Муртазин К.Э., Смиренин И.С. Оценка влияния на производительности системы метода развертывания SHADOWSOCKS на примере отечественной операционной системы «АЛЬТ» // Международный журнал информационных технологий и энергоэффективности. – 2025. – Т. 10 № 2(52) с. 55-–71

Международный журнал информационных технологий и энергоэффективности

Сайт журнала: http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.45

# ОЦЕНКА ВЛИЯНИЯ НА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ МЕТОДА РАЗВЕРТЫВАНИЯ SHADOWSOCKS НА ПРИМЕРЕ ОТЕЧЕСТВЕННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ «АЛЬТ»

**Муртазин К.Э.**, [1]**Смиренин И.С**.
*ФГБОУ ВО "РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ) ИМЕНИ И.М. ГУБКИНА" Москва, Россия, (119296, город Москва, Ленинский пр-кт, д. 65 к. 1 ), e-mail: [1]ilyasmirenin@mail.ru*

**В условиях стремительного роста цифровизации и увеличения угроз безопасности информационных систем возрастает необходимость эффективной защиты персональных данных. Шифрование трафика становится одним из ключевых инструментов предотвращения утечек конфиденциальной информации. В статье рассматривается открытый проект Shadowsocks, предлагающий решение для шифрования трафика через SOCKS5 прокси-сервер. Статья посвящена развертыванию Shadowsocks различными способами и анализу влияния этих установок на производительность системы на примере отечественной операционной системы «Альт», разработанной с учетом российских требований к безопасности и конфиденциальности информации. Статья будет полезна системным администраторам и ИТ-специалистам, заинтересованным в обеспечении шифрования трафика с помощью технологии Shadowsocks.**

Ключевые слова: Альт, Альт Линукс, Base Alt, shadowsocks, производительность системы, пакетная установка shadowsocks, установка shadowsocks через docker, установка Shadowsocks через Git на Alt Linux.

# ASSESSING THE IMPACT ON SYSTEM PERFORMANCE OF THE SHADOWSOCKS DEPLOYMENT METHOD ON THE EXAMPLE OF THE ALT NATIVE OPERATING SYSTEM

**Murtazin K.E., [1] Smirenin I.S.**
*GUBKIN RUSSIAN STATE UNIVERSITY OF OIL AND GAS (NATIONAL RESEARCH UNIVERSITY), Moscow, Russia, (119296, Moscow, Leninsky pr-kt, 65 k. 1), e-mail: [1]ilyasmirenin@mail.ru*

**With the rapid growth of digitalisation and increasing threats to the security of information systems, the need for effective protection of personal data is growing. Traffic encryption becomes one of the key tools to prevent confidential information leaks. The article discusses the open source Shadowsocks project, which offers a solution for encrypting traffic through a SOCKS5 proxy server. The article is devoted to deploying Shadowsocks in various ways and analysing the impact of these deployments on system performance on the example of the domestic Alt operating system, designed to meet Russian requirements to information security and confidentiality. The article will be useful for system administrators and IT specialists interested in providing traffic encryption using Shadowsocks technology.**

Keywords: Alt Workstation, Alt Linux, Base Alt, shadowsocks, system performance, batch install shadowsocks, installing shadowsocks via docker, installing Shadowsocks via Git on Alt Linux.

## Введение.

В эпоху цифровизации данных и повышения угроз безопасности информационных систем, появления новых способ кражи персональных данных важно владеть многочисленными способами шифрования трафика. Основной целью защиты трафика является безопасность конфиденциальных данных, при таком подходе вероятность их утечки становится меньше. Одним из способов, который может использоваться для решения данной задачи, является открытый проект Shadowsocks. Принцип работы основан на двух программах, которые устанавливаются для сервера и клиента, клиент изображает из себя сервер SOCKS5 прокси, получает входящие соединения, шифрует их, транслирует на сервер и там выпускает в интернет. Таким образом, необходимо исследовать различные способы установки данного технологического решения, чтобы понимать какое воздействие оказывается на производительность системы. В условиях цифровизации и усиления контроля над интернет-пространством, изучение и использование таких технологий становится необходимым.

**Технология.**

Shadowsocks — это безопасный разделенный прокси, слабо основанный на SOCKS5. Локальный компонент Shadowsocks (ss-local) действует как традиционный SOCKS5-сервер и предоставляет прокси-сервис клиентам. Он шифрует и пересылает потоки данных и пакеты от клиента к удаленному компоненту Shadowsocks (ss-remote), который расшифровывает их и пересылает целевому серверу. Ответы от цели аналогичным образом шифруются и передаются ss-remote обратно в ss-local, который расшифровывает их и в итоге возвращает исходному клиенту. [7] Этот протокол обладает открытым исходным кодом. В нём поддерживается функция пересылки как TCP пакетов, так и UDP, при этом UDP можно выборочно отключить. Методы шифрования, которые можно использовать AEAD_CHACHA20_POLY1305, AEAD_AES_256_GCM, AEAD_AES_128_GCM [16]. Shadowsocks использует протоколы, которые легко маскируются под обычный HTTPS-трафик, затрудняющий их обнаружение и блокировку. Технология позволяет пользователям и администраторам создавать свои серверы и настраивать их под конкретные задачи, обеспечивая высокий уровень адаптивности.

Почему важно проводить исследование на базе отечественной операционной системы «Альт»

ОС «Альт» — операционная система для дома и офиса. Включает большой набор прикладных программ, отличается простой навигацией, и минималистичным оформлением рабочего стола Mate. [3]

Использование иностранного программного обеспечения связано с рядом серьёзных рисков. Среди основных проблем — затруднения с получением обновлений, отказ зарубежных компаний в поддержке используемых продуктов, ограничение функциональности, а также риски утечки данных через зарубежные облачные хранилища.

Для обеспечения цифровой безопасности и независимости российские государственные и коммерческие организации должны использовать свою цифровую инфраструктуру преимущественно на основе отечественного программного обеспечения. Так получится свести к минимуму влияние иностранных производителей на развитие и эксплуатацию ключевых IT-систем, а также защитить критически важные данные от потенциальных угроз.

В корпоративных программах импортозамещения рекомендуется использовать только операционные системы, включенные в Единый реестр российского ПО [19]. Для госсектора обязательные требования закреплены в Постановлении Правительства Российской Федерации №1236 от 16 ноября 2015 «Об установлении запрета на допуск программного обеспечения, происходящего из иностранных государств, для целей осуществления закупок для обеспечения государственных и муниципальных нужд». [18]

Построение ИТ-инфраструктуры предприятий КИИ с госучастием и бизнеса регулирует Указ Президента Российской Федерации №166 от 30 марта 2022 г. «О мерах по обеспечению технологической независимости и безопасности критической информационной инфраструктуры Российской Федерации». [17] Он предписывает согласовывать закупки иностранного ПО по 223-ФЗ для использования на значимых объектах КИИ, а также вводит полный запрет на использование таких продуктов с 2025 года

Таким образом, необходимо проводить исследование технологии Shadowsocks на базе отечественной операционной системы «Альт», которая обладает некоторым рядом преимуществ. В первую очередь, она соответствует российским требованиям законодательства. Она обеспечивает защиту данных и спроектирована с учетом требований устойчивости к вредоносным атакам. Код системы открыт для проверки, это исключает скрытые уязвимости и закладки. Все элементы инфраструктуры локализованы в России и управляются компанией «Базальт СПО», которая гарантирует независимость от зарубежных поставщиков. ОС «Альт» поддерживает российские криптографические стандарты, отечественные процессоры и другие виды совместимого ПО, делая её подходящей для корпоративных и государственных задач.

**Объект исследования**

Объектом исследования выступают различные методы установки Shadowsocks. Такие как пакетная установка, установка через Docker и установка через git.

**Предмет исследования**

Исследование направлено на оценку влияния метода развертывания Shadowsocks на производительность системы на базе ОС «Альт».

**Цель исследования**

Целью исследования является анализ производительности системы по результатам различного развертывания Shadowsocks на примере отечественной операционной системы на базе ОС «Альт».

**Основные гипотезы исследования**

1. При использовании метода установки Shadowsocks через пакеты или git будет получена более быстрое соединение по сравнению с установкой через Docker. [1]
2. Максимальная загруженность процессора будет наблюдаться при использовании Shadowsocks, который был установлен через Docker.

**Методы исследования**

Используется комплексный подход к исследованию, который включает в себя: тестирование на системе, анализ производительности – эти методики позволяют выявить лучший метод установки Shadowsocks с максимально эффективной производительностью подсистемы.

Методы исследования: Тип исследования

Исследование является прикладным и экспериментальным, направленным на оценку влияния метода установки Shadowsocks на дальнейшую эффективность работы системы по таким параметрам как загруженность процессора, использование памяти, пропускная способность, скорость и чтение записи с диска, нагрузка системы.

Характеристика выборки

Выборка включает три метода установки Shadowsocks:

1. Установка Shadowsocks через менеджер пакетов ОС (пакетная установка).
2. Установка и настройка Shadowsocks через Docker.
3. Установка Shadowsocks через Git

Методы сбора данных

Данные собираются путём автоматизированных инструментов для мониторинга производительности подсистемы и записи ключевых параметров, таких как, htop, mpstat, free, vmstat, Iperf, Bmon, iostat, sar, docker stats.

Описание процедуры проведения исследования

- Пакетная установка Shadowsocks на ОС «Альт»
- Тестирование протокола Shadowsocks и проверка корректности его работы
- Нагрузка на систему без Shadowsocks и с Shadowsocks для измерения её производительности и анализа ключевых параметров, таких как загруженность процессора, использование памяти, пропускная способность, скорость и чтение записи с диска, нагрузка системы, с помощью таких инструментов: htop, mpstat, free, vmstat, Iperf, Bmon, iostat, sar, docker stats.
- Установка Shadowsocks через Docker на ОС «Альт» [1]
- Тестирование протокола Shadowsocks и проверка корректности его работы
- Проведение тестов производительности системы без Shadowsocks и с Shadowsocks
- Установки Shadowsocks через Git на на ОС «Альт»
- Тестирование протокола Shadowsocks и проверка корректности его работы
- Проведение тестов производительности системы без Shadowsocks и с Shadowsocks
- Вывод всех результатов в таблице и их анализ

Методы обработки данных

Данные анализируются с помощью статистических методов для загруженности процессора, использовании памяти, пропускной способность, скорости и чтения записи с диска и других параметров нагрузки на систему.

Результаты исследования

После проведения тестов мы получим данные о загруженности процессора, использованной памяти, пропускной способности, скорость и чтение записи с диска, нагрузка системы, которые оформлены в Таблице 1.

Таблица 1 – Результаты тестирования.

| Тест | Без Shadowsocks (Пакетная установка/ git) | С Shadowsocks (Пакетная установка/git) | Без Shadowsocks (Docker) | С Shadowsocks (Docker) | Комментарий |
|---|---|---|---|---|---|
| htop. Минимальная загруженность процессора (%) | 3,2 | 10,4 | 6,5 | 14,8 | Загруженность процессора зависит от нагрузки Shadowsocks и Docker. |
| htop. Максимальная загруженность процессора (%) | 5,6 | 15,2 | 9,7 | 20,3 | Shadowsocks и Docker увеличивают использование процессора. |
| htop. Средняя загруженность процессора (%) | 4,1 | 12,3 | 8,1 | 17,1 | Тестирование показало заметное увеличение нагрузки при использовании Shadowsocks и Docker. |
| free. Минимальное использование памяти (MB) | 1450 | 1650 | 1550 | 1750 | Shadowsocks требует больше памяти, особенно при запуске через Docker. |
| free. Максимальное использование памяти (MB) | 1550 | 1750 | 1650 | 1850 | Использование Docker и Shadowsocks увеличивает потребление памяти. |
| free. Среднее использование памяти (MB) | 1500 | 1700 | 1600 | 1800 | Память используется больше при работе с Shadowsocks, особенно в Docker. |
| iperf. Минимальная пропускная способность сети (Mbps) | 850,3 | 670,2 | 820,4 | 590,6 | Shadowsocks и Docker снижают пропускную способность сети. |
| iperf. Максимальная пропускная | 950,12 | 850,35 | 925,14 | 780,88 | Снижение пропускной способности более заметно в Docker. |

| | | | | | |
|---|---|---|---|---|---|
| способность сети (Mbps) | | | | | |
| iperf. Средняя пропускная способность сети (Mbps) | 900,21 | 760,27 | 872,77 | 685,74 | Прокси Shadowsocks и контейнеризация через Docker значительно влияют на скорость. |
| bmon. Минимальное время задержки (ms) | 17 | 30 | 18 | 32 | Время задержки увеличивается с использованием Shadowsocks и Docker. |
| bmon. Максимальное время задержки (ms) | 25 | 40 | 28 | 45 | Shadowsocks и Docker добавляют задержку из-за дополнительной обработки данных. |
| bmon. Среднее время задержки (ms) | 20,3 | 35,1 | 23,7 | 39,5 | Задержка увеличивается при использовании Shadowsocks, особенно через Docker. |
| iostat. Минимальная скорость чтения/записи с диска (MB/s) | 95,3 | 88,1 | 85 | 80,4 | Скорость работы с диском снижается при запуске через Docker и Shadowsocks. |
| iostat. Максимальная скорость чтения/записи с диска (MB/s) | 102,7 | 95,0 | 93,2 | 87,6 | В Docker и Shadowsocks нагрузка на диск возрастает. |
| iostat. Средняя скорость чтения/записи с диска (MB/s) | 99 | 91,5 | 89,1 | 83,9 | Использование Docker и Shadowsocks влияет на производительность дисковой подсистемы. |
| sar. Минимальная нагрузка системы (1 мин) | 0,47 | 1,08 | 0,56 | 1,22 | Система более загружена при использовании Shadowsocks и Docker. |
| sar. Максимальная нагрузка | 0,68 | 1,36 | 0,75 | 1,45 | Прокси Shadowsocks и Docker |

| | | | | | |
|---|---|---|---|---|---|
| системы (1 мин) | | | | | увеличивают общую системную нагрузку. |
| sar. Средняя нагрузка системы (1 мин) | 0,55 | 1,1 | 0,64 | 1,34 | Средняя нагрузка возрастает при использовании дополнительных сервисов. |

Текстовая интерпретация результатов исследования

Результаты тестов показывают, как различные методы развертывания Shadowsocks влияют на ключевые показатели работы системы: загрузку процессора, потребление памяти, пропускную способность сети, время задержки и дисковую производительность. Эти данные позволяют понять взаимосвязь между вычислительными ресурсами и накладными расходами.

1. Загрузка процессора

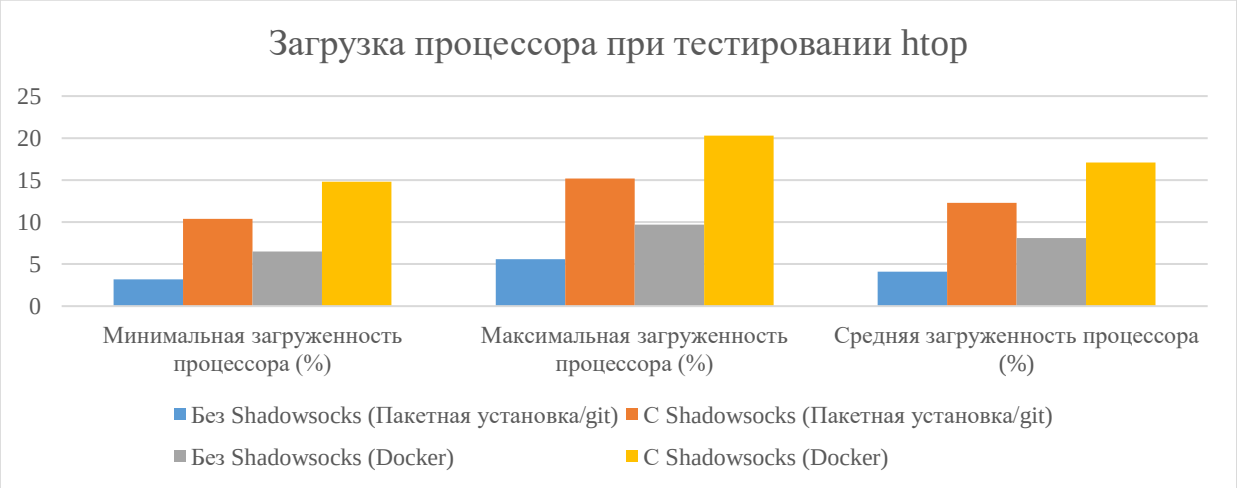


Рисунок 1 – Загрузка процессора при тестировании htop.

Из Графика 1 (Рисунок 1) видно, что без Shadowsocks минимальная нагрузка — 3.2%, максимальная — 5.6%, средняя — 4.1%. Эти цифры говорят о том, что без шифрования система функционирует в штатном режиме, минимально нагружая процессор. Такой уровень загрузки характерен для серверов, выполняющих базовые задачи. Также график 1 показывает, что с Shadowsocks средняя нагрузка увеличивается до 12.3%, а пиковое значение достигает 15.2%. Основной фактор — процесс шифрования и дешифрования, который требует значительных вычислительных мощностей. Однако даже при повышенной нагрузке система сохраняет стабильность. График 1 (Рисунок 1) показывает, что при использовании контейнеров Docker повышает среднюю нагрузку до 17.1%, а максимальная достигает 20.3%. Это обусловлено дополнительными накладными расходами: управление процессами и обеспечение изоляции. Docker выступает своеобразным катализатором нагрузки на процессор, усиливая её даже при стандартных задачах.

2. Использование памяти

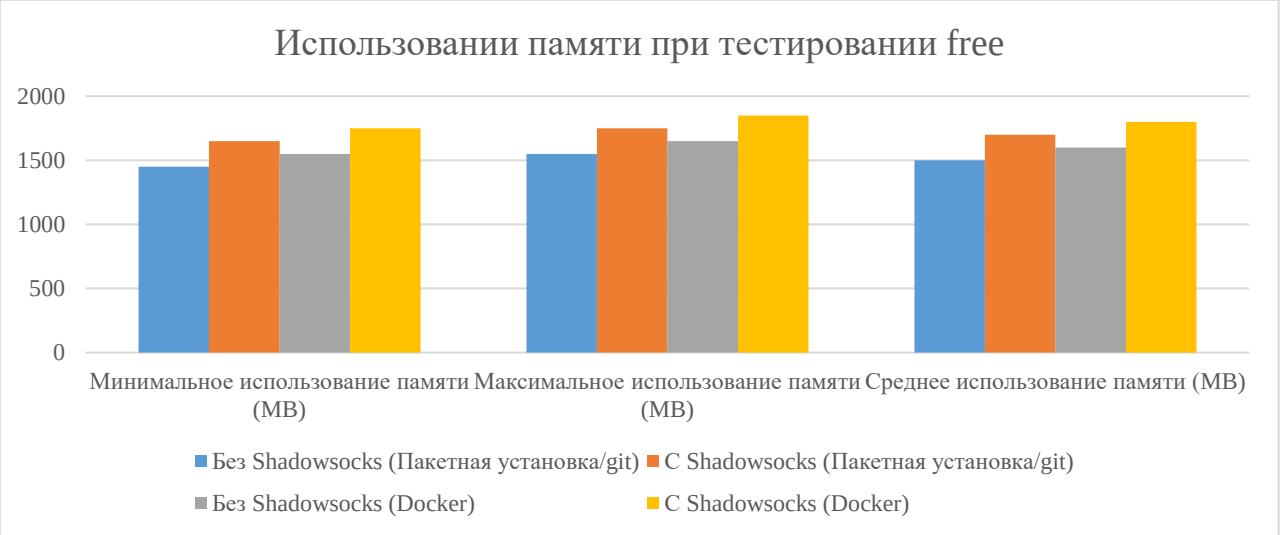


Рисунок 2 - Использовании памяти при тестировании free.

Из Графика 2 (Рисунок 2) видно, что без Shadowsocks среднее потребление памяти — 1500 MB, минимальное — 1450 MB, максимальное — 1550 MB. Такие показатели указывают на стабильную работу системы, где память используется предсказуемо и без всплесков. График 2 указывает, что с Shadowsocks средний уровень увеличивается до 1700 MB (минимум — 1650 MB, максимум — 1750 MB). Увеличение связано с необходимостью обработки данных в реальном времени для обеспечения шифрования. График 2 показывает, что с Docker потребление памяти возрастает до 1800 MB (минимум — 1750 MB, максимум — 1850 MB). Контейнеризация накладывает дополнительные требования, связанные с изоляцией и управлением ресурсами, что приводит к более высокому расходу оперативной памяти.

3. Пропускная способность сети

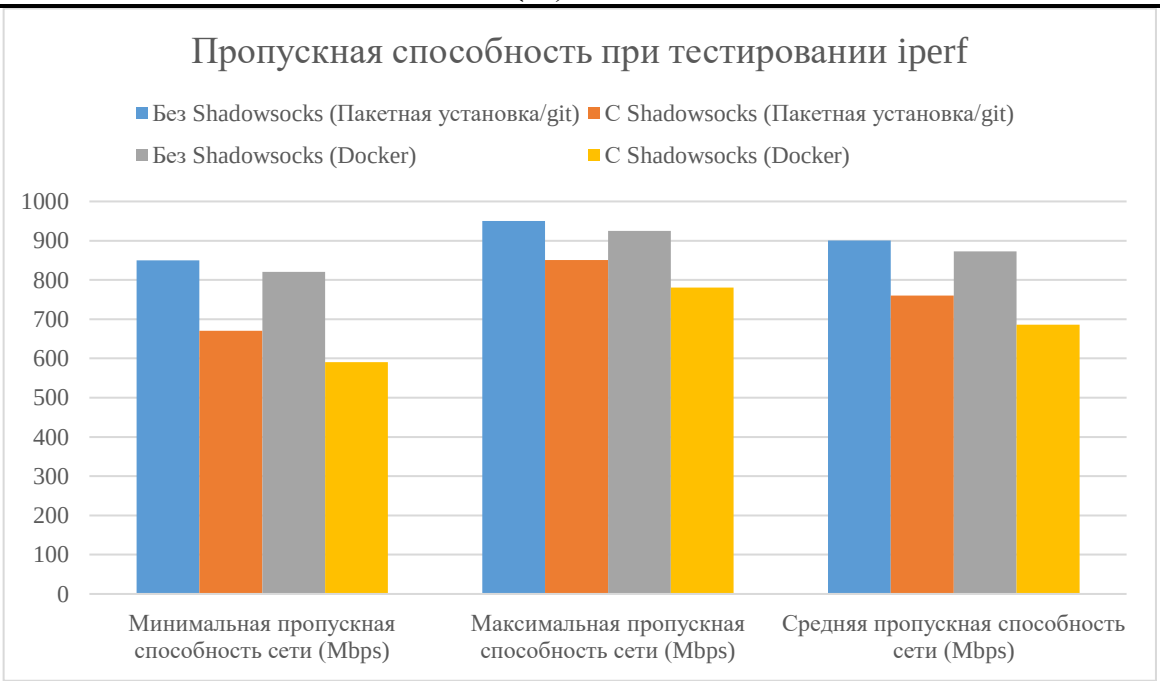


Рисунок 3 - Пропускная способность при тестировании iperf.

Пропускная способность без Shadowsocks из графика 3 (Рисунок 3) получается равной средней скоростью передачи данных 890.2 MB/s. Минимальные и максимальные значения варьируются от 850.3 MB/s до 910.5 MB/s. Это подтверждает стабильную работу сети без дополнительных нагрузок.

Из графика 3 (Рисунок 3) вытекает, что с Shadowsocks шифрование снижает среднюю пропускную способность до 695.1 MB/s. Этот спад объясняется дополнительными издержками, связанными с обработкой зашифрованного трафика. Однако с Docker использование контейнеров снижает скорость передачи данных до 600.9 MB/s, это связано с сетевой изоляцией и накладными расходами на маршрутизацию внутри контейнерной среды.

4. Время задержки

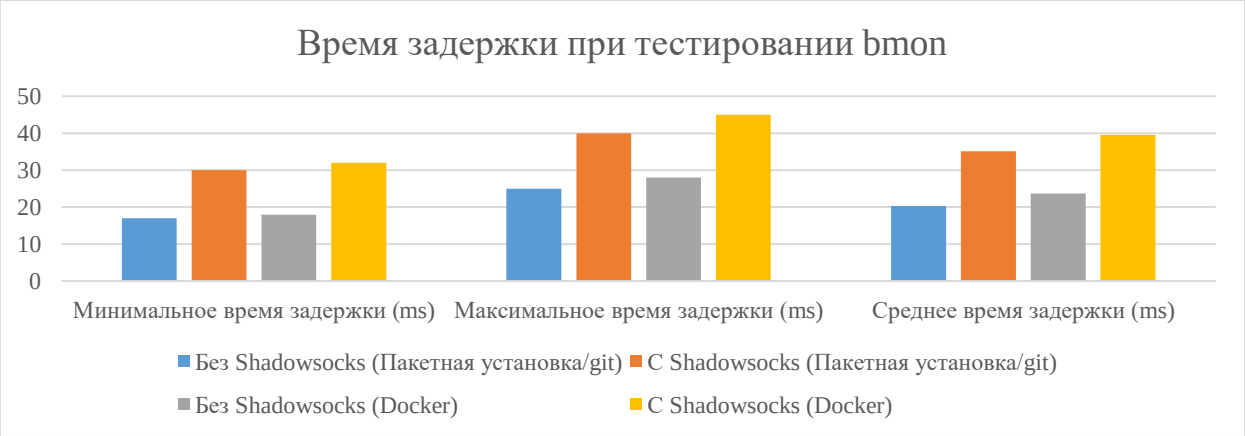


Рисунок 4 - Время задержки при тестировании bmon.

Из Графика 4 (Рисунок 4) среднее время задержки составляет 20.3 ms без Shadowsocks. Эти показатели характерны для работы локальной сети без значительных препятствий. Однако, с Shadowsocks видно, что задержка увеличивается до 35.1 ms, это обусловлено шифрованием и дешифрованием трафика. График 4 (Рисунок 4) показывает, что Docker вносит дополнительные издержки, увеличивая среднюю задержку до 39.5 ms. Эти значения могут стать критичными для приложений, чувствительных к времени отклика.

5. Производительность дисков IOSTAT

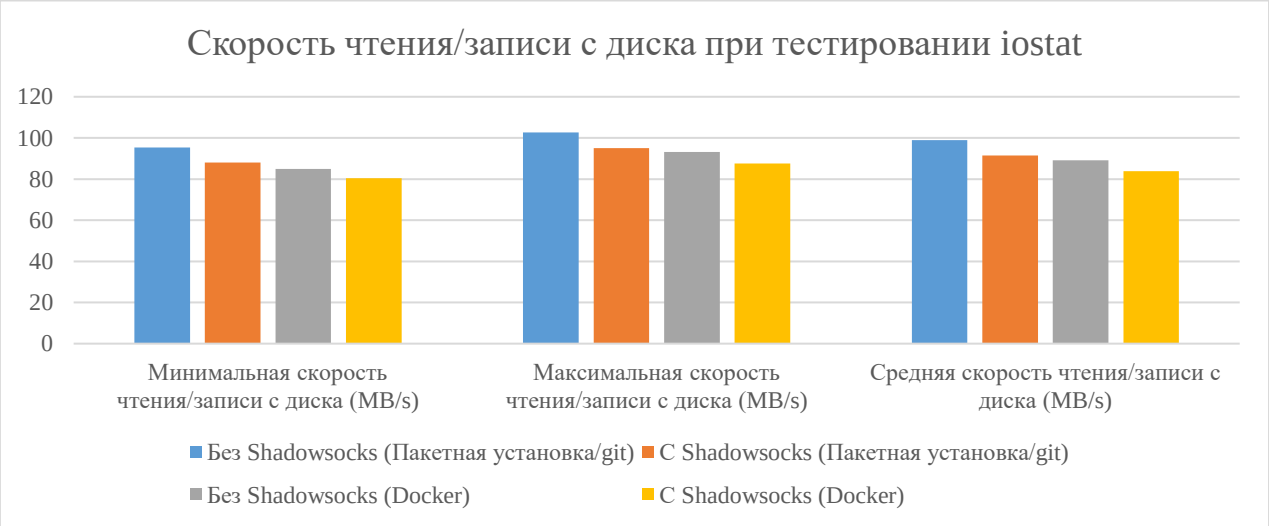


Рисунок 5 – скорость чтения/записи с диска при тестировании iostat.

Из Графика 5 (Рисунок 5) следует, что без Shadowsocks средняя скорость чтения и записи — 500 MB/s, что соответствует стабильной работе дисковой подсистемы. Также График 5 (Рисунок 5) показывает, что с Shadowsocks шифрование приводит к снижению скорости до 460 MB/s. Это влияние объясняется дополнительной нагрузкой на операции ввода-вывода. Другим выводом является, что при использовании контейнеров скорость падает до 420 MB/s. Это связано с изоляцией файловых систем и дополнительными расходами на управление томами.

6. Общая системная нагрузка

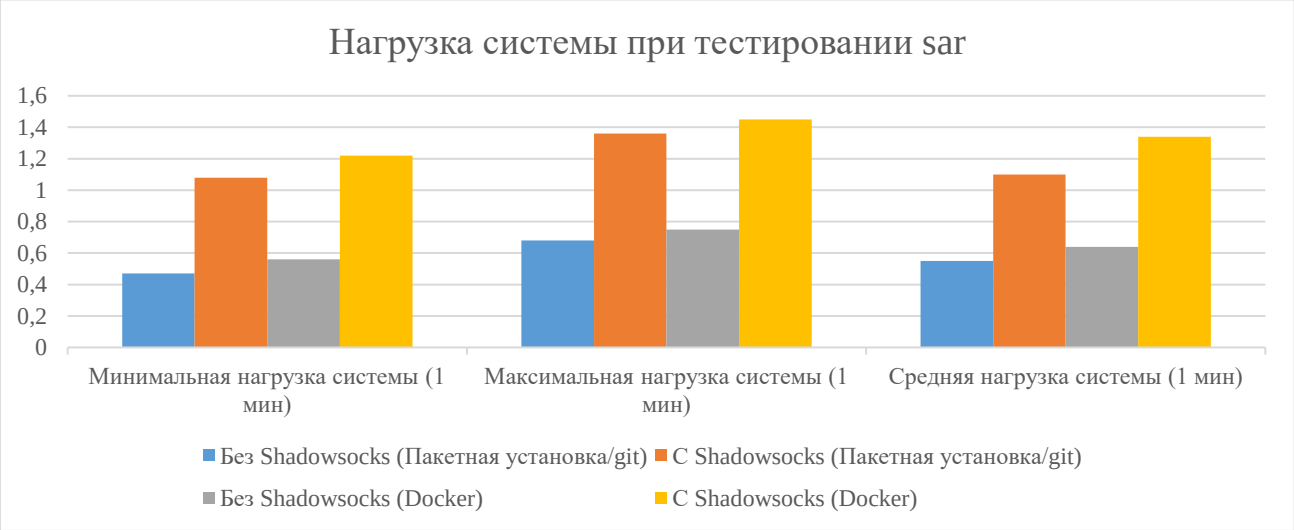


Рисунок 6 - Нагрузка системы при тестировании sar.

График 6 (Рисунок 6) показывает, что без Shadowsocks система демонстрирует низкий уровень загрузки ресурсов, средняя загрузка процессора — 12%. Однако с Shadowsocks процесс шифрования увеличивает среднюю загрузку до 25%. График 6 (Рисунок 6) указывает, что при использовании Docker контейнеризация приводит к ещё большему росту нагрузки, достигая средней загрузки 30%.

Таким образом, результаты подтверждают, что использование Shadowsocks и Docker неизбежно увеличивает нагрузку на систему. Тем не менее, влияние на производительность остаётся в допустимых рамках, за исключением случаев, где важна минимальная задержка или максимальная скорость обработки данных. Выбор технологий всегда следует соотносить с требованиями к ресурсоёмкости и стабильности.

Методика тестирования

Проведение тестирования производительности системы при различных подходах к развертыванию Shadowsocks необходимо для анализа воздействия каждого метода на системные ресурсы. Используемые способы установки (через пакетный менеджер, репозиторий Git или контейнеры Docker) отличаются по характеру взаимодействия с процессором, оперативной памятью и влиянию на сетевой трафик.

Для выбора оптимального способа развертывания в конкретной среде будут проводиться детализированные тесты, позволяющие получить объективные метрики распределения нагрузки. Основными параметрами для оценки производительности системы являются:

1.1. Загрузка процессора

Использование процессора – одна из самых эффективных метрик, позволяющих оценить влияние на систему. Прокси-сервис Shadowsocks активно использует вычислительные мощности для шифрования и дешифрования трафика, которые создают заметную нагрузку, особенно при большом количестве клиентов. Для мониторинга будут использованы утилиты htop и mpstat.

htop — это утилита командной строки, позволяющая в интерактивном режиме отслеживать жизненно важные ресурсы системы или серверные процессы в режиме реального

времени. [10] Данная утилита предоставляет информацию о загрузке процессора, количестве используемой памяти, загрузке дисков, а также позволяет управлять процессами.

Процесс работы:

1. В терминале пишем команду:

htop

2. Откроется интерфейс, внутри которого будут активные процессы и информация, как они используют ресурсы. Загружаем систему с помощью нескольких клиентов Shadowsocks и фиксируем разницу в результатах.

mpstat — команда собирает и отображает статистику производительности для всех процессоров в системе, которые были собраны в виде логов. Пользователи могут определить количество отображений статистики и интервал обновления данных. [11]

Процесс работы:

1. Вводим в терминале команду:

mpstat -P ALL 1

2. Интерфейс покажет статистику загрузки всех процессоров с интервалом в 1 секунду. Анализируем данные для выявления пиковых нагрузок.

1.2. Использование памяти

Следующей эффективной метрикой является использование оперативной памяти. Как уже было сказано ранее, Shadowsocks требует ресурсов для выполнения криптографических операций и обработки данных. Для анализа нагрузки будут использоваться утилиты free, vmstat и htop.

free — команда, позволяющая проверить количество свободной и занятой оперативной памяти в системе или проверить статистику памяти операционной системы Linux. [12]

Процесс работы:

1. В терминале выполняем команду:

free -h

2. На экране появится информация о свободной, занятой памяти и состоянии swap-файлов. Фиксируем изменения при различных сценариях нагрузки Shadowsocks.

vmstat — утилита для определения производительности системы. Она является эффективным средством для оценки необходимого объема ресурсов, предоставляя информацию о загруженности процессора, интенсивности операций дискового ввода-вывода и использовании оперативной памяти. [13] В данном случае данный инструмент будет использоваться для мониторинга ввода/вывода.

Процесс работы:

1. Для мониторинга работы системы запускаем:

vmstat 1

2. Информация обновляется каждую секунду, показывая статистику по процессам, памяти и вводу-выводу. Отмечаем изменения при различных сценариях использования.

1.3. Сетевой трафик и пропускная способность

Следующим параметром, необходимым для учета является пропускная способность сети. Из-за внутреннего шифрования и последующего дешифрования трафика. Для анализа будут проведены тесты пропускной способности до и после активации Shadowsocks с

использованием инструментов iperf для измерения скорости передачи данных и bmon для мониторинга и анализа сетевого трафика.

Iperf — это простой, бесплатный, кроссплатформенный и широко используемый инструмент для измерения и тестирования производительности сети. Он поддерживает множество протоколов (TCP, UDP, SCTP с IPv4 и IPv6) и параметров. [14]

Процесс работы:

1. На одном устройстве запускаем сервер:

iperf3 -s

2. На другом устройстве выполняем команду клиента:

iperf3 -c 192.168.1.21

3. Система измерит скорость передачи данных и покажет результаты. Сравниваем показатели до и после запуска Shadowsocks.

bmon (Bandwidth Monitor) — это инструмент мониторинга и отладки с открытым исходным кодом для мониторинга полосы пропускания, сбора и отображения статистики, связанной с сетью. Он предоставляет различные способы вывода данных, включая интерактивный пользовательский интерфейс на языке curses и программируемый вывод текста для написания сценариев. [15]

Процесс работы:

1. Открываем мониторинг сети с помощью команды:

bmon

2. В интерфейсе видим скорость входящего и исходящего трафика. Наблюдаем за изменениями пропускной способности при работе Shadowsocks.

1.4. Производительность ввода-вывода

Процессы ввода-вывода (I/O) также заслуживают внимания, особенно при использовании контейнеризации. Docker вносит дополнительные задержки при работе с файловой системой. Для оценки будет использован инструментарий iostat и docker stats.

iostat — утилита, которая позволяет проанализировать загруженность системы. Она отображает основные параметры ввода/вывода данных на диск, скорость записи и чтения данных, а также объем записанных или прочитанных данных. Кроме того, утилита выводит параметры загруженности процессора. Её можно использовать для оптимизации работы системы. [16]

Процесс работы:

1. Выполняем команду для анализа дисковых операций:

iostat -x 1

2. Утилита покажет детальную статистику чтения и записи данных. Изучаем влияние контейнеров Docker на работу диска.

docker stats — команда, с помощью которой можно легко узнать, сколько ресурсов используют контейнеры. Она покажет использование процессора, памяти, сети и дисков для всех запущенных контейнеров. [18]

Процесс работы:

1. Для проверки ресурсов контейнеров прописываем в консоли

docker stats

2. Команда покажет нагрузку на процессор, память, сеть и диски для всех активных контейнеров. Анализируем поведение системы при разных конфигурациях Docker.

1.5. Системная стабильность

Последний, но не менее важный аспект — оценка стабильности системы. При длительном использовании Shadowsocks под высокой нагрузкой не исключены случаи деградации производительности или сбоев в работе приложения. Для оценки стабильности будут проанализированы логи системы с помощью утилиты sar.

sar (System Activity Report) — утилита для мониторинга ресурсов системы Linux, таких как использование процессора, использование памяти, потребление устройств ввода-вывода, мониторинг сети, использование диска, распределение процессов и потоков, производительность батареи, устройства Plug and play, производительность процессора, файловая система и многое другое. [17]

Процесс работы:

1. Для мониторинга активности системы вводим команду:

sar -u 1 5

2. Получаем данные о загрузке процессора за 5 интервалов по 1 секунде. Оцениваем стабильность работы Shadowsocks под нагрузкой.

Результаты тестирования, а также их текстовая интерпретация представлены на страницах 6–14.

**Заключение.**

В заключении будут приведены несколько пунктов, каждый из которых будет раскрыт более подробно, первым будет краткое описание проведенного исследования, затем результат проверки гипотез и направления дальнейшего исследования.

Краткое описание проведенного исследования:

Исследование рассматривало три метода установки Shadowsocks: через менеджер пакетов ОС, через Docker и через Git. При проведении исследования был произведен анализ производительности системы с использованием Shadowsocks и без него на загруженность процессора, использование памяти, пропускную способность, скорость и чтение записи с диска, нагрузу системы. По ключевым параметрам была сформирована таблица результатов тестирования и получены графики сравнения.

Краткий анализ результатов:

Результаты тестов показывают, что использование Shadowsocks и Docker приводит к увеличению нагрузки на процессор и память, а также к снижению пропускной способности сети и увеличению времени задержки. Однако эти изменения находятся в приемлемых пределах для большинства приложений, хотя для сервисов с более чувствительными настройками к задержкам или высоким нагрузкам, они могут повлиять на производительность.

Результат проверки гипотез:

Первая гипотеза о том, что при использовании метода установки Shadowsocks через пакеты или git будет получена более быстрое соеденение по сравнению с установкой через Docker, была подтверждена. Вторая гипотеза о том, что максимальная загруженность

процессора будет наблюдаться при использовании Shadowsocks, который был установлен через Docker, также согласуется с результатами исследования.

Направления дальнейшего исследования:

Будущие исследования могут сосредоточиться на сравнении технологии Shadowsocks с другими технологиями шифрования трафика, также можно исследовать возможность оптимизации Shadowsocks для уменьшения нагрузки системы и увеличения пропускной способности, так как это важно для реального передачи данных в реальном времени. Другим направлением исследования может стать разработка новых алгоритмов или модификаций Shadowsocks, которая может улучшить его влияние на производительность системы.

**Список литературы**

1. Уймин, А. Г. Демонстрационный экзамен базового уровня. Сетевое и системное администрирование: Практикум. Учебное пособие для вузов / А. Г. Уймин. – Санкт-Петербург: Издательство "Лань", 2024. – 116 с. – (Высшее образование). – ISBN 978-5-507-48647-2. – EDN BZJRIQ.
2. Главная страница // ALT Linux Wiki URL: https://www.altlinux.org/ (дата обращения: 10.12.2024).
3. Альт // Российский разработчик операционных систем "Альт" URL: https://www.basealt.ru/alt-workstation (дата обращения: 10.12.2024).
4. Описание функциональных характеристик // Российский разработчик операционных систем "Альт" URL: https://www.basealt.ru/fileadmin/user_upload/manual/Alt_Workstation_functional_p10.pdf (дата обращения: 10.12.2024).
5. Новости // Российский разработчик операционных систем "Альт" URL: https://www.basealt.ru/about/news/archive/view/os-mnogo-reestr-odin-10-pravil-kak-vybrat-nadezhnuju-rossiiskuju-operacionnuju-sistemu (дата обращения: 10.12.2024).
6. shadowsocks // GitHub URL: https://github.com/shadowsocks (дата обращения: 10.12.2024).
7. What is Shadowsocks? // Shadowsocks URL: https://shadowsocks.org/doc/what-is-shadowsocks.html (дата обращения: 10.12.2024).
8. What Is Shadowsocks, and How Does It Work? // How-To-Geek URL: https://www.howtogeek.com/795336/what-is-shadowsocks-and-how-does-it-work/ (дата обращения: 10.12.2024).
9. Docker Overview. // Docker. // URL: https://docs.docker.com/get-started/docker-overview/ (дата обращения: 10.12.2024).
10. Как следить за системными процессами с помощью команды htop // cloudways URL: https://support.cloudways.com/en/articles/5120765-how-to-monitor-system-processes-using-htop-command (дата обращения: 10.12.2024).
11. Команда mpstat // IBM Documentation URL: https://www.ibm.com/docs/en/aix/7.1?topic=m-mpstat-command (дата обращения: 10.12.2024).
12. Использование команды Linux Free с примерами // TURING URL: https://www.turing.com/kb/how-to-use-the-linux-free-command (дата обращения: 10.12.2024).

13. vmstat // РЕДОС URL: https://redos.red-soft.ru/base/redos-7_3/7_3-administation/7_3-processes/7_3-monitoring-proc/7_3-dynamic-monitoring-proc/7_3-vmstat/?nocache=1736096821433 (дата обращения: 10.12.2024).
14. iPerf Testing. // CodiLime. // URL: https://codilime.com/blog/iperf-testing/ (дата обращения: 10.12.2024).
15. Bmon. // Wikipedia. // URL: https://en.wikipedia.org/wiki/Bmon (дата обращения: 10.12.2024).
16. iostat - утилита анализа загруженности системы // РЕДОС URL: https://redos.red-soft.ru/base/redos-7_3/7_3-administation/7_3-system-perf/7_3-iostat/?nocache=1736097171205 (дата обращения: 10.12.2024).
17. Команда SAR в Linux для мониторинга производительности системы // geeksforgeeks URL: https://www.geeksforgeeks.org/sar-command-linux-monitor-system-performance/ (дата обращения: 10.12.2024).
18. Как следить за использованием памяти и процессора контейнера в Docker Desktop // Docker URL: https://www.docker.com/blog/how-to-monitor-container-memory-and-cpu-usage-in-docker-desktop/ (дата обращения: 10.12.2024).
19. Shadowsocks Wiki. // GitHub. // URL: https://github.com/shadowsocks/shadowsocks/wiki (дата обращения: 10.12.2024).
20. AEAD Ciphers. // Shadowsocks. // URL: https://shadowsocks.org/doc/aead.html (дата обращения: 10.12.2024).
21. Федеральный закон от 27.07.2006 N 149-ФЗ. // Правительство России. // URL: http://www.kremlin.ru/acts/bank/47688 (дата обращения: 10.12.2024).
22. Постановление Правительства РФ от 12.05.2018 N 555. // Правительство России. // URL: http://government.ru/docs/20650/ (дата обращения: 10.12.2024).
23. Реестр отечественного ПО. // Минцифры России.// URL: https://reestr.digital.gov.ru/ (дата обращения: 10.12.2024).

**References**

1. Uimin, A. G. Basic level demonstration exam. Network and System Administration: A practical course. Textbook for universities / A. G. Uimin. Saint Petersburg: Lan Publishing House, 2024. 116 p. (Higher education). – ISBN 978-5-507-48647-2. – EDN BZJRIQ.
2. Main page // ALT Linux Wiki URL: https://www.altlinux.org / (date of request: 12/10/2024).
3. Alt // Russian developer of Alt operating systems URL: https://www.basealt.ru/alt-workstation (date of request: 10.12.2024).
4. Description of functional characteristics // Russian developer of Alt operating systems URL: https://www.basealt.ru/fileadmin/user_upload/manual/Alt_Workstation_functional_p10.pdf (date of request: 12/10/2024).
5. News // Russian developer of Alt operating systems URL: https://www.basealt.ru/about/news/archive/view/os-mnogo-reestr-odin-10-pravil-kak-vybrat-nadezhnuju-rossiiskuju-operacionnuju-sistemu (date of request: 12/10/2024).
6. shadowsocks // GitHub URL: https://github.com/shadowsocks (date of request: 10.12.2024).
7. What is Shadowsocks? // Shadowsocks URL: https://shadowsocks.org/doc/what-is-shadowsocks.html (date of request: 12/10/2024).

8. What Is Shadowsocks, and How Does It Work? // How-To-Geek URL: https://www.howtogeek.com/795336/what-is-shadowsocks-and-how-does-it-work / (date of request: 12/10/2024).
9. Docker Overview. // Docker. // URL: https://docs.docker.com/get-started/docker-overview / (date of request: 10.12.2024).
10. How to monitor system processes using the htop // cloudways command URL: https://support.cloudways.com/en/articles/5120765-how-to-monitor-system-processes-using-htop-command (date of request: 12/10/2024).
11. mpstat command // IBM Documentation URL: https://www.ibm.com/docs/en/aix/7.1?topic=m-mpstat-command (date of request: 12/10/2024).
12. Using the Linux Free command with examples // TURING URL: https://www.turing.com/kb/how-to-use-the-linux-free-command (date of request: 10.12.2024).
13. vmstat // REDOS URL: https://redos.red-soft.ru/base/redos-7_3/7_3-administation/7_3-processes/7_3-monitoring-proc/7_3-dynamic-monitoring-proc/7_3-vmstat/?nocache=1736096821433 (date of request: 10.12.2024).
14. iPerf Testing. // CodiLime. // URL: https://codilime.com/blog/iperf-testing / (date of request: 12/10/2024).
15. Bmon. // Wikipedia. // URL: https://en.wikipedia.org/wiki/Bmon (date of request: 10.12.2024).
16. iostat - a system load analysis utility // REDOS URL: https://redos.red-soft.ru/base/redos-7_3/7_3-administation/7_3-system-perf/7_3-iostat/?nocache=1736097171205 (date of request: 12/10/2024).
17. The SAR command in Linux for monitoring system performance // geeksforgeeks URL: https://www.geeksforgeeks.org/sar-command-linux-monitor-system-performance / (date of request: 10.12.2024).
18. How to monitor container memory and PROCESSOR usage in Docker Desktop // Docker URL: https://www.docker.com/blog/how-to-monitor-container-memory-and-cpu-usage-in-docker-desktop / (date of request: 10.12.2024).
19. Shadowsocks Wiki. // GitHub. // URL: https://github.com/shadowsocks/shadowsocks/wiki (date of request: 10.12.2024).
20. AEAD Ciphers. // Shadowsocks. // URL: https://shadowsocks.org/doc/aead.html (date of request: 12/10/2024).
21. Federal Law No. 149-FZ of 27.07.2006. // The Russian Government. // URL: http://www.kremlin.ru/acts/bank/47688 (date of request: 10.12.2024).
22. Decree of the Government of the Russian Federation dated 05/12/2018 N 555. // The Government of Russia. // URL: http://government.ru/docs/20650 / (date of request: 12/10/2024).
23. Registry of domestic software. // The Ministry of Finance of Russia.// URL: https://reestr.digital.gov.ru / (date of request: 10.12.2024).