004.056

*Бакайкина Виктория Геннадиевна,  
ассистент кафедры Программной инженерии  
ФГБОУ ВО «Поволжский государственный университет  
телекоммуникаций и информатики»  
Некрасов Михаил Владимирович,  
студент,  
ФГБОУ ВО «Поволжский государственный университет  
телекоммуникаций и информатики»*

**МЕТОДЫ ИЗОЛЯЦИИ РАБОЧИХ НАГРУЗОК В СОВРЕМЕННЫХ  
ОПЕРАЦИОННЫХ СИСТЕМАХ НА ОСНОВЕ АППАРАТНОЙ  
ВИРТУАЛИЗАЦИИ: СРАВНИТЕЛЬНЫЙ АНАЛИЗ EBPF, GVISOR И  
МИКРОЯДЕРНЫХ АРХИТЕКТУР**

*Аннотация:* Проблема эффективной изоляции рабочих нагрузок является ключевой для безопасности современных облачных и контейнерных сред. Традиционные механизмы операционных систем на основе монолитного ядра зачастую не обеспечивают достаточный уровень защиты от потенциальных атак на гипервизор или хостовую систему. В данной статье проводится сравнительный анализ трех современных архитектурных подходов к усилению изоляции: расширенного фильтра Беркли (eBPF), платформы пользовательского ядра gVisor и микроядерной парадигмы на примере seL4. Цель работы: оценить эффективность каждого метода по совокупности критериев, включая силу изоляции, производительность, совместимость и операционную сложность. Методология основана на теоретико-архитектурном анализе и синтезе данных из актуальных исследований. Результатом работы является четкое разграничение областей применения рассматриваемых технологий, что позволяет

*принимать обоснованные инженерные решения при проектировании безопасных систем. Наиболее универсальным решением для массового облачного развертывания демонстрирует себя подход с пользовательским ядром, в то время как eBPF идеален для тонкой настройки безопасности внутри доверенного периметра, а микроядра для критически важных систем с верифицируемыми гарантиями.*

***Ключевые слова:** операционные системы, изоляция, безопасность, eBPF, gVisor, микроядро, контейнеры, виртуализация.*

***Bakaikina Victoria Gennadievna,**  
**assistant Professor of Software Engineering**  
**Volga Region State University of Telecommunications and Informatics***

***Nekrasov Mikhail Vladimirovich,**  
**student,**  
**Volga Region State University of Telecommunications and Informatics***

## **WORKLOAD ISOLATION METHODS IN MODERN OPERATING SYSTEMS BASED ON HARDWARE VIRTUALIZATION: COMPARATIVE ANALYSIS OF EBPF, GVISOR, AND MICRONUCLEAR ARCHITECTURES**

***Abstract:** The problem of effective isolation of workloads is key to the security of modern cloud and container environments. Traditional monolithic kernel-based operating system mechanisms often do not provide sufficient protection against potential attacks on the hypervisor or host system. This article provides a comparative analysis of three modern architectural approaches to enhanced isolation: the extended Berkeley filter (eBPF), the gVisor user core platform, and the micronucleus paradigm using seL4 as an example. The purpose of the work is to evaluate the effectiveness of each method based on a set of*

*criteria, including isolation strength, performance, compatibility, and operational complexity. The methodology is based on theoretical and architectural analysis and synthesis of data from current research. The result of the work is a clear delineation of the areas of application of the technologies under consideration, which allows us to make informed engineering decisions when designing secure systems. The most versatile solution for mass cloud deployment is the custom core approach, while eBPF is ideal for fine-tuning security within a trusted perimeter, and microkernels for mission-critical systems with verifiable guarantees.*

**Keywords:** *operating systems, isolation, security, eBPF, gVisor, microkernel, containers, virtualization.*

## **Введение**

Эволюция облачных вычислений и повсеместное внедрение контейнерных технологий коренным образом изменили ландшафт развертывания приложений. Однако эта парадигма принесла с собой новые вызовы в области безопасности, в частности, проблему обеспечения надежной изоляции между множественными рабочими нагрузками, выполняющимися на одном физическом хосте [1]. Классические механизмы изоляции на основе пространств имен (namespaces) и групп управления (cgroups) в монолитном ядре Linux, будучи эффективными для управления ресурсами, не являются достаточным барьером в случае компрометации ядра или эксплуатации уязвимости типа «побег из контейнера». В качестве ответа на эти вызовы индустрией были предложены и получили развитие несколько альтернативных архитектурных подходов, стремящихся усилить границы изоляции. Настоящая статья фокусируется на трех наиболее релевантных и современных из них: использование расширенного фильтра Беркли (eBPF) для программируемой безопасности ядра, концепция пользовательского ядра, реализованная в проекте gVisor от Google, и микроядерная архитектура, представленная верифицируемым микроядром seL4. Целью исследования

является проведение систематического сравнительного анализа данных технологий для выявления их сильных и слабых сторон, а также определения оптимальных сценариев применения. Актуальность темы обусловлена растущими требованиями к безопасности облачной инфраструктуры и необходимостью выбора адекватных инструментов для ее построения [2].

### **Методология сравнительного анализа**

Для достижения поставленной цели была разработана методология, основанная на качественном сравнительном анализе по заранее определенному набору ключевых критериев. Эти критерии были выбраны как наиболее значимые с точки зрения практического внедрения и эксплуатации систем изоляции в производственных средах. Первым ключевым критерием является гранулярность и сила изоляции, то есть способность технологии минимизировать поверхность атаки и предотвращать эскалацию привилегий или выход за пределы отведенного периметра. Следующим критерием выступает производительность, а именно накладные расходы, вносимые механизмом изоляции в выполнение системных вызовов, операций ввода-вывода и сетевого взаимодействия. Также рассматривается прозрачность и совместимость, под которыми понимается объем изменений, требуемых для запуска существующих непривилегированных приложений. Наконец, учитывается операционная сложность, включающая в себя легкость развертывания, управления и мониторинга решения [3][4]. Объектами исследования выступают: подсистема eBPF в Linux с акцентом на механизмы безопасного контроля доступа (BPF LSM), архитектура gVisor с его компонентом runsc, а также микроядерная операционная система seL4 в качестве эталонного примера подхода с формально верифицируемым минималистичным ядром. Анализ носит теоретико-архитектурный характер и опирается на документацию проектов, академические публикации и результаты независимых тестирований.

## **Анализ результатов и сравнительная оценка методов изоляции**

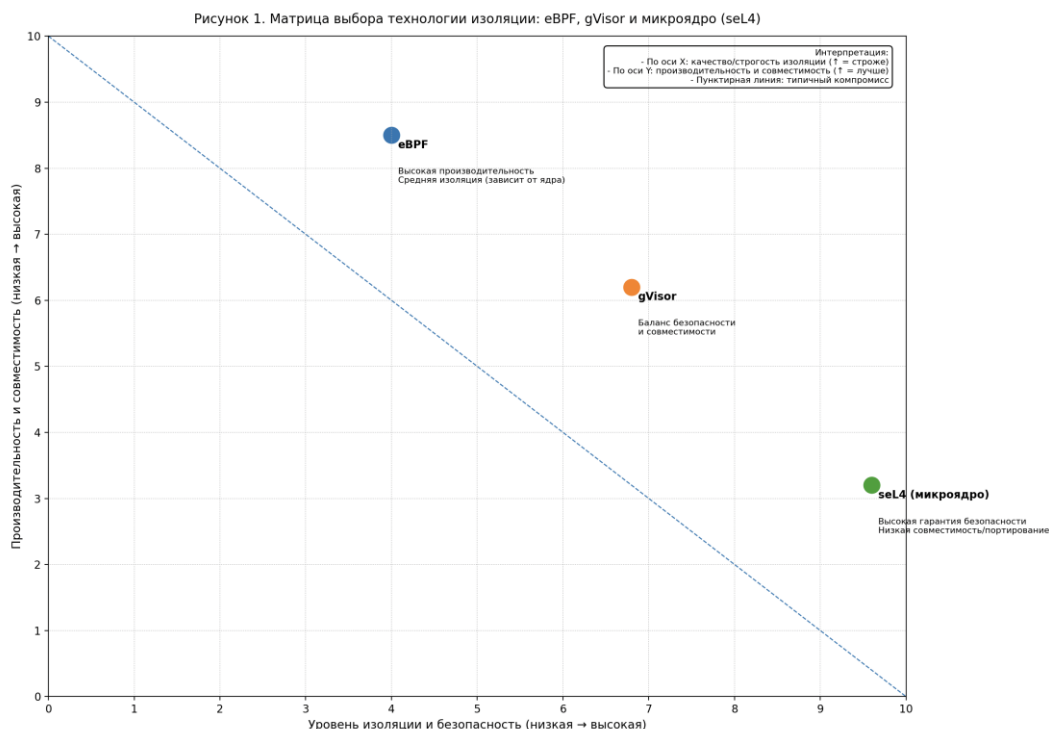
Проведенный анализ трех архитектурных подходов: eBPF, gVisor и микроядерной парадигмы (seL4), позволяет провести их систематическую оценку по установленным критериям. Каждая технология демонстрирует уникальный баланс между безопасностью, производительностью и практической применимостью. Подход на основе расширенного фильтра Беркли (eBPF) представляет собой технологию выполнения детерминированных программ в изолированной виртуальной машине внутри самого ядра операционной системы. Эти программы могут прикрепляться к различным точкам (kprobes, tracerpoints, сокетам) для инспекции, фильтрации и модификации данных и событий в реальном времени. Для задач изоляции eBPF позволяет создавать сложные, динамически загружаемые политики безопасности, например, для контроля системных вызовов (через seccomp-BPF) или мандатного контроля доступа (через LSM-хуки). Главным преимуществом eBPF является его высокая производительность и минимальные накладные расходы, так как фильтрация происходит в контексте ядра без необходимости переключения между пространствами. Однако сила изоляции здесь принципиально ограничена целостностью и корректностью самого монолитного ядра Linux. Успешная атака на ядро позволяет обойти все политики eBPF. Таким образом, eBPF демонстрирует наивысшие показатели по критерию производительности и полную прозрачность для приложений, но предлагает условно средний уровень фундаментальной безопасности, зависимый от надежности более крупной системы [5]. Архитектура gVisor предлагает иной путь, реализуя модель так называемого «пользовательского ядра». В этой модели каждому контейнеру сопоставляется отдельный легковесный процесс-наблюдатель (sentry), который исполняется в пользовательском пространстве хоста и перехватывает все системные вызовы от приложения. Sentry содержит собственную, написанную на Go, минималистичную реализацию критических подсистем

ядра, эмулируя их для рабочей нагрузки. Создается дополнительный, сильно укрепленный барьер между приложением и хост-ядром, что резко сокращает поверхность атаки. Компромиссом становится более высокая стоимость системных вызовов из-за необходимости переключения контекста и эмуляции. Следовательно, gVisor занимает срединную позицию: он обеспечивает существенно более высокий, чем eBPF, уровень изоляции за счет умеренного снижения производительности и неполной прозрачности, требуя совместимости с ограниченным набором эмулируемых syscalls [6].

Микроядерный подход, репрезентативным примером которого является seL4, кардинально меняет саму архитектуру операционной системы. В микроядре в привилегированном режиме выполняется лишь минимальный набор примитивов: управление виртуальной памятью, IPC и планирование потоков. Все остальные компоненты работают как изолированные процессы в пользовательском пространстве. Это обеспечивает максимальный теоретический уровень изоляции, поскольку сбой или компрометация одного компонента не затрагивают другие. Формальная верификация ядра seL4 математически доказывает отсутствие в его коде целого класса уязвимостей. Основным практическим препятствием является низкая аппаратная и программная совместимость, требующая значительных усилий по портированию. Данная парадигма доминирует по критерию силы и верифицируемости изоляции, но демонстрирует наименьшие показатели по совместимости и производительности, которая сильно зависит от качества реализованных в пользовательском пространстве компонентов [7].

Для наглядного отображения баланса между двумя ключевыми критериями, силой изоляции и производительностью/совместимостью была построена сводная матрица (Рисунок 1). Она отражает условную позицию каждой технологии в координатном пространстве «безопасность или удобство». eBPF располагается в области высокой производительности и прозрачности, gVisor занимает центральное, компромиссное положение, а микроядро seL4

находится в зоне максимальных гарантий безопасности при низкой совместимости.



*Ри  
сун  
ок  
1.  
Ма  
тр  
иц  
а  
вы  
бор  
а*

**Рисунок 1. Технологии изоляции: сравнение eBPF, gVisor и микроядерной архитектуры (seL4)**

Итоговая сравнительная характеристика определяет четкие области эффективного применения каждой технологии. eBPF идеально подходит для наблюдения, глубокого аудита и тонкой настройки политик безопасности внутри доверенного периметра, например, в приватном облаке. gVisor является оптимальным решением для публичных облачных платформ (PaaS), сред выполнения недоверенного кода или мультитенантных сред, где баланс между безопасностью и совместимостью критически важен. Микроядерная архитектура остается специализированным решением для встроенных систем и критически важной инфраструктуры, где формально доказанная безопасность превалирует над требованиями универсальности [8].

### **Заключение**

В рамках настоящего исследования был проведен всесторонний

сравнительный анализ трех передовых архитектурных подходов к обеспечению изоляции в операционных системах: eBPF, gVisor и микроядерной парадигмы. Показано, что эволюция технологий изоляции движется по пути создания дополнительных, специализированных уровней абстракции и безопасности поверх или вместо традиционного монолитного ядра. Каждый из рассмотренных методов решает задачу с различным балансом приоритетов. eBPF усиливает безопасность изнутри ядра, предлагая беспрецедентную гибкость и производительность, но в рамках существующей модели доверия. gVisor переосмысливает границу изоляции, вынося ее в пользовательское пространство, что обеспечивает сильную защиту ценой умеренных накладных расходов. Микроядра атакуют проблему фундаментально, минимизируя и верифицируя доверенную вычислительную базу, что делает их эталоном безопасности, но ограничивает применимость. Практическая ценность работы заключается в предоставлении систематизированных критериев для выбора технологии. Для массовых облачных развертываний gVisor представляется наиболее сбалансированным решением. Дальнейшее развитие, вероятно, будет связано с гибридизацией подходов, например, использованием eBPF для оркестрации и мониторинга изолированных сред на основе пользовательских ядер, а также с прогрессом в инструментах верификации для снижения порога входа микроядерных технологий в мейнстрим.

#### **Список использованных источников:**

1. Горбунов В.А., Смирнов А.В. Безопасность контейнерных технологий: анализ угроз и методы защиты [Текст] // Труды СПИИРАН. – 2020. – Т. 19, № 6. – С. 1325–1350.
2. Петров И.Д., Козлов Д.Л. Архитектурные паттерны изоляции в облачных операционных системах [Текст] // Системный администратор. – 2022. – № 3. – С. 18-25.

3.Официальная документация по eBPF [Электронный ресурс] // The eBPF Foundation. – Режим доступа: <https://ebpf.io/what-is-ebpf/> (свободный). – Заглавие с экрана. – Дата обращения: 12.12.2025.

4.gVisor: Architecture Guide [Электронный ресурс] // Google Open Source. – 2023. – Режим доступа: <https://github.com/google/gvisor/blob/master/README.md> (свободный ). – Заглавие с экрана. – Дата обращения: 12.12.2025.

5.Лукичев М.А. Использование eBPF для мониторинга и безопасности в Linux [Текст] // Открытые системы. СУБД. – 2021. – № 2. – С. 40-45.

6.Кузнецов Д.В., Ярмолик В.О. Сравнительный анализ sandbox-решений для контейнерных сред: gVisor, Kata Containers и Firecracker [Текст] // Вестник компьютерных и информационных технологий. – 2023. – № 1(235). – С. 33-40.

7.Зегжда Д.П., Иванов М.А. Формально верифицированные микроядра: состояние и перспективы [Текст] // Информационная безопасность. – 2019. – Т. 16, № 4. – С. 80-89.

8.Таненбаум Э., Бос Х. Современные операционные системы [Текст]. – 4-е изд. – СПб.: Питер, 2020. – 1120 с.

9.Васильев Н.К. Тенденции развития изоляции исполняемых сред: от виртуальных машин к безопасным контейнерам [Текст] // Программная инженерия. – 2022. – Т. 13, № 8. – С. 365-375.

10.Klein G., Andronick J., Elphinstone K., et al. seL4: Formal Verification of an Operating-System Kernel [Текст] // Communications of the ACM. – 2014. – Vol. 57, No. 6. – P. 107-115.