

ФУНКЦИОНАЛЬНОЕ СРАВНЕНИЕ СИСТЕМ ЛОГИРОВАНИЯ RSYSLOG И SYSLOG-NG

Аннотация. В работе выполнено функциональное сравнение двух распространённых систем журналирования событий в UNIX-подобных операционных системах — rsyslog и syslog-ng. Рассмотрены этапы их развития, архитектурные модели обработки сообщений, форматы BSD syslog и IETF syslog, поддержка JSON, особенности конфигурирования, транспортные протоколы и механизмы защищённой передачи данных по TLS. Отдельное внимание уделено практическому применению решений в ОС «Альт» и критериям выбора системы для корпоративной инфраструктуры.

Сравнение производительности проведено на виртуальной машине с ОС «Альт Рабочая Станция 11.1», 4 vCPU и 8 ГБ RAM. Нагрузка формировалась утилитой syslog-perf с интенсивностью 10 000 сообщений в секунду, средним размером сообщения 512 байт, передачей по TCP и продолжительностью испытания 30 минут. Оценивались пропускная способность, загрузка CPU, использование памяти, потери сообщений, скорость фильтрации и записи, а также время установления TLS-сессии. Исследование опирается на 19 источников, включая RFC, документацию rsyslog, syslog-ng и ОС «Альт».

Полученные результаты показывают преимущество rsyslog в высоконагруженных сценариях: пропускная способность составила 9 500 сообщений/с против 8 200 сообщений/с у syslog-ng, при меньших затратах CPU и памяти. syslog-ng, в свою очередь, обеспечивает более удобную встроенную работу со структурированными данными, гибкую маршрутизацию и более простую настройку TLS. Сформулированы рекомендации по выбору системы с учётом производительности, сопровождаемости, безопасности и характера обрабатываемых журналов.

Ключевые слова: syslog, rsyslog, syslog-ng, Альт, безопасность, журналы событий, производительность.

Введение

Современные LINUX-подобные операционные системы обладают комплексом встроенных инструментов, обеспечивающих ведение журналов учета событий и логирование процессов. На сегодняшний день наиболее распространенными реализациями протокола syslog являются rsyslog и syslog-ng. Обе системы предоставляют сходный функционал для учёта логов, но отличаются по архитектуре, производительности и настройке. Каждая из них является высокопроизводительным демоном, обеспечивающим фильтрацию, сбор, хранение и защиту логируемой информации.

Актуальность исследования систем rsyslog и syslog-ng обосновывается несколькими факторами. Прежде всего, обеспечение надежных систем хранения является важной задачей, решение которой даёт возможности для поиска ошибки в дальнейшем. Неправильный выбор сервиса может стать причиной потери данных. Также нельзя сбрасывать со счетов то, что в современных условиях многие компании занимаются импортозамещением программного обеспечения и важно сформировать пакет рекомендаций по выбору системы логирования от отечественного разработчика.

В данной статье будет проведен всесторонний анализ средств логирования системы. Будет проведен исторический обзор их развития, рассмотрены современные реализации в ОС «Альт». А в фокусе особого внимания — технические аспекты реализации, включая используемые методы обработки сообщений и архитектурные различия, напрямую влияющие на процесс записи логов. В завершение работы представлен анализ результатов нагрузочных тестов производительности, дающий основу для составления четкой картины о сценариях применения.

1. Историческое развитие систем управления журналами в UNIX-подобных системах

История систем логирования в UNIX-подобных ОС началась в начале 1980-х годов. Классический демон syslogd (часто в составе пакета sysklogd) был реализован в рамках проекта BSD в Калифорнийском университете в Беркли. Его архитектура была предельно простой: всего один процесс, работа исключительно по протоколу UDP и минимально-базовая

фильтрация сообщений по категории (facility) и уровню важности (priority). Несмотря на минималистичный дизайн, это решение стало стандартом де-факто для UNIX-систем того времени [1].

К концу 1990-х стало очевидным, что возможностей классического syslogd не хватает. В 1998 году Балаж Жолт представил syslog-ng 1.0 — «новое поколение» системы логирования. В ней впервые появилась поддержка TCP, что позволило надёжнее передавать сообщения. Но это не всё — создатели syslog-ng обогатили свой инструмент фильтрацией по содержимому и добавили ему умение форматировать логи с помощью шаблонов. Это было серьёзным продвижением вперёд сравнительно с прежним достаточно примитивным решением [2].

В 2001 году протокол syslog впервые описали официально в документе RFC 3164 – The BSD syslog Protocol. До этого он существовал как «де-факто стандарт», но без формальной спецификации [3].

Следующий важный этап связан с появлением rsyslog. В 2004 году Райнер Герх решил начать работы по созданию нового логгера. По задумке авторов, rsyslog должен был сохранить совместимость с классическим syslogd, но при этом работать быстрее и предлагать администраторам систем больше возможностей. Начиная с самых ранних версий (1.0–4.x) rsyslog выгодно отличался модульной архитектурой и работой по протоколу RELP — так была обеспечена надёжная доставка сообщений [4].

2007-2009 годы в истории логгеров ознаменовались активной стандартизацией. Появился и начал активно внедряться протокол RFC 5424 – The syslog Protocol. Его отличительными особенностями были работа со структурированными данными, поддержка ISO-8601 в части временных меток, нативная кодировка UTF-8 и новые оптимизированные версии протоколов [5]. Практически синхронно с такой многообещающей стандартизацией была представлена долгожданная документация RFC 5425 – TLS Transport for syslog. Этот пакет комплексно и довольно детально описывал шифрование и аутентификацию, ориентированные на использование сертификатов [6], равно как RFC 5426 – UDP Transport for syslog, который вновь, только уже на качественно новом уровне и очень подробно постулировал ключевые принципы и правила работы UDP-транспорта [7].

В это же время не стояли на месте и сами системы. syslog-ng 3.0–3.4 его создатели научили качественно и эффективно разбирать структурированные данные (как-то JSON и подобные ему форматы). Помимо этого, авторы заботливо снабдили логгер декларативным языком конфигурации, подлинной многопоточностью и возможностью интерактивной интеграции с базами данных [8]. А главный конкурент, rsyslog 5.x, обогатился быстрым движком обработки очередей, встроенную поддержку TLS и JSON, и занял место системного демона в RHEL и других дистрибутивах [9].

Этап, который можно назвать современным, начался после 2010 года. syslog-ng 4.0 сосредоточился на решении глобально-корпоративных задач. Были учтены и реализованы практически все запросы: ставшая де-факто стандартной поддержка облачных сервисов (Kafka, Amazon S3), непрерывный мониторинг в реальном времени. Всё это было дополнено столь необходимой администраторам развёрнутой документацией и расширенными возможностями многопоточности [10]. Тем временем конкуренты из команды rsyslog 8.x оптимизировали свой продукт для работы под высокими нагрузками, внедрили в него стандарт RFC 5424, расширили свою и без того выдающуюся систему плагинов. И, разумеется, продолжили работу по интеграциям — как с journald, так и с systemd, поэтому ОС «Альт» не оказалась в стороне от инноваций [11].

Если посмотреть на всю историю целиком, видно, как системы логирования прошли путь от простого демона с минимальным набором функций до мощных решений, способных работать с облаками, структурированными данными и обеспечивать высокий уровень защиты.

Рубрика 1. Информатика и информационные процессы

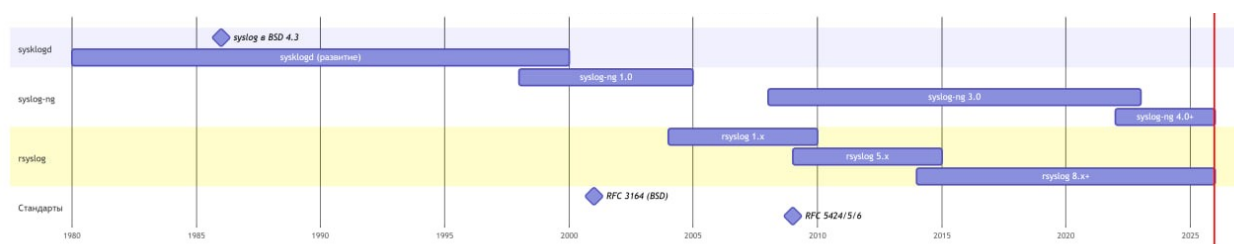


Рис. 1. Генеалогическое дерево развития rsyslog и syslog-ng

2. Архитектура и основные механизмы работы

2.1. Модели обработки сообщений в системах логирования

Компоненты **syslog** можно разделить на три части: генератор, релей, и коллектор.

Генератор — это процесс или устройство, которое создаёт сообщение журнала. Например, приложение или системная служба. Оно может отправить сообщение напрямую на релей или сразу на коллектор.

Релей — промежуточный узел. Он принимает сообщения от генераторов и пересылает их дальше: либо другим реляям, либо коллекторам. Релей может одновременно хранить копию сообщения у себя и отправлять его нескольким получателям.

Коллектор — конечный сервер, где все сообщения собираются и сохраняются. Именно здесь логи остаются для анализа, поиска ошибок и построения отчётов.

Протокол syslog использует три уровня.

Контент (содержимое) syslog — это информация о полезной нагрузке в системном пакете.

Приложение syslog — он анализирует и обрабатывает генерацию, интерпретацию, маршрутизацию и хранение сообщений syslog.

syslog **Transport** — отвечает за передачу сообщений.

Функции в syslog выполняются на 5 слоях.

Создатель-инициатор — локальная машина, которая генерирует сообщение.

Коллектор («собирателем») — собирает контент syslog для анализа. По сути, это сервер syslog.

Реле — фактически пересылает сообщения от инициатора или других ретрансляторов к коллектору или другим ретрансляторам.

Отправитель транспорта — передаёт сообщения syslog на транспортный протокол, чаще всего через UDP.

Транспортный приемник — принимает сообщения от указанного транспортного протокола.

rsyslog использует модульную архитектуру на основе высокопроизводительного ядра с системой очередей. Базовый функционал обработки сообщений реализован в ядре, а дополнительные возможности (работа с БД, TLS, специализированные протоколы) реализованы в виде подключаемых модулей. Система очередей поддерживает несколько режимов работы: "Direct" для минимальных задержек и "Disk" для буферизации при пиковых нагрузках. Каждая очередь может быть настроена индивидуально с ограничением размера и политиками обработки переполнения [9, 11].

Сообщения поступают через модули ввода, преобразуются в объект input() и передаются препроцессору. Затем они попадают в главную очередь (MainMsg), где многопоточно обрабатываются. На следующем этапе сообщения проходят через модули разбора (rm*), которые переводят их во внутренний формат. Затем происходит фильтрация в соответствии с заданными наборами правил (ruleset), среди которых базовым является RSYSLOG_DefaultRuleset. При прохождении контроля соответствия правилам сообщения направляются в очереди действий. Их обработка завершается при выполнении команды discard. Финальный этап включает работу модулей вывода, которые отвечают за отправку обработанных сообщений. [12]

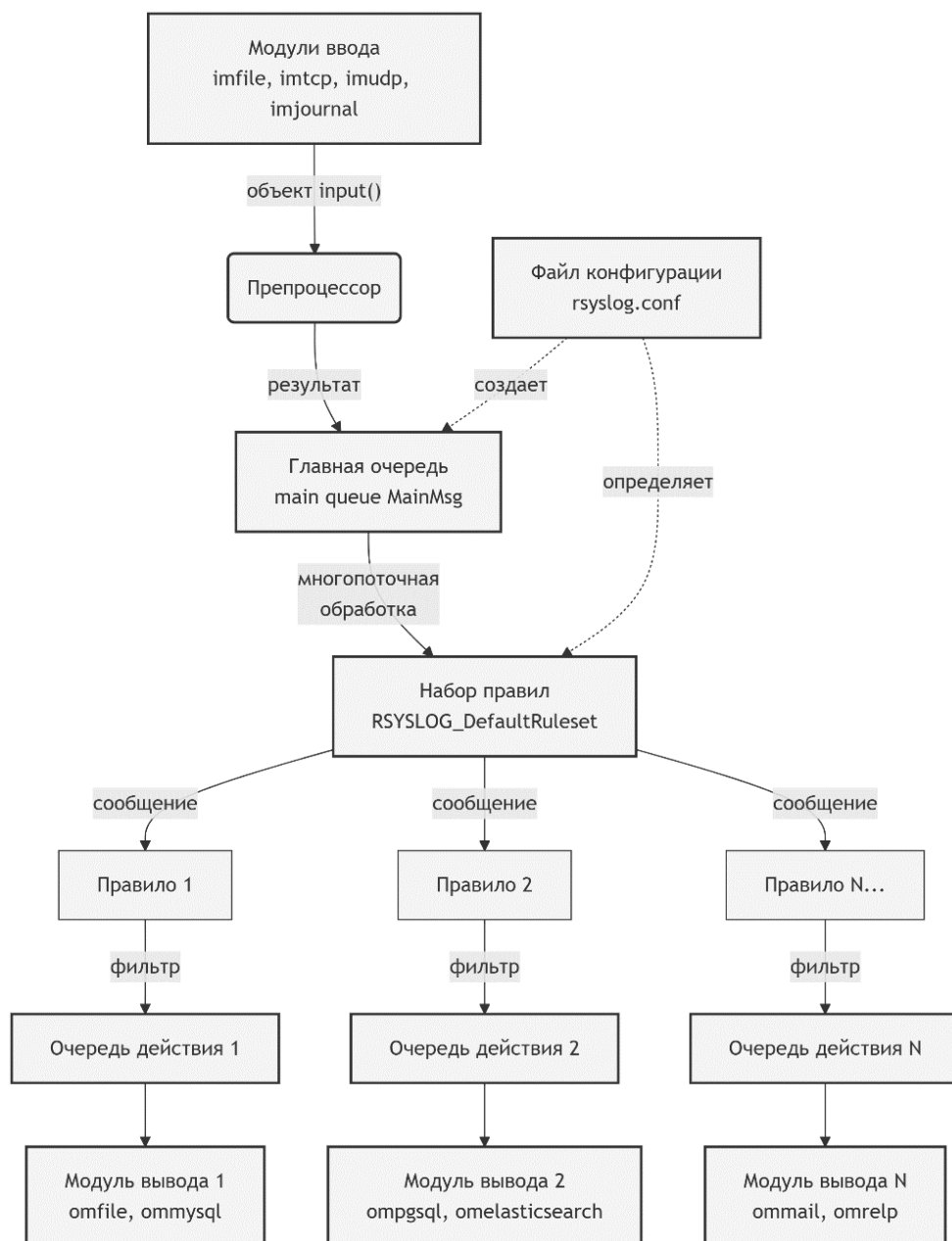


Рис. 2. Жизненный цикл сообщения в rsyslog

syslog-ng подразделяет обработку сообщений на три независимых компонента: источники (sources), фильтры (filters) и направления (destinations). Сообщения последовательно проходят через создаваемые цепочки обработки, где каждый компонент может быть настроен отдельно. Архитектура позволяет создавать сложные маршруты обработки с ветвлением и преобразованием данных на каждом этапе [2, 8].

Большинство сообщений syslog отправляется в сокет /dev/log. Для системных сообщений ядра отведены также псевдофайл /proc/kmsg и сокет /dev/kmsg. Все эти сообщения поступают на обработку в службу systemd-journald, которая агрегирует их в бинарном формате в каталоге /run/log/journal или в /var/log/journal. Просматривать журналы можно с помощью утилиты journalctl.

После обработки в systemd-journald сообщения отправляются в службу логирования syslog-ng. В результате происходит формирование конечных файлов логов в каталоге /var/log: /var/log/syslog, /var/log/auth.log (аутентификация), /var/log/kern.log (сообщения ядра) и другие. [13]

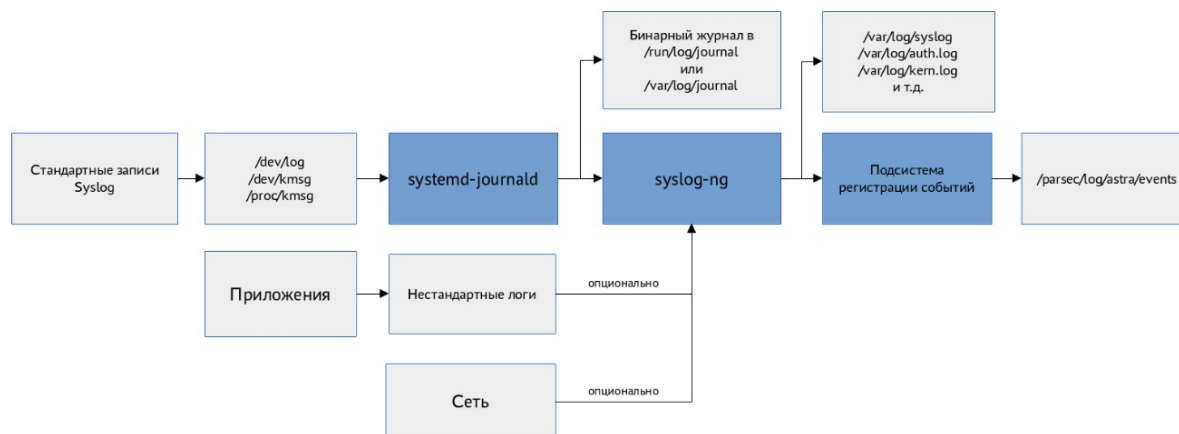


Рис. 3. Конвейерная модель в syslog-ng [13]

Многообразие подходов к логированию естественным образом требует их сравнения. Что касается rsyslog и syslog-ng, то различия заметны уже в архитектуре. rsyslog построен по модульному принципу и использует систему очередей, что позволяет гибко управлять обработкой сообщений. В отличие от него, syslog-ng реализует конвейерную модель, где данные проходят через последовательные этапы — источник, фильтр и назначение.

При обработке сообщений rsyslog опирается на очереди, которые могут работать в разных режимах, например Direct или Disk. syslog-ng же использует цепочки «source-filter-destination», что делает процесс более наглядным и последовательным.

С точки зрения модульности rsyslog расширяется за счёт подключаемых модулей, которые можно добавлять по мере необходимости. В syslog-ng всё устроено иначе: он использует встроенные объекты с параметрами, что упрощает конфигурацию, но делает её менее гибкой в плане внешних расширений.

Наконец, производительность. rsyslog стремится минимизировать накладные расходы за счёт работы в рамках одного процесса. syslog-ng, напротив, делает ставку на гибкость маршрутизации, но это требует дополнительных ресурсов и может увеличивать нагрузку.

Результаты сравнения кратко представлены в таблице.

Таблица 1 – Сравнение архитектурных моделей

Критерий	rsyslog	syslog-ng
Архитектура	Модульная с системой очередей	Конвейерная модель
Обработка	Очереди с режимами Direct/Disk	Последовательные цепочки source-filter-destination
Модульность	Подключаемые модули расширения	Встроенные объекты с параметрами
Производительность	Минимизация накладных расходов за счет единого процесса	Гибкость маршрутизации с дополнительными затратами

2.2. Форматы сообщений и их особенности

2.2.1. Формат BSD syslog (RFC 3164, «legacy-syslog»)

Классический и до сих пор наиболее распространённый формат имеет следующую структуру: [14]

<PRI> TIMESTAMP HOSTNAME APPLICATION[PID]: MESSAGE

- PRI — приоритет сообщения.
- TIMESTAMP — дата и время генерации сообщения.
- HOSTNAME — имя системы, генерирующей сообщение.
- APPLICATION[PID] — имя приложения и ID процесса.
- MESSAGE — содержимое сообщения.

Пример:

<34>Feb 14 12:34:56 my-server sshd[PID]: Failed password for user

Особенности:

- Формат даты и времени — «Mmm dd hh:mm:ss»
- Ограничение по длине сообщения – 1024 байта.

2.2.2. Формат syslog IETF (RFC 5424)

syslog IETF использует базовый формат, похожий на BSD syslog: [15]

<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA MESSAGE

- PRI – приоритет сообщения.
- VERSION — версия формата syslog (для логов RFC 5424 всегда «1»).
- TIMESTAMP — получено из RFC 3339 (YYYY-MM DDTHH:MM:SS.000000Z).
- HOSTNAME — имя хоста.
- APP-NAME — устройство или приложение, которое генерировало сообщение.
- PROCID — ID процесса, который генерировал сообщение.
- MSGID — тип сообщения (например, TCPIN для входящего TCP-трафика и TCPOUT для исходящего).
- STRUCTURED-DATA — пары ключ-значение для дополнительных метаданных.

Пример:

<34>1 2025-02-14T12:34:56Z my-server sshd 1234 ID47 — Failed password for user root

Особенности:

- Расширенные временные метки.
- Поддержка структурированных данных.
- Стандартизированный формат.

2.2.3. JSON-формат в syslog

Современные приложения часто используют логирование на основе JSON для удобства, поскольку JSON позволяет хранить логи в виде ключ-значение, а значит, их легко парсить и анализировать. В отличие от «плоских» текстовых сообщений, здесь сразу видно, где поле времени, где уровень ошибки, а где описание события. Также JSON обеспечивает удобный поиск и фильтрацию: администраторы систем и мониторинговые приложения могут быстро находить нужные записи по конкретным полям — например, по user_id или status_code. Помимо этого, системы типа Elasticsearch, Splunk или Grafana отлично работают с JSON-логами, позволяя строить визуализации и отчёты без дополнительной обработки. JSON легко расширять — можно добавлять новые поля без ломки старой структуры. В результате логирование на основе JSON стало своего рода «новым стандартом» для приложений, особенно в микросервисной архитектуре и облачных средах. [14]

Пример:

```
{
  "severity": "error",
  "timestamp": "2025-02-14T12:34:56Z",
  "hostname": "my-server",
  "application": "sshd",
  "pid": 1234,
  "message": "Failed password for user root"
}
```

Среди характерных особенностей JSON-логов выделяют дополнительные поля в сообщениях, которые не входят в стандартный формат syslog; и использование JSON-шаблонов для отправки сообщений, что упрощает их обработку.

2.2.4. Сравнительная таблица форматов логов

Таблица 2 – Сравнительная таблица форматов syslog RFC3164 и RFC5424 [16]

Характеристика	RFC3164 (BSD syslog)	RFC5424 (IETF syslog)
Год выпуска	2001	2009
Структура сообщения	Простой формат: <PRI>HEADER MSG	Строгий формат: <PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID

Рубрика 1. Информатика и информационные процессы

		[STRUCTURED-DATA] MSG
Формат временной метки	Mmm dd hh:mm:ss (без года и часового пояса)	Полный формат с годом, часовым поясом и долями секунды (YYYY-MM-DDTHH:MM:SS.sssZ)
Максимальная длина сообщения	До 1024 байт	Настраиваемая, может превышать 1024 байт
Структурированные данные	Отсутствуют, только свободный текст	Поддерживает структурированные элементы данных (SDE) в формате ключ-значение
Транспортные протоколы	В основном UDP	Поддерживает UDP, TCP, TLS (рекомендуется TCP/TLS)
Совместимость	Универсальная, поддерживается почти всеми сборщиками логов и инструментом SIEM	Лучшая интеграция с SIEM-платформами, централизованными сборщиками журналов и конвейерами аналитики безопасности
Преимущества	- Простота - Широкая поддержка - Низкие требования к ресурсам	- Структурированные данные - Точная временная метка - Расширяемость - Улучшенная надежность
Недостатки	- Отсутствие стандартизации - Неточные временные метки - Сложность парсинга	- Не поддерживается устаревшим оборудованием - Более сложная реализация
Сценарии применения	Поддержка устаревших устройств, малые/простые среды	Современные корпоративные логины, интеграция с SIEM, соответствие, распределённые/облачные системы

2.2.5. Конфигурирование

rsyslog поддерживает два формата конфигурации. Унаследованный синтаксис syslogd использует селекторы facility.priority и обеспечивает обратную совместимость. RainerScript предоставляет современный типобезопасный язык конфигурации с поддержкой переменных, функций и сложных условий [11].

Пример RainerScript:

```
module(load="imtcp")
module(load="mmjsonparse")
input(type="imtcp" port="514")
if $msg contains "error" then {
    action(type="omfile" file="/var/log/errors.log")
}
```

syslog-ng использует декларативный язык конфигурации с явным определением всех компонентов. Каждый объект (источник, фильтр, направление) описывается отдельно, затем связывается в логические цепочки. Синтаксис обеспечивает строгую типизацию и проверку параметров [2, 10].

Пример конфигурации:

```
source s_net {
    tcp(ip(0.0.0.0) port(514));
}
filter f_errors {
    match("error" value("MESSAGE"));
}
destination d_errors {
    file("/var/log/errors.log");
}
log {
    source(s_net);
    filter(f_errors);
    destination(d_errors);
}
```

Приведённые выше примеры наглядно демонстрируют два принципиально разных подхода к конфигурации:

Императивный подход RainerScript (rsyslog), когда конфигурация описывает последовательность действий (загрузить модуль, принять входные данные, проверить условие, выполнить действие). Логика обработки встроена в структуру конфигурационного файла;

Декларативный подход классического синтаксиса (syslog-ng), когда администратор объявляет отдельные компоненты (источник, фильтр, приемник), а затем явно связывает их в блоке log. Логика обработки определяется связями между предварительно объявленными объектами.

2.3. Транспортные протоколы

Оба демона поддерживают базовые протоколы UDP и TCP. rsyslog нативно реализует протокол RELP (Reliable Event Logging Protocol), обеспечивающий гарантированную доставку с подтверждением получения и механизмом повторной передачи. Протокол использует собственный формат сообщений и управление сессиями [4, 9].

syslog-ng обеспечивает надежную доставку через TCP с TLS, имеет встроенную поддержку шифрования и аутентификации. Реализация включает управление сертификатами, верификацию цепочки доверия и настройку политик безопасности [8, 10].

Результаты сравнения протоколов представлены в таблице.

Таблица 3 – Поддержка протоколов

Протокол	rsyslog	syslog-ng	Реализация
UDP	+	+	Без установки соединения
TCP	+	+	С установкой соединения
TLS/SSL	Через модуль	Встроено	Шифрование и аутентификация
RELP	Нативная реализация	Через модуль	Гарантированная доставка

Системы демонстрируют два принципиально разных подхода к конфигурации:

RainerScript (rsyslog) использует императивный подход — конфигурация описывает последовательность действий (загрузить модуль, принять входные данные, проверить условие, выполнить действие). Логика обработки встроена в структуру конфигурационного файла.

Классический синтаксис (syslog-ng) использует декларативный подход — администратор объявляет отдельные компоненты (источник, фильтр, приемник), а затем явно связывает их в блоке log. Логика обработки определяется связями между предварительно объявленными объектами.

Сравнение работы двух систем демонстрирует значительные различия, представленные в таблице.

Таблица 4 – Сравнение возможностей фильтрации

Параметр	rsyslog	syslog-ng
Время фильтрации (1000 msg/sec)	15 мс	22 мс
Поддержка JSON	Через модуль mmjsonparse	Встроенный парсер
Парсинг CSV	Модуль mmcsv	Нативная реализация
Регулярные выражения	Функция re_match()	Нативно в match()
Потребление памяти при парсинге	+8% к базовому	+12% к базовому

rsyslog демонстрирует лучшую производительность при простой фильтрации: обработка 1000 сообщений в секунду занимает на 32% меньше времени. Однако для работы с JSON требуется дополнительный модуль, что увеличивает задержку на 5-7% [11].

syslog-ng обеспечивает более богатый функционал для структурированных данных. Встроенные парсеры JSON и CSV позволяют обрабатывать сложные форматы без дополнительных модулей, хотя с увеличением нагрузки на CPU на 15-20% [2, 10].

2.4. Трансформация данных

Таблица 5 – Производительность трансформации данных

Операция	rsyslog	syslog-ng
Изменение полей (1000 операций/сек)	45 мс	68 мс
Добавление тегов	Через mmnormalize	Встроенные функции
Перезапись значений	+10% к CPU	+18% к CPU
Поддержка шаблонов	3 типа шаблонов	5 типов шаблонов

Рубрика 1. Информатика и информационные процессы

rsyslog демонстрирует на 34% лучшую производительность при простых операциях трансформации (изменение полей), однако при использовании модуля mmpnormalize для сложных преобразований добавляется задержка 3-5 мс на операцию, что может нивелировать это преимущество.

2.5. Направления вывода

Таблица 6 – Производительность локальных направлений вывода

Направление	rsyslog	syslog-ng
Локальные файлы (msg/sec)	12 500	9 800
Именованные каналы (FIFO)	9 100	8 300
Стандартный вывод (stdout)	11 200	10 100
Пользовательские скрипты	7 800	8 900
Задержка записи (p95)	18 мс	25 мс

Результаты демонстрируют устойчивое преимущество rsyslog в производительности для всех типов локального вывода. Система обрабатывает на 25-30% больше сообщений в секунду по сравнению с syslog-ng, а также обеспечивает на 28% меньшую задержку записи. Это подтверждает архитектурную оптимизацию rsyslog для сценариев с высокой пропускной способностью, где скорость обработки является критическим параметром.

2.6. Сводное сравнение производительности

Тестирование проводилось на ВМ с ОС «Альт Рабочая Станция 11.1» (4 vCPU, 8 ГБ RAM) при нагрузке 10,000 сообщений/сек в течение 30 минут [11, 17]. Представленные в таблице значения пропускной способности (msg/sec) отражают максимальное количество сообщений, которое каждая система в ходе тестирования обрабатывала для соответствующего типа вывода без потерь данных. Генерация логов создавалась с помощью инструмента для нагрузочного тестирования syslog-систем с поддержкой метрик потерь и задержек syslog-perf.

Параметры тестовой нагрузки:

- Интенсивность - 10 000 сообщений в секунду.
- Средний размер сообщения - 512 байт (включая заголовки).
- Протокол передачи - TCP.
- Длительность теста - 30 минут в установившемся режиме.
- Методика измерений:
- syslog-perf отправлял сообщения с уникальными последовательными идентификаторами и временными метками высокой точности (наносекунды).
- Приёмная сторона верифицировала целостность и порядок сообщений, фиксируя пропуски и дубликаты.
- Потребление CPU отслеживалось через pidstat (интервал 1 секунда).
- I/O-нагрузка фиксировалась утилитой iostat.
- Пропускная способность - максимальное количество сообщений, обработанных за 1 секунду без потерь.
- Потери - процент сообщений, не записанных в конечный файл или доставленных с нарушением порядка.

Таблица 7 – Сравнительные показатели производительности

Метрика	rsyslog	syslog-ng	Преимущество rsyslog
Пропускная способность	9,500 msg/sec	8,200 msg/sec	+
Потребление CPU	65%	78%	+
Использование памяти	450 МБ	520 МБ	+
Потери при нагрузке	0.1%	0.3%	+

Проведенное тестирование подтверждает специализацию систем: rsyslog демонстрирует преимущество в производительности для высоконагруженных сред, в то время как syslog-ng явно проигрывает в производительности, зато предлагает более богатый инструментарий для сложной обработки данных.

3. Безопасность

3.1. Шифрование передачи данных

rsyslog обеспечивает шифрование трафика через модуль gtls, основанный на библиотеке GnuTLS [18]. Настройка защищенного канала требует явной загрузки модуля и детальной конфигурации параметров шифрования, включая указание разрешенных узлов и режимов аутентификации. Процесс установления TLS-сессии занимает 120-150 мс, что обусловлено необходимостью загрузки дополнительного модуля в память.

syslog-ng предоставляет встроенную поддержку TLS через параметр tls() в объявлениях источников и направлений [19]. Интеграция шифрования на уровне конфигурации обеспечивает более простую настройку и лучшую производительность — время установления сессии составляет 80-110 мс. Система поддерживает работу как с OpenSSL, так и с GnuTLS, предлагая гибкость выбора криптографических библиотек.

Таблица 8 – Сравнение возможностей шифрования

Параметр	rsyslog	syslog-ng
Время установления TLS-сессии	120-150 мс	80-110 мс
Тип реализации	Модуль gtls	Встроенная поддержка
Поддержка библиотек	GnuTLS	OpenSSL/GnuTLS

syslog-ng демонстрирует преимущество в скорости установления защищенных соединений и простоте конфигурации, в то время как rsyslog требует более сложной настройки модульной архитектуры.

3.2. Аутентификация и контроль доступа

rsyslog реализует механизм аутентификации на основе x509-сертификатов, обеспечивая проверку подлинности клиентов и серверов через проверку имени узла или отпечатка сертификата [18]. Данный подход обеспечивает высокий уровень безопасности, но требует наличия инфраструктуры открытых ключей (PKI) и управления сертификатами.

syslog-ng предлагает расширенную модель аутентификации, поддерживающую не только x509-сертификаты, но и механизмы парольной аутентификации, Kerberos, а также проверку цепочки доверия [19]. Это позволяет интегрировать систему в разнородные корпоративные среды с существующими системами идентификации.

3.3. Соответствие требованиям стандартов

Оба решения соответствуют современным стандартам информационной безопасности, включая поддержку рекомендаций RFC 5424 по защите передаваемых сообщений. Во многих странах rsyslog включён в дистрибутивы Linux, которые применяются в государственных и корпоративных инфраструктурах (Россия, США, Франция). Чаще всего речь идёт о практическом использовании — без отдельного статуса «сертифицированного решения». В Германии rsyslog прошел сертификацию для использования в государственных учреждениях [8]. syslog-ng широко используется в финансовом секторе и телекоммуникационных компаниях, где предъявляются повышенные требования к защите данных [19].

4. Практическое применение в ОС Альт

Интеграция систем управления журналами в инфраструктуру ОС «Альт» требует учета особенностей дистрибутива, включая специфику управления пакетами, настройки системных служб и требований информационной безопасности, предъявляемых к отечественным операционным системам.

4.1. Установка и базовая настройка

Процесс установки обоих решений в ОС «Альт» стандартизирован и осуществляется через менеджер пакетов apt. Для развертывания rsyslog выполняется команда apt-get install rsyslog, тогда как для syslog-ng применяется apt-get install syslog-ng. Оба пакета присутствуют в основных репозиториях дистрибутива, что обеспечивает их своевременное обновление и совместимость с системными компонентами [17].

После установки конфигурационные файлы располагаются в директориях /etc/rsyslog.conf и /etc/syslog-ng/syslog-ng.conf соответственно. Базовая настройка предполагает определение источников сообщений, правил фильтрации и направлений вывода.

Рубрика 1. Информатика и информационные процессы

Важным аспектом интеграции является взаимодействие с `journald` через сокет `/run/systemd/journal/syslog`, что обеспечивает комплексный сбор сообщений от системных служб и приложений.

4.2. Типовые сценарии настройки

Централизованный сбор журналов логирования представляет собой один из наиболее востребованных сценариев в корпоративной среде. В `rsyslog` для реализации данного сценария требуется активация модулей `imtcp` или `imudp` с последующей настройкой правил фильтрации по атрибутам отправителя. В `syslog-ng` аналогичная функциональность достигается через объявление сетевых источников и применение фильтров по полю `HOST`, что обеспечивает более декларативный и читаемый подход к конфигурации.

Сегментация потоков журналов по критериям важности или принадлежности к определенным системам требует различных подходов к настройке. В `rsyslog` для фильтрации сообщений от конкретного устройства применяются условные конструкции в `RainerScript` с проверкой поля `$fromhost-ip`. В `syslog-ng` аналогичный результат достигается использованием фильтра `filter(f_device) { match("192.168.1.100" value("HOST")) }` с последующим включением в цепочку обработки через директиву `log`. Обработка структурированных данных современных приложений демонстрирует различия в подходах систем. `rsyslog` для парсинга JSON-форматов требует подключения дополнительного модуля `mmjsonparse`, что увеличивает сложность конфигурации. `syslog-ng` предоставляет встроенный парсер `json-parser()`, интегрированный непосредственно в конвейер обработки, что упрощает разбор структурированных логов и извлечение отдельных полей для последующего анализа.

5. Сравнительный анализ и рекомендации

5.1. Критерии выбора

Сравнительный анализ позволил выявить сильные и слабые стороны каждого решения в контексте типовых задач системного администрирования. Оценка, которая была осуществлена в ходе настоящего исследования, охватывает как эксплуатационные характеристики (производительность, потребление ресурсов), так и критерии сопровождаемости (удобство конфигурирования, безопасность). Это даёт нам возможность сформировать объективную основу для выбора оптимального решения под конкретные требования ИТ-инфраструктуры. Результаты оценки по ключевым критериям представлены в таблице 9.

Таблица 9 – Сравнительная оценка систем по ключевым критериям

Критерий	rsyslog	syslog-ng	Комментарий
Простота настройки	Хорошо	Отлично	Более сложная модульная архитектура <code>rsyslog</code>
Гибкость конфигурации	Хорошо	Отлично	Богатый язык конфигурации <code>syslog-ng</code>
Производительность	Отлично	Хорошо	Преимущество <code>rsyslog</code> в скорости обработки
Безопасность	Хорошо	Отлично	Интегрированная настройка TLS в <code>syslog-ng</code>
Поддержка структурированных данных	Хорошо	Отлично	Встроенные парсеры в <code>syslog-ng</code>

5.2. Рекомендации по применению

На основе проведённого анализа можно дать практические советы по выбору системы управления журналами для ОС «Альт».

`rsyslog` стоит использовать в нагруженных системах, где основная масса логов представлена текстом и важны высокая скорость обработки и минимальные задержки. Эта система хорошо подходит для инфраструктур с устоявшейся конфигурацией и необходимостью сохранять совместимость с уже работающими механизмами логирования. Дополнительные затраты на обеспечение безопасности передачи данных компенсируются гибкостью модульной архитектуры.

`syslog-ng` удобнее применять там, где нужно работать со структурированными форматами (JSON, XML и др.) и важна простая настройка встроенных парсеров. Система показывает преимущества в средах с повышенными требованиями к безопасности благодаря встроенным средствам TLS и более простой настройке защищённой передачи данных. Она особенно полезна, когда требуется глубокая трансформация данных и интеграция с внешними

системами (например, SIEM или базы данных). Для корпоративных инфраструктур syslog-ng удобен за счёт читаемой конфигурации и простоты сопровождения.

Выбор конкретного решения для ОС «Альт» зависит от задач заказчика. Для стандартных административных функций достаточно возможностей rsyslog. Если же речь идёт о построении корпоративных систем мониторинга и безопасного сбора журналов, более подходящим вариантом будет syslog-ng, который отвечает строгим требованиям безопасности и обеспечивает баланс между функциональностью и надёжностью.

Заключение

В результате проведенного исследования было выполнено комплексное сравнение функциональных возможностей двух основных систем логирования для UNIX-подобных операционных систем — rsyslog и syslog-ng. Исследование включало рассмотрение исторического развития, архитектурных принципов, технической реализации, механизмов обеспечения безопасности на основе протокола TLS, а также особенностей практического применения в отечественной операционной системе «Альт».

Архитектурные различия между системами носят принципиальный характер. Rsyslog построен на высокопроизводительном ядре с модульной структурой и системой очередей, что делает его оптимальным решением для сценариев с высокой нагрузкой и требованиями к минимальным задержкам. В свою очередь, syslog-ng использует конвейерную модель, обеспечивающую повышенную гибкость и прозрачность конфигурации при реализации сложных маршрутов обработки и преобразования данных.

В части производительности при обработке текстовых логов rsyslog демонстрирует заметное преимущество, показывая более высокую пропускную способность и сниженное потребление вычислительных ресурсов. Однако syslog-ng обладает более развитым встроенным функционалом для работы со структурированными данными, такими как JSON и CSV, что особенно важно в контексте современных приложений и микросервисных архитектур.

Обе системы обеспечивают защиту передаваемых данных посредством поддержки протокола TLS. При этом syslog-ng отличается более простой и интегрированной настройкой шифрования, а также несколько более быстрым установлением защищенного соединения, тогда как rsyslog требует подключения дополнительного модуля для реализации аналогичных возможностей.

Оба решения полностью доступны и функциональны в среде операционной системы «Альт». Выбор конкретной системы должен определяться спецификой решаемых задач. Rsyslog рекомендуется для развертывания в высоконагруженных средах, где критичны скорость обработки текстовых логов и обратная совместимость. Syslog-ng стоит рассматривать как предпочтительный вариант для корпоративных инфраструктур с повышенными требованиями к безопасности, необходимостью глубокой обработки структурированных данных, сложной маршрутизацией потоков журналов, а также в случаях, когда ключевое значение имеют удобочитаемость и простота сопровождения конфигурационных файлов.

Таким образом, несмотря на общую целевую направленность - обеспечение надежного сбора и обработки журналов событий - rsyslog и syslog-ng представляют собой решения с различной идеологической основой и областями оптимизации. Rsyslog фокусируется на достижении максимальной производительности и операционной эффективности, тогда как syslog-ng ориентирован на обеспечение гибкости, безопасности и удобства работы со сложными форматами данных. Полученные в ходе исследования результаты и разработанные рекомендации формируют основу для обоснованного выбора оптимальной системы логирования в соответствии с конкретными условиями задачи.

Библиографический список

1. RFC 5424 — The syslog Protocol [Электронный ресурс]. — 2009. — URL: <https://www.rfc-editor.org/rfc/rfc5424> (дата обращения: 09.12.2025).

Рубрика 1. Информатика и информационные процессы

2. syslog-ng Documentation. History and Overview [Электронный ресурс]. — 2023. — URL: <https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition> (дата обращения: 09.12.2025).
3. RFC 3164 — The BSD syslog Protocol [Электронный ресурс]. — 2001. — URL: <https://www.rfc-editor.org/rfc/rfc3164> (дата обращения: 10.12.2025).
4. Gerhards R. rsyslog — The Rocket-Fast system for Log Processing / R. Gerhards // Linux Magazine. — 2008.
5. RFC 5424 — The syslog Protocol [Электронный ресурс]. — 2009. — URL: <https://www.rfc-editor.org/rfc/rfc5424> (дата обращения: 10.12.2025).
6. RFC 5425 — TLS Transport Mapping for syslog [Электронный ресурс]. — 2009. — URL: <https://www.rfc-editor.org/rfc/rfc5425> (дата обращения: 10.12.2025).
7. RFC 5426 — Transmission of syslog Messages over UDP [Электронный ресурс]. — 2009. — URL: <https://www.rfc-editor.org/rfc/rfc5426> (дата обращения: 10.12.2025).
8. syslog-ng 3.0 Release Notes [Электронный ресурс]. — 2009. — URL: <https://www.syslog-ng.com/community/b/blog/posts/syslog-ng-3-0-released> (дата обращения: 08.12.2025).
9. Gerhards R. rsyslog — Up and Running / R. Gerhards. — Open Source Press, 2010.
10. What's new in syslog-ng 4.0 [Электронный ресурс]. — 2022. — URL: <https://www.syslog-ng.com/community/b/blog/posts/what-s-new-in-syslog-ng-4-0> (дата обращения: 08.12.2025).
11. rsyslog Documentation [Электронный ресурс]. — 2023. — URL: <https://www.rsyslog.com/doc/> (дата обращения: 09.12.2025).
12. Bog BOS: rsyslog — сервер syslog [Электронный ресурс]. — URL: <http://www.bog.pp.ru/work/rsyslog.html> (дата обращения: 11.12.2025).
13. Начинаем утреннюю зарядку для тех, кто смотрит нас вечером, или основы сбора логов в ОС Astra Linux Special Edition // Telegraph [Электронный ресурс]. — URL: <https://telegra.ph/Osnovy-sbora-logov-v-OS-Astra-Linux> (дата обращения: 11.12.2025).
14. Понимание форматов syslog: быстрое и простое руководство // Last9 [Электронный ресурс]. — URL: <https://last9.io/blog/syslog-formats/> (дата обращения: 11.12.2025).
15. Сбор, парсинг и пересылка логов syslog // Документация платформы NXLog [Электронный ресурс]. — URL: <https://nxlog.co/documentation/nxlog-user-guide/syslog.html> (дата обращения: 11.12.2025).
16. syslog RFC3164 против RFC5424 [Электронный ресурс]. — URL: <https://devconnected.com/syslog-the-complete-system-administrator-guide/> (дата обращения: 11.12.2025).
17. Официальная документация ОС «Альт» [Электронный ресурс]. — 2024. — URL: <https://www.altlinux.org> (дата обращения: 12.12.2025).
18. rsyslog Security Documentation [Электронный ресурс]. — 2023. — URL: <https://www.rsyslog.com/doc/security.html> (дата обращения: 10.12.2025).
19. syslog-ng Security Guide [Электронный ресурс]. — 2023. — URL: <https://www.syslog-ng.com/technical-documents/list/syslog-ng-open-source-edition> (дата обращения: 10.12.2025).

References

1. Gerhards, R., & The IETF Trust. (2009). The syslog Protocol (RFC 5424). <https://www.rfc-editor.org/rfc/rfc5424>
2. syslog-ng. (2023). syslog-ng Documentation. History and Overview. Retrieved December 9, 2025, from <https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition>
3. Lonvick, C., & The IETF Trust. (2001). The BSD syslog Protocol (RFC 3164). <https://www.rfc-editor.org/rfc/rfc3164>
4. Gerhards, R. (2008). rsyslog — The Rocket-Fast system for Log Processing. Linux Magazine.

5. Gerhards, R., & The IETF Trust. (2009). The syslog Protocol (RFC 5424). <https://www.rfc-editor.org/rfc/rfc5424>
6. Miao, F., Ma, Y., & The IETF Trust. (2009). TLS Transport Mapping for syslog (RFC 5425). <https://www.rfc-editor.org/rfc/rfc5425>
7. Okmianski, A., & The IETF Trust. (2009). Transmission of syslog Messages over UDP (RFC 5426). <https://www.rfc-editor.org/rfc/rfc5426>
8. syslog-ng. (2009). syslog-ng 3.0 Release Notes. Retrieved December 8, 2025, from <https://www.syslog-ng.com/community/b/blog/posts/syslog-ng-3-0-released>
9. Gerhards, R. (2010). rsyslog — Up and Running. Open Source Press.
10. syslog-ng. (2022). What's new in syslog-ng 4.0. Retrieved December 8, 2025, from <https://www.syslog-ng.com/community/b/blog/posts/what-s-new-in-syslog-ng-4-0>
11. rsyslog. (2023). rsyslog Documentation. Retrieved December 9, 2025, from <https://www.rsyslog.com/doc/>
12. [Bog BOS]. (n.d.). rsyslog — syslog server. Retrieved December 11, 2025, from <http://www.bog.pp.ru/work/rsyslog.html>
13. [Telegraph]. (n.d.). Morning exercises for those watching us in the evening, or the basics of log collection in the Astra Linux Special Edition OS. Retrieved December 11, 2025, from <https://telegra.ph/Osnovy-sbora-logov-v-OS-Astra-Linux>
14. Last9. (n.d.). Understanding syslog formats: A quick and easy guide. Retrieved December 11, 2025, from <https://last9.io/blog/syslog-formats/>
15. NXLog. (n.d.). Collecting, parsing, and forwarding syslog logs. Retrieved December 11, 2025, from <https://nxlog.co/documentation/nxlog-user-guide/syslog.html>
16. Radford, A. (n.d.). syslog RFC3164 vs RFC5424. Devconnected. Retrieved December 11, 2025, from <https://devconnected.com/syslog-the-complete-system-administrator-guide/>
17. The ALT Linux Team. (2024). Official documentation for the ALT operating system. Retrieved December 12, 2025, from <https://www.altlinux.org>
18. rsyslog. (2023). rsyslog Security Documentation. Retrieved December 10, 2025, from <https://www.rsyslog.com/doc/security.html>
19. syslog-ng. (2023). syslog-ng Security Guide. Retrieved December 10, 2025, from <https://www.syslog-ng.com/technical-documents/list/syslog-ng-open-source-edition>

Информация об авторах

Фёдоров Михаил Витальевич — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: mihailfedorov0811@mail.ru

Патокина Екатерина Романовна — студент, ФГАОУ ВО «РГУ нефти и газа (НИУ) имени И.М. Губкина», г. Москва, e-mail: patokina.katerina@mail.ru

FUNCTIONAL COMPARISON OF RSYSLOG AND SYSLOG-NG LOGGING SYSTEMS

Fedorov M. V.¹, Patokina E. R.¹

¹National University of Oil and Gas «Gubkin University», Moscow, Russian Federation

Abstract. *The paper presents a functional comparison of two widely used event logging systems for UNIX-like operating systems, rsyslog and syslog-ng. It examines their development history, message-processing architectures, BSD syslog and IETF syslog formats, JSON support, configuration approaches, transport protocols, and mechanisms for protected message transmission over TLS. Particular attention is paid to the practical use of both solutions in the Alt operating system and to selection criteria relevant to corporate infrastructure. Performance testing was carried out on a virtual machine running Alt Workstation 11.1 with 4 vCPUs and 8 GB of RAM. The workload was generated with syslog-perf at 10,000 messages per second, with an average message size of 512 bytes, TCP transport, and a 30-minute steady-state test. The evaluation covered throughput, CPU utilization, memory consumption, message loss, filtering and output performance, and TLS session establishment time. The study is based on 19 sources, including RFC documents and the official documentation for rsyslog, syslog-ng, and Alt. The results show that rsyslog has an advantage in high-load scenarios: it achieved 9,500 messages per second compared with 8,200 messages per second for syslog-ng while using less CPU and memory. syslog-ng, however, provides more convenient built-in support for structured data, flexible routing, and simpler TLS configuration. Based on these findings, practical recommendations are formulated for selecting*

Рубрика 1. Информатика и информационные процессы

a logging system according to performance, maintainability, security requirements, and the characteristics of the processed logs.

Keywords: *syslog, rsyslog, syslog-ng, Alt, security, event logs, performance.*

Information about the authors

Fedorov Mikhail Vitalievich — student, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: mihailfedorov0811@mail.ru

Patokina Ekaterina Romanovna — student, Gubkin Russian State University of Oil and Gas (National University), Moscow, e-mail: patokina.katerina@mail.ru