

Доклад

Система Syslog и журналы событий в Linux

Верниковская Екатерина Андреевна

Содержание

1	Вводная часть	5
2	Введение	6
2.1	Что такое Syslog и зачем он нужен?	6
2.2	История и развитие Syslog	7
3	Система Syslog	8
3.1	Протоколы Syslog	8
3.2	Клиенты и серверы Syslog	9
3.3	Архитектура Syslog	11
3.4	Принцип работы Syslog	14
3.5	Формат сообщений Syslog	15
4	Журналы событий в Linux	19
4.1	Основные журналы событий в Linux	19
5	Настройка и использование Syslog	21
5.1	Конфигурация Syslog	21
5.2	Основные параметры настройки	21
5.3	Настройка локального и удаленного логирования	22
5.4	Настройки фильтров и маршрутизации	23
6	Инструменты для работы с журналами	24
6.1	Rsyslog и Syslog-ng	24
6.2	Команды для анализа журналов	25
7	Практическое применение и анализ журналов	26
7.1	Настройка Syslog на серверах (Linux)	26
7.2	Проверка настроек Syslog	35
7.3	Практические примеры поиска ошибок и предупреждений	37
8	Выводы	39
	Список литературы	40

Список иллюстраций

3.1	UDP и TCP	9
3.2	Архитектура 1	12
3.3	Архитектура 2	13
3.4	Архитектура 3	14
3.5	Формат сообщения syslog	18
7.1	Проверка статуса rsyslog	26
7.2	Установка rsyslog	27
7.3	Проверка статуса rsyslog после установки	27
7.4	Файлы с логами	28
7.5	Файл /etc/rsyslog.conf (1)	28
7.6	Файл /etc/rsyslog.conf (2)	29
7.7	Настройка rsyslog принимать удалённые логи	30
7.8	Настройка отправки логов на удалённый сервер	30
7.9	Фильтрация по программе	31
7.10	Маршрутизация на основе тегов	32
7.11	Фильтрация по уровню сообщений	33
7.12	Файл /etc/rsyslog.conf после редактирования(1)	33
7.13	Файл /etc/rsyslog.conf после редактирования(2)	34
7.14	Перезапуск rsyslog	34
7.15	Проверка статуса rsyslog после редактирования файла	35
7.16	Файл /var/log/error.log	35
7.17	Вход на сервер с помощью SSH	36
7.18	Файл /var/log/sshd.log	36
7.19	Файл /var/log/messages.log	36
7.20	Файл /var/log/auth.log	37
7.21	Файл /var/log/mail.log	37
7.22	Поиск ошибок	37
7.23	Поиск предупреждений	38

Список таблиц

3.1	Коды объектов Syslog и их описание	17
3.2	Уровни серьезности в Syslog	18

1 Вводная часть

Актуальность темы и проблема: система Syslog и журналы событий в Linux играют ключевую роль в обеспечении безопасности, мониторинга и диагностики систем. В условиях растущей киберугрозы и возрастающей сложности IT-инфраструктур необходимость в эффективном управлении логами становится особенно актуальной. Syslog позволяет централизованно собирать, хранить и анализировать события, что значительно упрощает администрирование и повышает уровень безопасности

Объект и предмет исследования: система Syslog и журналы событий в Linux

Цель: цель данного доклада - рассмотреть основные принципы работы системы Syslog и функционирование журналов событий в операционной системе Linux

Задачи исследования: изучить архитектуру системы Syslog и типы журналов событий в Linux

Материалы и методы и инструменты исследования: интернет-ресурсы, аналитика и практические навыки работы на своей операционной системе Linux (Ubuntu)

2 Введение

В процессе своей работы система отслеживает и сохраняет в специальные файлы некоторые события, которые она считает важными или просто нужными для использования в целях исправления и отладки ошибок, сбойных конфигураций и т.д. Файлы, в которых хранятся эти события называются файлами журналов или файлами регистрации. Нередко файлы регистрации занимают слишком много дискового пространства, что может свидетельствовать как о неисправности системы, ошибках конфигураций, так и о просто неправильной настройке демонов регистрации событий, которые отслеживают и собирают всё подряд. Таким образом работа с системой регистрации событий — важная составляющая в работе любого системного администратора, от которой всецело зависит качество обслуживания систем и как следствие — их надёжность и долговечность. [1]

2.1 Что такое Syslog и зачем он нужен?

Syslog (от англ. **system log** — **системный журнал**) — это стандартная система журналирования в операционных системах, включая Linux, регистрирующая события в системе. Позволяет собирать, сохранять и передавать сообщения, генерируемые программами, службами и ядром операционной системы. Помогает администратору отслеживать события, состояния системных служб и выявлять проблемы, возникающие в процессе работы системы.

2.2 История и развитие Syslog

Syslog был разработан в 1980 году Эриком Оллманом (Eric Allman) как часть проекта Sendmail, и использовался первоначально только для Sendmail. Рекомендовав себя как стабильное и удобное решение, Syslog был использован и в других приложениях, став стандартом ведения журналов в системах UNIX и GNU/Linux. Позднее появились реализации и под другие операционные системы. [2]

- 1980-е годы: Syslog сначала использовался только в Sendmail, но вскоре его стабильность и удобство привлекли внимание разработчиков других приложений. Это привело к тому, что он стал стандартом в UNIX-системах для журналирования событий.
- 1990-е годы: Расширение применения Syslog привело к появлению различных реализаций и дополнений, таких как BSD Syslog и другие, что обеспечило поддержку нескольких форматов и уровней логирования.
- 2000-е годы и далее: Syslog продолжал развиваться. Появились версии, соответствующие современным требованиям, включая поддержку сетевого журналирования, новые протоколы (например, RFC 5424) и улучшенные функции безопасности. Современные реализации Syslog поддерживают структурированные данные, что позволяет более точно интерпретировать логи.

Таким образом, Syslog трансформировался из простого механизма журналирования в мощный инструмент для мониторинга и диагностики в различных операционных системах, включая Linux, Windows и другие платформы.

3 Система Syslog

3.1 Протоколы Syslog

Протокол syslog определяет стандарт передачи сообщений журнала между клиентами syslog (отправителями) и серверами syslog (получателями). Он определяет формат и структуру сообщений журнала, а также методы их передачи по сети. [3]

- **UDP (User Datagram Protocol)** — это протокол без установления соединения, который обеспечивает более высокую производительность, но не гарантирует надежную доставку сообщений журнала. Он обычно используется для syslog из-за своей простоты и эффективности, особенно в средах с высоким трафиком. Реализации syslog в Linux часто используют UDP из-за своей скорости.
- **TCP (Transmission Control Protocol)** — это протокол, ориентированный на соединение, который обеспечивает надежную доставку сообщений журнала путем установления постоянного соединения между клиентом и сервером. Хотя он может быть медленнее UDP, он обеспечивает дополнительную надежность и поддерживает такие функции, как подтверждение и повторная передача сообщений.

Протокол syslog работает на стандартизированных номерах портов:

- UDP syslog обычно использует порт syslog 514

- TCP syslog обычно использует порт 601 (официально назначенный) или иногда порт 514 (из соображений совместимости)

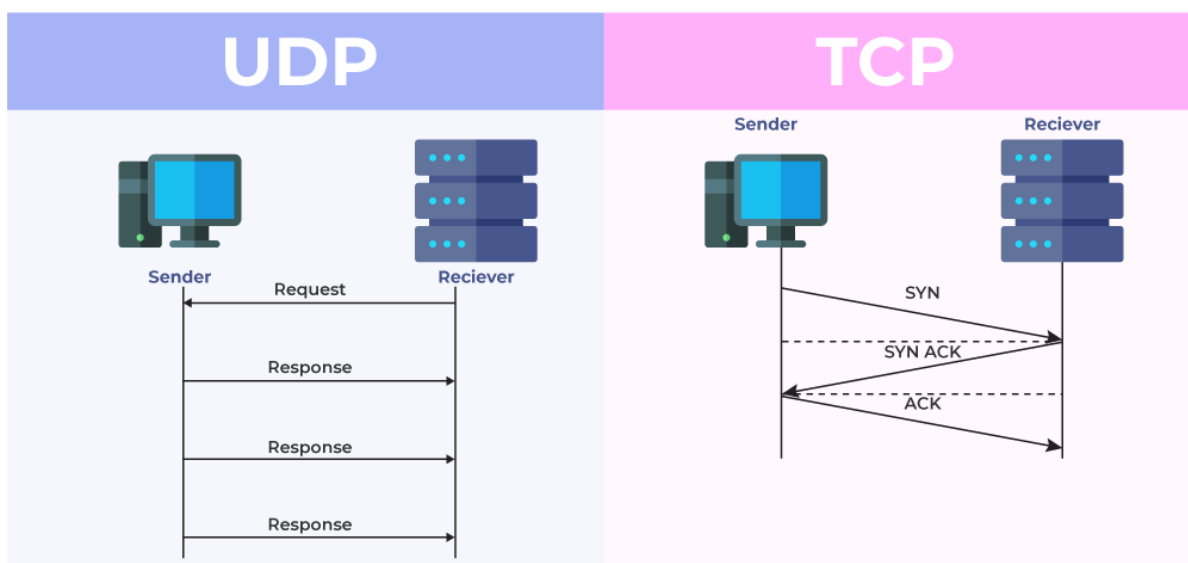


Рис. 3.1: UDP и TCP

Важно отметить, что хотя сам протокол syslog стандартизирован, различные реализации (такие как syslogd, rsyslog и syslog-ng) могут вносить дополнительные функции или изменения в форматы сообщений и механизмы передачи.

3.2 Клиенты и серверы Syslog

Syslog работает на основе клиент-серверной архитектуры. Клиент syslog отвечает за генерацию сообщений журнала и отправку их на сервер syslog. Сервер syslog получает и сохраняет эти сообщения журнала, делая их доступными для анализа, мониторинга и устранения неполадок. [4]

1. Серверы Syslog

Сервер syslog, также известный как syslog-приемник или syslog-демон, представляет собой централизованную систему управления журналами,

которая получает и хранит сообщения журнала из различных источников (syslog-клиентов). Его основные функции включают:

- Прием сообщений по протоколам UDP или TCP
- Фильтрация и маршрутизация сообщений логов в соответствии с установленными правилами
- Хранение логов в файлах для дальнейшего анализа
- Возможность передачи логов на другие серверы для централизации данных (например, для SIEM)
- Поддержка различных форматов выводимых логов для интеграции с другими системами

2. Клиенты Syslog

Клиент syslog, также известный как отправитель syslog или агент syslog, — это программный компонент или приложение, которое генерирует сообщения журнала и отправляет их на назначенный сервер syslog. Клиенты syslog могут быть найдены в различных системах и устройствах, таких как:

- Операционные системы (Linux, Windows, Unix)
- Сетевые устройства (маршрутизаторы, коммутаторы, брандмауэры)
- Приложения и сервисы (веб-серверы, базы данных, облачные сервисы)

Основные функции клиента syslog включают в себя:

- Генерация сообщений логов (например, информации об ошибках, предупреждениях или событиях)
- Отправка сообщений на сервер через протоколы TCP или UDP
- Настройка уровня важности логов (например, отладочные сообщения, предупреждения, ошибки и т.д.)

3.3 Архитектура Syslog

Архитектура Syslog может включать несколько клиентов, формирующих логи, которые по умолчанию отправляют свои сообщения напрямую на центральный коллектор (сервер Syslog). Это базовая конфигурация, которая подходит для небольших инфраструктур. Однако, когда количество клиентов увеличивается, или когда необходимо обеспечить дополнительный уровень безопасности и отказоустойчивости, следует рассмотреть архитектуру с реляями. [5]

Релей (или релей-агент) в системе логирования Syslog — это компонент, который выступает в качестве промежуточного узла между клиентами, генерирующими логи, и центральным сервером, куда эти логи отправляются.

Основные функции реляя:

- *Сбор логов*: Релей принимает логи от нескольких клиентов, что позволяет централизованно управлять данными
- *Агрегация*: Он может объединять сообщения от разных клиентов, уменьшая количество отправок к серверу
- *Кэширование*: Если центральный сервер недоступен, релей может временно сохранять логи и отправлять их после восстановления соединения
- *Безопасность*: Релеи могут использовать шифрование для безопасной передачи логов
- *Гибкая маршрутизация*: Они могут направлять логи на разные серверы в зависимости от их типа или уровня важности

Примеры архитектур:

1. Один клиент и один сервер

Самая простая конфигурация, подходящая для небольших систем, где один логирующий клиент отправляет сообщения прямо на сервер.

АРХИТЕКТУРА 1, ОДНО УСТРОЙСТВО, ОДИН СЕРВЕР



miro

Рис. 3.2: Архитектура 1

2. Много клиентов и один сервер

Увеличивает масштабируемость и позволяет централизованно накапливать логи от нескольких источников, но может быть уязвима к сбоям.

АРХИТЕКТУРА 2, МНОГО УСТРОЙСТВ, ОДИН СЕРВЕР

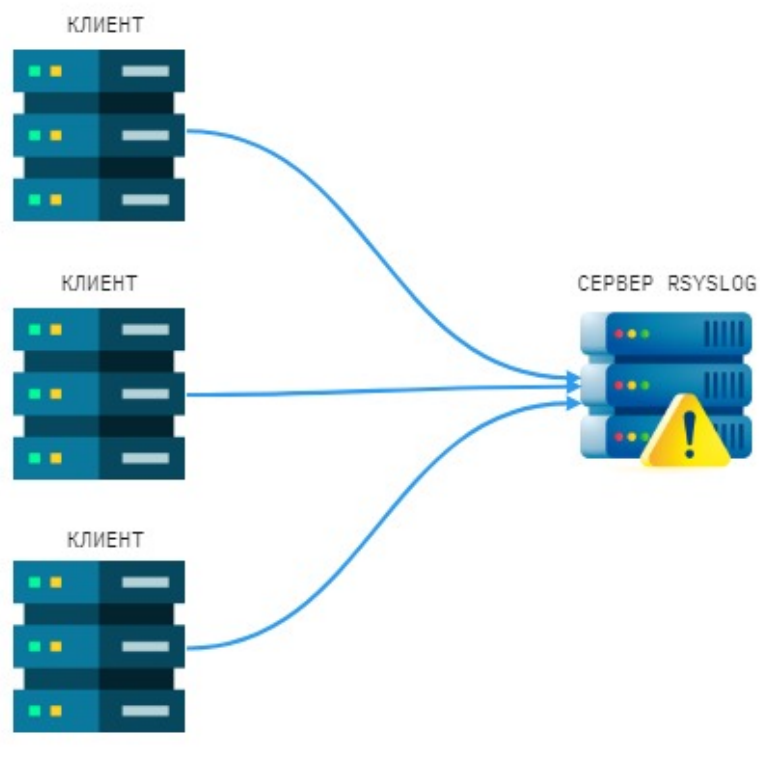


Рис. 3.3: Архитектура 2

3. Много клиентов, один релей, один сервер

Релей действует как промежуточный узел, собирая логи от множества клиентов и отправляя их на сервер. Это обеспечивает отказоустойчивость, распределение нагрузки и централизованное управление.

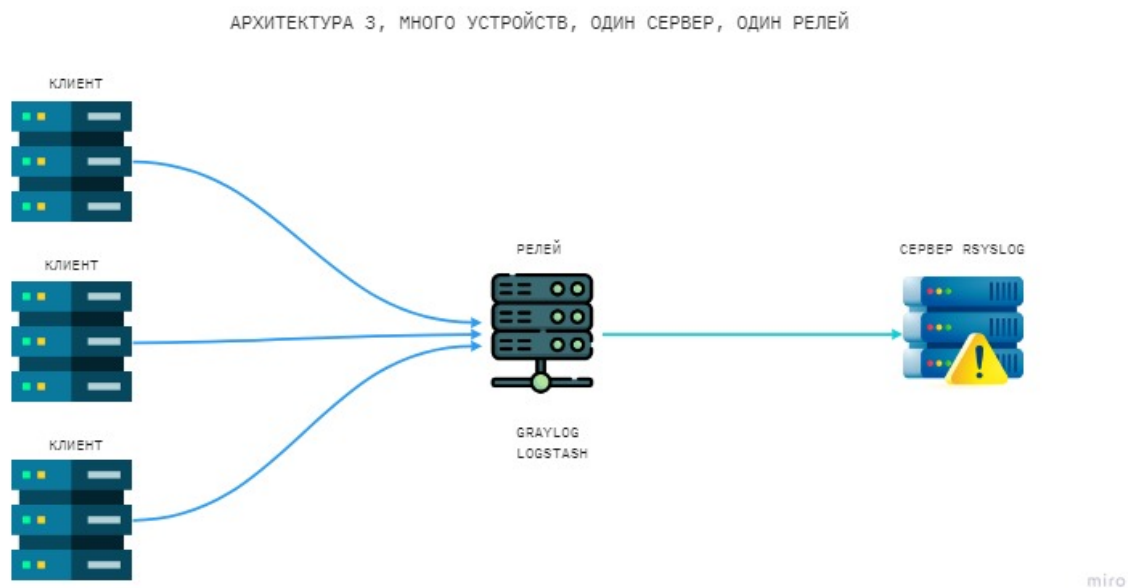


Рис. 3.4: Архитектура 3

3.4 Принцип работы Syslog

1. Генерация журнала

Приложения, системы и устройства генерируют сообщения журнала на основе predetermined событий или условий. Эти сообщения журнала обычно включают такие сведения, как дата и время, источник сообщения журнала и описательное сообщение. Формат syslog помогает стандартизировать эти сообщения, упрощая их интерпретацию. [6]

2. Пересылка журнала

Клиент syslog пересылает сообщения журнала на назначенный сервер syslog, используя протокол syslog. Протокол syslog поддерживает как UDP, так и TCP для передачи сообщений, причем TCP обеспечивает надежную доставку, но потенциально более медленную производительность, чем UDP.

3. Прием журнала

Сервер syslog прослушивает определенный порт (обычно порт 514 для UDP или порт 601 для TCP) и получает сообщения журнала от нескольких клиентов syslog. Правильная настройка порта syslog имеет решающее значение для обеспечения получения сообщений журнала без проблем.

4. Хранение журналов

Сервер syslog сохраняет полученные сообщения журнала в файле журнала или базе данных, в зависимости от конфигурации и требований сервера. Такие инструменты, как rsyslog и syslog-ng, предлагают расширенные возможности для хранения и обработки журналов, поддерживая как традиционные файлы журналов, так и более сложные бэкэнды хранения.

5. Анализ журнала

Системные администраторы, аналитики безопасности и автоматизированные инструменты могут получать доступ и анализировать централизованные данные журнала, хранящиеся на сервере syslog, чтобы получать информацию, выявлять проблемы и контролировать поведение системы и приложений. Анализаторы syslog и визуальные серверы syslog могут помочь в визуализации и понимании данных журнала.

3.5 Формат сообщений Syslog

Сообщения syslog следуют определенному формату, который включает несколько компонентов, что упрощает определение источника, серьезности и другой важной информации о зарегистрированном событии. Понимание структуры сообщений syslog имеет решающее значение для эффективного анализа журнала и устранения неполадок.

Формат syslog делится на три части:

1. *Часть PRI*: в которой подробно описываются уровни приоритета сообщения (от отладочного сообщения до экстренного), а также уровни средств (mail, auth, core);

PRI вычисляется по формуле: **$PRI = facility * 8 + severity$**

Facility (коды объектов/категории) - используются для категоризации источника или типа сообщения журнала и принимают значения от 0 до 23. Они помогают идентифицировать программное обеспечение или компонент, сгенерировавший запись журнала. Обычные коды объектов syslog включают:

Таблица 3.1: Коды объектов Syslog и их описание

Код объекта	Ключевое слово	Описание
0	kern	Сообщения ядра
1	user	Сообщения, сгенерированные в пространстве пользователя
2	mail	Сообщения, связанные с электронной почтой
3	daemon	Сообщения системного демона
4	auth	Сообщения аутентификации и авторизации
5	syslog	Сообщения, генерируемые самим процессом syslog
6	lpr	Сообщения подсистемы строчного принтера
7	news	Сообщения подсистемы сетевых новостей
8	uucp	Сообщения, созданные устаревшей системой UUCP
9	cron	Сообщения, генерируемые службой crond
10	authpriv	Сообщения аутентификации и авторизации
11	ftp	FTP-демон
12	ntp	Подсистема NTP
13	security	Аудит журнала
14	console	Оповещение журнала
15	solaris-cron	Планирование демона
16-23	local0-local7	Сообщения, генерируемые службами, которые настроены из локальных объектов

Severity (приоритеты/серьёзность) - сообщениям syslog назначается уровень серьезности syslog, указывающий важность или срочность зарегистрированного события. Уровни серьезности, варьирующиеся от 0 (Emergency) до 7 (Debug):

Таблица 3.2: Уровни серьезности в Syslog

Значение	Серьёзность	Ключевое слово	Описание
0	Emergency	emerg	система не пригодна для использования
1	Alert	alert	необходимо немедленно принять меры
2	Critical	crit	критические условия
3	Error	err	ошибочные состояния
4	Warning	warning	предупреждающие условия
5	Notice	notice	нормальные, но существенные условия
6	Informational	info	информационные сообщения
7	Debug	debug	сообщения уровня отладки

2. *Часть HEADER:* состоит из двух полей – *TIMESTAMP* (время, обычно в формате “Feb 6 18:45:01”. Согласно RFC 3164, может записываться в формате времени ISO 8601: “2017-02-06T18:45:01.519832+03:00” с большей точностью и с учётом используемой временной зоны) и *HOSTNAME* (имя хоста, сгенерировавшего сообщение);
3. *Часть MSG:* эта часть содержит фактическую информацию о произошедшем событии. Она также делится на поле *TAG* (содержит имя программы, сгенерировавшей сообщение) и поле *CONTENT*

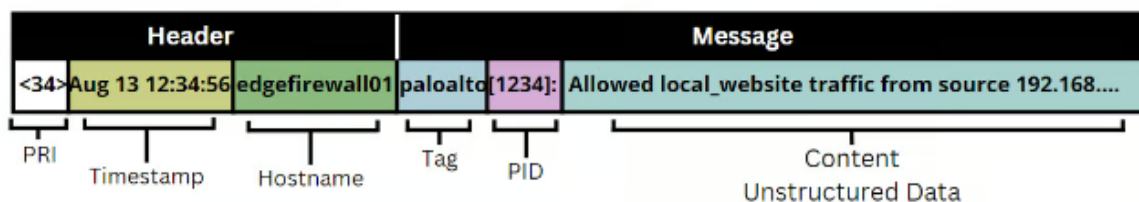


Рис. 3.5: Формат сообщения syslog

4 Журналы событий в Linux

В системах на базе Unix/Linux важное место при администрировании занимает отслеживание системных событий (и в частности возникновение возможных ошибок в процессе настройки каких-то служб) через ведение log-файлов процессов системы. Журналирование системных событий заключается в фиксации с помощью сокета syslog в лог-файлах сообщений об ошибках и сообщений о состоянии работы практически всех процессов системы. Все файлы журналов, можно отнести к одной из следующих категорий:

- приложения (журналы, генерируемые конкретными приложениями, фиксирующие их работу, ошибки, предупреждения и другую активность)
- события (журналы, фиксирующие конкретные события системы, такие как вход в систему, изменения прав доступа и другие действия, которые могут влиять на безопасность и производительность)
- службы (журналы, относящиеся к работе системных служб или демонов, фиксирующие их активность, состояния и ошибки)
- системный (общие системные журналы, описывающие состояние системы и взаимодействие её компонентов)

4.1 Основные журналы событий в Linux

Большинство же лог файлов содержится в директории `/var/log`. В Linux существует несколько основных типов журналов событий:

- */var/log/messages* или */var/log/syslog* — общий файл журнала, в который записывается большинство сообщений системы с момента запуска системы (наиболее часто используемый файл журнала)
- */var/log/dmesg* или */var/log/kern.log* — журнал сообщений ядра системы
- */var/log/secure* — журнал сообщений, связанных с аутентификацией в системе
- */var/log/boot.log* — журнал сообщений, связанных с запуском системы
- */var/log/audit/audit.log* — журнал сообщений аудита (например, в него записываются сообщения SELinux)
- */var/log/maillog* — журналы сообщений, связанных с почтовой службой
- */var/log/samba* — журналы сообщений службы samba (samba по умолчанию не управляется через rsyslogd);
- */var/log/sss* — журналы сообщений службы sssd
- */var/log/cups* — журналы службы печати cups
- */var/log/httpd/* — каталог с журналами веб-службы Apache (Apache записывает сообщения в эти файлы напрямую, а не через rsyslog)
- */var/log/faillog* — неудачные попытки входа

5 Настройка и использование Syslog

Настройка Syslog позволяет эффективно управлять журналированием событий в системе. Основным компонентом для этого является файл конфигурации, где указываются правила фильтрации и маршрутизации сообщений. [7]

5.1 Конфигурация Syslog

Для настройки Syslog используется файл конфигурации, который может находиться в одном из следующих путей:

- Для rsyslog: /etc/rsyslog.conf
- Для syslogd: /etc/syslogd.conf

и /etc/syslog.conf для более старых версий

В этом файле описываются правила и фильтры, определяющие, какие события записывать в лог-файлы, куда их отправлять и как обрабатывать.

5.2 Основные параметры настройки

Основные параметры настройки Syslog, особенно в контексте rsyslog, включают в себя несколько ключевых элементов, которые позволяют гибко управлять логированием в системе.

В файле конфигурации можно настроить следующие параметры:

1. Уровни важности:

- *emerg*: экстренные ситуации, система не может функционировать
- *alert*: необходимо немедленное вмешательство
- *crit*: критические ошибки
- *err*: ошибки
- *warning*: предупреждения
- *notice*: сообщения, которые не являются ошибками, но требуют внимания
- *info*: информационные сообщения
- *debug*: отладочная информация

2. Типы сообщений:

- *auth*: сообщения аутентификации.
- *cron*: сообщения планировщика задач.
- *daemon*: системные демоны.
- *mail*: почтовые службы.
- *user*: пользовательские программы.
- *kern*: сообщения ядра операционной системы.

5.3 Настройка локального и удаленного логирования

Логирование является одной из ключевых операций в системном администрировании, позволяя отслеживать события, производить аудит и устранять неполадки. В современных системах важно не только локальное логирование, но и централизованное удаленное логирование, которое облегчает анализ данных и способствует повышению безопасности.

Локальное логирование подразумевает, что сообщения о событиях сохраняются непосредственно на сервере. Для этого чаще всего используется системный демон, такой как *rsyslog* или *syslog-ng*.

Удаленное логирование обеспечивает централизованное хранение логов с нескольких серверов, что упрощает их анализ и ведение учета. На стороне

удаленного сервера необходимо убедиться в том, что он настроен на прием логов, добавив соответствующие записи в `rsyslog.conf`.

5.4 Настройки фильтров и маршрутизации

Фильтрация и маршрутизация логов — это ключевые аспекты эффективного управления журналами в системах логирования, таких как `Rsyslog` и `Syslog-ng`. Настройка фильтров и маршрутизации позволяет системным администраторам контролировать, какие сообщения записываются, куда они отправляются и как обрабатываются.

Фильтрация — это процесс отбора только тех сообщений, которые соответствуют определённым критериям, позволяющий избегать хранения ненужных или маловажных логов.

Маршрутизация — это процесс отправки выбранных логов в определенные файлы или на удалённые серверы, в зависимости от заданных правил.

Примеры:

- Фильтрация по уровню сообщений
- Фильтрация по программе
- Маршрутизация на основе тегов

6 Инструменты для работы с журналами

В настоящее время для эффективного управления и анализа журналов используются различные инструменты. Два наиболее популярных решения — Rsyslog и Syslog-ng. Они обеспечивают сбор, обработку и отправку системных сообщений, но имеют некоторые ключевые отличия в функциональности и возможностях настройки.

6.1 Rsyslog и Syslog-ng

Rsyslog и Syslog-ng — это два популярных инструмента для управления системными логами, которые предлагают расширенные возможности по сравнению с традиционным syslog.

Rsyslog — это стандартный инструмент для ведения логов в большинстве дистрибутивов Linux. Он предлагает простоту настройки и мощные функции для сбора и обработки логов. Среди его ключевых особенностей — поддержка протоколов TCP и UDP, модульная архитектура и возможность фильтрации и маршрутизации сообщений.

- Поддерживает различные модули для обработки логов (например, для работы с базами данных, удаленными серверами и т.д.)
- Позволяет настраивать фильтрацию и маршрутизацию сообщений
- Поддерживает различные форматы журналов, включая JSON

В то время как Rsyslog отлично подходит для базовых задач по ведению логов, **Syslog-ng** предлагает более широкие возможности для сложных сценариев. Он отличается более гибкой архитектурой и широким спектром дополнительных функций, таких как поддержка различных источников данных и продвинутая фильтрация сообщений. Эти возможности делают Syslog-ng идеальным выбором для более сложных и многогранных окружений.

- Обеспечивает более гибкую архитектуру, позволяя интегрировать данные из различных источников (например, базы данных, сетевые устройства)
- Поддерживает работу с протоколами TLS для безопасной передачи логов
- Предлагает расширенные функции фильтрации и форматирования сообщений

6.2 Команды для анализа журналов

После настройки инструментов для сбора логов, следующим важным шагом становится их анализ. Для этого существует множество команд, которые позволяют фильтровать, обрабатывать и анализировать сообщения из журналов. Для анализа и управления журналами в Linux можно использовать несколько команд:

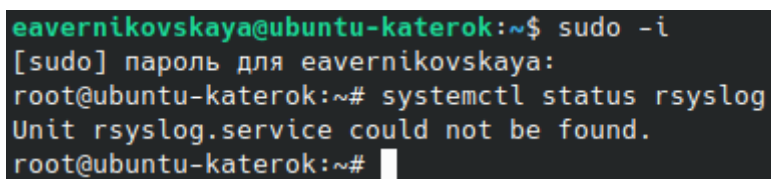
- *tail*: Используется для просмотра последних строк файла журнала.
- *grep*: Позволяет искать строки, содержащие определенные слова или шаблоны.
- *less*: Удобный просмотрщик файлов, позволяющий прокручивать и искать текст.
- *cat*: Отображает содержимое файла целиком.
- *awk* и *sed*: Мощные инструменты для обработки текстовых данных.

7 Практическое применение и анализ журналов

Настроим syslog на Linux(Ubuntu 22.04).

7.1 Настройка Syslog на серверах (Linux)

Установим Rsyslog (на большинстве дистрибутивов Linux Rsyslog установлен по умолчанию).



```
eavernikovskaya@ubuntu-katerok:~$ sudo -i
[sudo] пароль для eavernikovskaya:
root@ubuntu-katerok:~# systemctl status rsyslog
Unit rsyslog.service could not be found.
root@ubuntu-katerok:~#
```

Рис. 7.1: Проверка статуса rsyslog

```

root@ubuntu-katerok:~# sudo apt install rsyslog
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libestr0 libfastjson4
Предлагаемые пакеты:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl | rsyslog-gnutls rsyslog-gssapi
  rsyslog-relp
Следующие НОВЫЕ пакеты будут установлены:
  libestr0 libfastjson4 rsyslog
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакетов, и 225 пакетов не обновлено.
Необходимо скачать 527 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 1 894 kB.
Хотите продолжить? [Д/Н] y
Пол:1 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libestr0 amd64 0.1.10-2.1build3 [7 796 B]
Пол:2 http://ru.archive.ubuntu.com/ubuntu jammy/main amd64 libfastjson4 amd64 0.99.9-1build2 [23,0 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu jammy-updates/main amd64 rsyslog amd64 8.2112.0-2ubuntu2.2 [497 kB]
Получено 527 kB за 0с (1 930 kB/s)
Выбор ранее не выбранного пакета libestr0:amd64.
(Чтение базы данных ... на данный момент установлено 549094 файла и каталога.)
Подготовка к распаковке .../libestr0_0.1.10-2.1build3_amd64.deb ...
Распаковывается libestr0:amd64 (0.1.10-2.1build3) ...
Выбор ранее не выбранного пакета libfastjson4:amd64.
Подготовка к распаковке .../libfastjson4_0.99.9-1build2_amd64.deb ...
Распаковывается libfastjson4:amd64 (0.99.9-1build2) ...
Выбор ранее не выбранного пакета rsyslog.
Подготовка к распаковке .../rsyslog_8.2112.0-2ubuntu2.2_amd64.deb ...
Распаковывается rsyslog (8.2112.0-2ubuntu2.2) ...
Настраивается пакет libestr0:amd64 (0.1.10-2.1build3) ...
Настраивается пакет libfastjson4:amd64 (0.99.9-1build2) ...
Настраивается пакет rsyslog (8.2112.0-2ubuntu2.2) ...
Пользователь «syslog» уже является членом группы «adm».

Creating config file /etc/rsyslog.d/50-default.conf with new version
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
Created symlink /etc/systemd/system/multi-user.target.wants/dmesg.service → /lib/systemd/system/dmesg.service.
Created symlink /etc/systemd/system/syslog.service → /lib/systemd/system/rsyslog.service.
Created symlink /etc/systemd/system/multi-user.target.wants/rsyslog.service → /lib/systemd/system/rsyslog.service.
Обрабатываются триггеры для libc-bin (2.35-0ubuntu3.8) ...
Обрабатываются триггеры для man-db (2.10.2-1) ...
root@ubuntu-katerok:~#

```

Рис. 7.2: Установка rsyslog

Проверим статус службы и убедимся, что она запущена.

```

root@ubuntu-katerok:~# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-14 15:46:00 MSK; 34s ago
 TriggeredBy: ● syslog.socket
   Docs: man:rsyslogd(8)
        man:rsyslog.conf(5)
        https://www.rsyslog.com/doc/
   Main PID: 11359 (rsyslogd)
     Tasks: 4 (limit: 18904)
    Memory: 1.0M
       CPU: 8ms
    CGroup: /system.slice/rsyslog.service
            └─11359 /usr/sbin/rsyslogd -n -iNONE

окт 14 15:46:00 ubuntu-katerok systemd[1]: Starting System Logging Service...
окт 14 15:46:00 ubuntu-katerok rsyslogd[11359]: imuxsock: Acquired UNIX socket '/run/systemd/journal/syslog' (fd 3)>
окт 14 15:46:00 ubuntu-katerok rsyslogd[11359]: rsyslogd's groupid changed to 111
окт 14 15:46:00 ubuntu-katerok systemd[1]: Started System Logging Service.
окт 14 15:46:00 ubuntu-katerok rsyslogd[11359]: rsyslogd's userid changed to 104
окт 14 15:46:00 ubuntu-katerok rsyslogd[11359]: [origin software="rsyslogd" swVersion="8.2112.0" x-pid="11359" x-in>
lines 1-20/20 (END)

```

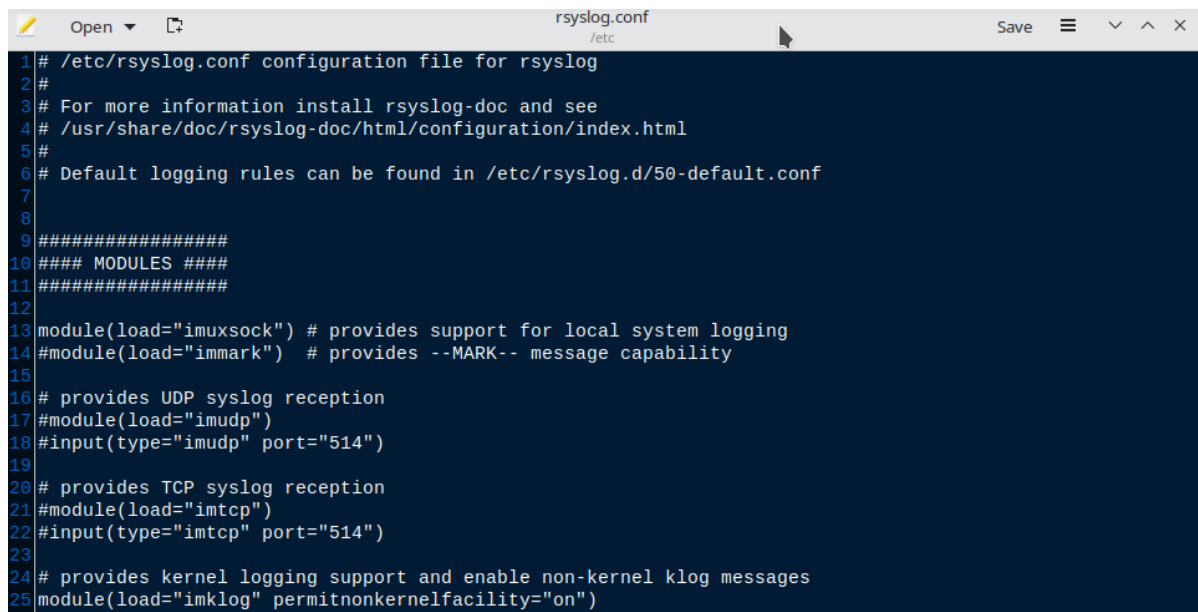
Рис. 7.3: Проверка статуса rsyslog после установки

Проверим, существуют ли какие-то файлы куда записываются логи.

```
root@ubuntu-katerok:~# cat /var/log/error.log
cat: /var/log/error.log: Нет такого файла или каталога
root@ubuntu-katerok:~# cat /var/log/sshd.log
cat: /var/log/sshd.log: Нет такого файла или каталога
root@ubuntu-katerok:~# cat /var/log/mail.log
cat: /var/log/mail.log: Нет такого файла или каталога
root@ubuntu-katerok:~# cat /var/log/auth.log
cat: /var/log/auth.log: Нет такого файла или каталога
root@ubuntu-katerok:~# cat /var/log/messages
cat: /var/log/messages: Нет такого файла или каталога
root@ubuntu-katerok:~#
```

Рис. 7.4: Файлы с логами

Мы видим, что таких файлов не существует. Далее отредактируем файл */etc/rsyslog.conf* для определения того, какие логи и куда направлять.

A screenshot of a text editor window showing the contents of the file /etc/rsyslog.conf. The window has a title bar with 'rsyslog.conf' and '/etc'. The text inside the editor is as follows:

```
1 # /etc/rsyslog.conf configuration file for rsyslog
2 #
3 # For more information install rsyslog-doc and see
4 # /usr/share/doc/rsyslog-doc/html/configuration/index.html
5 #
6 # Default logging rules can be found in /etc/rsyslog.d/50-default.conf
7
8
9 #####
10 #### MODULES ####
11 #####
12
13 module(load="imuxsock") # provides support for local system logging
14 #module(load="immark") # provides --MARK-- message capability
15
16 # provides UDP syslog reception
17 #module(load="imudp")
18 #input(type="imudp" port="514")
19
20 # provides TCP syslog reception
21 #module(load="imtcp")
22 #input(type="imtcp" port="514")
23
24 # provides kernel logging support and enable non-kernel klog messages
25 module(load="imklog" permitnonkernelfacility="on")
```

Рис. 7.5: Файл /etc/rsyslog.conf (1)

```

27 #####
28 #### GLOBAL DIRECTIVES ####
29 #####
30
31 #
32 # Use traditional timestamp format.
33 # To enable high precision timestamps, comment out the following line.
34 #
35 $ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
36
37 # Filter duplicated messages
38 $RepeatedMsgReduction on
39
40 #
41 # Set the default permissions for all log files.
42 #
43 $FileOwner syslog
44 $FileGroup adm
45 $FileCreateMode 0640
46 $DirCreateMode 0755
47 $Umask 0022
48 $PrivDropToUser syslog
49 $PrivDropToGroup syslog
50
51 #
52 # Where to place spool and state files
53 #
54 $WorkDirectory /var/spool/rsyslog
55
56 #
57 # Include all config files in /etc/rsyslog.d/
58 #
59 $IncludeConfig /etc/rsyslog.d/*.conf

```

Рис. 7.6: Файл /etc/rsyslog.conf (2)

1. Удалённое логирование:

- На сервере-получателе:

Добавляем следующие строки, чтобы настроить rsyslog принимать удалённые логи:

```

module(load="imudp") # Для UDP
input(type="imudp" port="514")
module(load="imtcp") # Для TCP
input(type="imtcp" port="514")

```

```
#####
#### MODULES ####
#####

module(load="imuxsock") # provides support for local system logging
#module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
```

Рис. 7.7: Настройка rsyslog принимать удалённые логи

- На сервере-отправителе:

Добавляем строки для настройки отправки логов на удалённый сервер:

```
*.* @remote-server-ip:514 # Для UDP
*.* @@remote-server-ip:514 # Для TCP
```

```
# Send logs to remote Syslog server
*.* @172.16.82.68:514 # Используйте @ для UDP
*.* @@172.16.82.68:514 # Используйте @@ для TCP
```

Рис. 7.8: Настройка отправки логов на удалённый сервер

2. Фильтрация по программе:

Добавим правило, которое перенаправляет все логи, полученные от процесса sshd (сервер SSH) в отдельный файл `/var/log/ssh.log`:

```
if $programname == 'sshd' then /var/log/secure
if $programname == 'sshd' then ~
```

- if \$programname == 'sshd': это условие, которое определяет, какие лог-события будут затронуты правилом

- \$programname - это переменная rsyslog, содержащая имя процесса, который генерирует лог-событие. Условие проверяет, равно ли имя процесса 'sshd'
- then /var/log/ssh.log: это действие, выполняемое в случае выполнения условия. Лог-события, удовлетворяющие условию, будут записаны в файл /var/log/ssh.log
- then ~ : это действие останавливает дальнейшую обработку лог-событий, удовлетворяющих условию. После записи в /var/log/ssh.log эти события больше не будут обрабатываться другими правилами в rsyslog.conf

Таким образом это правило гарантирует, что все логи от SSH-сервера будут собраны в отдельном файле, упрощая их анализ и отслеживание.

```
# Filter SSHD logs to a separate file
if $programname == 'sshd' then /var/log/ssh.log
if $programname == 'sshd' then ~ # Не записывать эти сообщения больше нигде
```

Рис. 7.9: Фильтрация по программе

3. Маршрутизация на основе тегов:

Настроим фильтр для сортировки логов и их записи в соответствующие файлы. Например, пусть все сообщения с уровнем серьезности “error” будут записываться в отдельный файл /var/log/error.log. Для этого добавим строки:

```
if $syslogseverity-text == 'error' then /var/log/error.log
```

- if \$syslogseverity-text == 'error': это условие, которое определяет, какие лог-события будут затронуты правилом
 - \$syslogseverity-text - это переменная rsyslog, содержащая текстовое описание уровня серьезности лог-события
 - Условие проверяет, равно ли значение переменной 'error'.

- `then /var/log/error.log`: это действие, выполняемое при выполнении условия. Лог-события, удовлетворяющие условию (имеющие уровень серьезности “error”), будут записаны в файл `/var/log/error.log`

Это позволяет централизованно собирать все сообщения об ошибках в отдельный файл, упрощая анализ и отслеживание проблем.

```
# Filter error messages to a separate file
if $syslogseverity-text == 'error' then /var/log/error.log
```

Рис. 7.10: Маршрутизация на основе тегов

4. Фильтрация по уровню сообщений:

И наконец добавим правила, которые перенаправляют логи разных системных служб в отдельные файлы:

```
# Log various facilities to specific files
authpriv.* /var/log/auth.log          # Все сообщения авторизации
mail.* /var/log/mail.log              # Все сообщения, связанные с почтой
*.info;mail.none;authpriv.none;cron.none /var/log/messages # Остальные сообщения
```

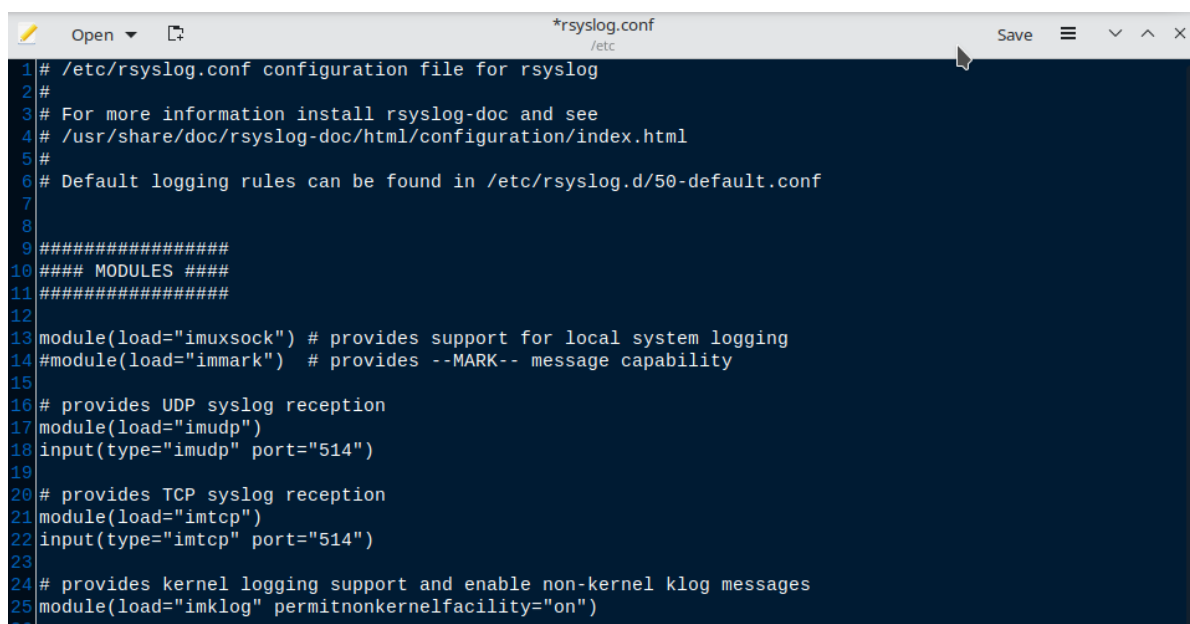
- `authpriv.* /var/log/auth.log`: это правило перенаправляет все сообщения, относящиеся к авторизации (facility `authpriv`), в файл `/var/log/auth.log`
 - `authpriv.*`: соответствует всем лог-событиям, относящимся к `authpriv`
 - `/var/log/auth.log`: путь к файлу, куда будут записаны события
- `mail.* /var/log/mail.log`: это правило перенаправляет все сообщения, связанные с почтой (facility `mail`), в файл `/var/log/mail.log`
 - `mail.*`: соответствует всем лог-событиям, относящимся к `mail`
 - `/var/log/mail.log`: путь к файлу, куда будут записаны события
- `*.info;mail.none;authpriv.none;cron.none /var/log/messages`: это правило перенаправляет все сообщения с уровнем серьезности `info` и выше, кроме тех, которые относятся к `mail`, `authpriv`, `cron`, в файл `/var/log/messages`

- *.info: соответствует всем сообщениям с уровнем серьезности info и выше
- mail.none: исключает все сообщения, относящиеся к mail
- authpriv.none: исключает все сообщения, относящиеся к authpriv
- cron.none: исключает все сообщения, относящиеся к cron
- /var/log/messages: путь к файлу, куда будут записаны события

В итоге, эти правила разделяют логи разных системных служб на отдельные файлы, что облегчает анализ и отслеживание событий.

```
# Log various facilities to specific files
authpriv.* /var/log/auth.log      # Все сообщения авторизации
mail.* /var/log/mail.log         # Все сообщения, связанные с почтой
*.info;mail.none;authpriv.none;cron.none /var/log/messages # Остальные сообщения
```

Рис. 7.11: Фильтрация по уровню сообщений



```
*rsyslog.conf
/etc

1 # /etc/rsyslog.conf configuration file for rsyslog
2 #
3 # For more information install rsyslog-doc and see
4 # /usr/share/doc/rsyslog-doc/html/configuration/index.html
5 #
6 # Default logging rules can be found in /etc/rsyslog.d/50-default.conf
7
8
9 #####
10 ##### MODULES #####
11 #####
12
13 module(load="imuxsock") # provides support for local system logging
14 #module(load="immark") # provides --MARK-- message capability
15
16 # provides UDP syslog reception
17 module(load="imudp")
18 input(type="imudp" port="514")
19
20 # provides TCP syslog reception
21 module(load="imtcp")
22 input(type="imtcp" port="514")
23
24 # provides kernel logging support and enable non-kernel klog messages
25 module(load="imklog" permitnonkernelfacility="on")
26
```

Рис. 7.12: Файл /etc/rsyslog.conf после редактирования(1)

```

27 #####
28 #### GLOBAL DIRECTIVES ####
29 #####
30
31 #
32 # Use traditional timestamp format.
33 # To enable high precision timestamps, comment out the following line.
34 #
35 $ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
36
37 # Filter duplicated messages
38 $RepeatedMsgReduction on
39
40 #
41 # Set the default permissions for all log files.
42 #
43 $FileOwner syslog
44 $FileGroup adm
45 $FileCreateMode 0640
46 $DirCreateMode 0755
47 $UMask 0022
48 $PrivDropToUser syslog
49 $PrivDropToGroup syslog
50
51 #
52 # Where to place spool and state files
53 #
54 $WorkDirectory /var/spool/rsyslog
55
56 # Send logs to remote Syslog server
57 *.* @172.16.82.68:514 # Используйте @ для UDP
58 *.* @@172.16.82.68:514 # Используйте @@ для TCP
59
60 # Filter SSHD logs to a separate file
61 if $programname == 'sshd' then /var/log/sshd.log
62 if $programname == 'sshd' then ~ # Не записывать эти сообщения больше нигде
63
64 # Filter error messages to a separate file
65 if $syslogseverity-text == 'error' then /var/log/error.log
66
67 # Log various facilities to specific files
68 authpriv.* /var/log/auth.log # Все сообщения авторизации
69 mail.* /var/log/mail.log # Все сообщения, связанные с почтой
70 *.info;mail.none;authpriv.none;cron.none /var/log/messages # Остальные сообщения
71
72
73 #
74 # Include all config files in /etc/rsyslog.d/
75 #
76 $IncludeConfig /etc/rsyslog.d/*.conf

```

Рис. 7.13: Файл /etc/rsyslog.conf после редактирования(2)

После внесения изменений файла */etc/rsyslog.conf* перезапустим службу rsyslog и снова проверим её статус

```

root@ubuntu-katerok:~# sudo systemctl restart rsyslog
root@ubuntu-katerok:~#

```

Рис. 7.14: Перезапуск rsyslog

```

root@ubuntu-katerok:~# systemctl status rsyslog
● rsyslog.service - System Logging Service
   Loaded: loaded (/lib/systemd/system/rsyslog.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-14 15:59:15 MSK; 14s ago
 TriggeredBy: ● syslog.socket
    Docs: man:rsyslogd(8)
          man:rsyslog.conf(5)
          https://www.rsyslog.com/doc/
 Main PID: 12900 (rsyslogd)
    Tasks: 10 (limit: 18904)
  Memory: 1.5M
     CPU: 12ms
   CGroup: /system.slice/rsyslog.service
           └─12900 /usr/sbin/rsyslogd -n -iNONE

окт 14 15:59:15 ubuntu-katerok systemd[1]: Starting System Logging Service...
окт 14 15:59:15 ubuntu-katerok rsyslogd[12900]: warning: ~ action is deprecated, consider using the 'stop' statement
окт 14 15:59:15 ubuntu-katerok systemd[1]: Started System Logging Service.
окт 14 15:59:15 ubuntu-katerok rsyslogd[12900]: rsyslogd's groupid changed to 111
окт 14 15:59:15 ubuntu-katerok rsyslogd[12900]: rsyslogd's userid changed to 104
окт 14 15:59:15 ubuntu-katerok rsyslogd[12900]: [origin software="rsyslogd" swVersion="8.2112.0" x-pid="12900" x-inp=
lines 1-21/21 (END)

```

Рис. 7.15: Проверка статуса rsyslog после редактирования файла

7.2 Проверка настроек Syslog

Проверим, работают ли настройки Syslog. Для этого сначала создадим тестовое сообщение об ошибке с помощью logger. Теперь мы видим, что создан файл `/var/log/error.log` куда записываются все сообщения с уровнем серьёзности “error”

```

root@ubuntu-katerok:~# logger -p user.error "Это тестирует сообщение об ошибке."
root@ubuntu-katerok:~# cat /var/log/error.log
Oct 14 16:00:21 ubuntu-katerok root: Это тестирует сообщение об ошибке.
root@ubuntu-katerok:~#

```

Рис. 7.16: Файл `/var/log/error.log`

Также проверим файл `/var/log/sshd.log`. Для этого выполним событие, которое будет записано в логи sshd. Например, выполним вход на сервер с помощью SSH: `ssh наш_пользователь@наш_сервер`

```

root@ubuntu-katerok:~# ssh eavernikovskaya@172.16.82.85
eavernikovskaya@172.16.82.85's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Расширенное поддержание безопасности (ESM) для Applications выключено.

219 обновлений может быть применено немедленно.
4 из этих обновлений, являются стандартными обновлениями безопасности.
Чтобы просмотреть дополнительные обновления выполните: apt list --upgradable

41 дополнительное обновление безопасности может быть применено с помощью ESM Apps.
Подробнее о включении службы ESM Apps at https://ubuntu.com/esm

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

2 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Mon Oct 14 15:25:38 2024 from 172.16.82.85
eavernikovskaya@ubuntu-katerok:~$

```

Рис. 7.17: Вход на сервер с помощью SSH

После мы видим, что создан файл `/var/log/sshd.log`, в который записываются все логи, полученные от процесса `sshd` (сервер SSH)

```

eavernikovskaya@ubuntu-katerok:~$ cat /var/log/sshd.log
Oct 14 16:01:47 ubuntu-katerok sshd[13141]: Accepted password for eavernikovskaya from 172.16.82.85 port 60156 ssh2
Oct 14 16:01:47 ubuntu-katerok sshd[13141]: pam_unix(sshd:session): session opened for user eavernikovskaya(uid=1000) by (uid=0)
eavernikovskaya@ubuntu-katerok:~$

```

Рис. 7.18: Файл `/var/log/sshd.log`

Далее проверяем всё остальное

```

eavernikovskaya@ubuntu-katerok:~$ cat /var/log/messages
Oct 14 15:59:15 ubuntu-katerok systemd[1]: Stopping System Logging Service...
Oct 14 15:59:15 ubuntu-katerok rsyslogd: [origin software="rsyslogd" swVersion="8.2112.0" x-pid="11359" x-info="https://www.rsyslog.com"] exiting on signal 15.
Oct 14 15:59:15 ubuntu-katerok systemd[1]: rsyslog.service: Deactivated successfully.
Oct 14 15:59:15 ubuntu-katerok systemd[1]: Stopped System Logging Service.
Oct 14 15:59:15 ubuntu-katerok systemd[1]: Starting System Logging Service...

```

Рис. 7.19: Файл `/var/log/messages.log`

```
eavernikovskaya@ubuntu-katerok:~$ cat /var/log/auth.log
Oct 14 15:59:15 ubuntu-katerok sudo: pam_unix(sudo:session): session closed for user root
Oct 14 15:59:15 ubuntu-katerok sudo: pam_unix(sudo:session): session closed for user root
Oct 14 16:01:47 ubuntu-katerok systemd-logind[1007]: New session 8 of user eavernikovskaya.
Oct 14 16:02:56 ubuntu-katerok sudo: pam_unix(sudo:auth): authentication failure; logname=eavernikovskaya uid=1000 e
uid=0 tty=/dev/pts/3 ruser=eavernikovskaya rhost= user=eavernikovskaya
Oct 14 16:02:56 ubuntu-katerok sudo: pam_unix(sudo:auth): authentication failure; logname=eavernikovskaya uid=1000 e
uid=0 tty=/dev/pts/3 ruser=eavernikovskaya rhost= user=eavernikovskaya
Oct 14 16:03:05 ubuntu-katerok sudo: eavernikovskaya : 3 incorrect password attempts ; TTY=pts/3 ; PWD=/home/eaverni
kovskaya ; USER=root ; COMMAND=/usr/bin/rm /var/log/messages
Oct 14 16:03:05 ubuntu-katerok sudo: eavernikovskaya : 3 incorrect password attempts ; TTY=pts/3 ; PWD=/home/eaverni
kovskaya ; USER=root ; COMMAND=/usr/bin/rm /var/log/messages
eavernikovskaya@ubuntu-katerok:~$
```

Рис. 7.20: Файл /var/log/auth.log

```
eavernikovskaya@ubuntu-katerok:~$ cat /var/log/mail.log
Oct 14 16:06:17 ubuntu-katerok eavernikovskaya: Тестовое сообщение для почты
Oct 14 16:06:17 ubuntu-katerok eavernikovskaya: Тестовое сообщение для почты
eavernikovskaya@ubuntu-katerok:~$
```

Рис. 7.21: Файл /var/log/mail.log

7.3 Практические примеры поиска ошибок и предупреждений

После настройки Syslog, важно уметь анализировать журналы для нахождения ошибок и предупреждений. Найдём с помощью грег ошибки и предупреждения

```
eavernikovskaya@ubuntu-katerok:~$ grep -t "error" /var/log/syslog
Oct 14 15:51:36 ubuntu-katerok kernel: [ 3407.396294] ACPI Error: No handler for Region [VRTC] (00000000af667901) [S
ystemCMOS] (20210730/evregion-130)
Oct 14 15:51:36 ubuntu-katerok kernel: [ 3407.396321] ACPI Error: Region SystemCMOS (ID=5) has no handler (20210730/
exfldio-261)
Oct 14 15:51:36 ubuntu-katerok kernel: [ 3407.396367] ACPI Error: Aborting method \_SB.PCI0.LPCB.EC._Q9A due to prev
ious error (AE_NOT_EXIST) (20210730/psparse-529)
Oct 14 15:55:32 ubuntu-katerok org.freedesktop.impl.portal.desktop.kde[11883]: xdp-kde-inhibit: Inhibition error: "
The name org.kde.Solid.PowerManagement was not provided by any .service files"
Oct 14 15:57:10 ubuntu-katerok kglobalaccel5[12579]: kf.kio.gui: Failed to register new cgroup: "app-\x2fusr\x2fli
b\x2fqt5\x2fbins\x2fdbus-14906b824c9e4a65ac5bb5f18ccc1b.scope" "org.freedesktop.DBus.Error.UnixProcessIdUnknown
" "Process with ID 12582 does not exist."
Oct 14 15:57:30 ubuntu-katerok kglobalaccel5[12630]: kf.kio.gui: Failed to register new cgroup: "app-\x2fusr\x2fli
b\x2fqt5\x2fbins\x2fdbus-82d269ca7e2c494696931468e45e18d4.scope" "org.freedesktop.DBus.Error.UnixProcessIdUnknown
" "Process with ID 12633 does not exist."
Oct 14 16:52:41 ubuntu-katerok kernel: [ 7071.702652] ACPI Error: No handler for Region [VRTC] (00000000af667901) [S
ystemCMOS] (20210730/evregion-130)
Oct 14 16:52:41 ubuntu-katerok kernel: [ 7071.702679] ACPI Error: Region SystemCMOS (ID=5) has no handler (20210730/
exfldio-261)
Oct 14 16:52:41 ubuntu-katerok kernel: [ 7071.702723] ACPI Error: Aborting method \_SB.PCI0.LPCB.EC._Q9A due to prev
ious error (AE_NOT_EXIST) (20210730/psparse-529)
```

Рис. 7.22: Поиск ошибок

```
eavernikovskaya@ubuntu-katerok:~$ grep -i "warning" /var/log/syslog
Oct 14 15:59:15 ubuntu-katerok rsyslogd: warning: ~ action is deprecated, consider using the 'stop' statement instead [v8.2112.0 try https://www.rsyslog.com/e/2307 ]
Oct 14 16:09:29 ubuntu-katerok kernel: [ 4480.209986] [WARNING][BB][halbb_fw_set_reg] IO offload fail: 1
Oct 14 16:42:07 ubuntu-katerok kernel: [ 6437.946601] [WARNING][BB][halbb_fw_set_reg] IO offload fail: 1
Oct 14 16:43:26 ubuntu-katerok kernel: [ 6517.177310] [WARNING][BB][halbb_fw_set_reg] IO offload fail: 1
Oct 14 16:44:01 ubuntu-katerok kernel: [ 6551.739265] [WARNING][BB][halbb_fw_set_reg] IO offload fail: 1
```

Рис. 7.23: Поиск предупреждений

8 Выводы

Таким образом, Syslog — это важный инструмент для централизованного сбора, хранения и анализа системных событий в Linux и других операционных системах. Благодаря гибкой архитектуре, он позволяет настраивать как локальное, так и удаленное логирование, что облегчает мониторинг и управление системами в распределенных инфраструктурах. Инструменты, такие как *rsyslog* и *syslog-ng*, предоставляют дополнительные возможности для фильтрации и маршрутизации данных.

Важным аспектом использования Syslog является правильная конфигурация системы для обеспечения безопасности и производительности. Это позволяет администраторам эффективно отслеживать незапланированные события, выявлять угрозы и устранять сбои в работе системы. Настроенная система логирования помогает анализировать состояние серверов, повышая их надежность и оптимизируя работу инфраструктуры.

Кроме того, современные инструменты для анализа логов дают возможность предсказывать возможные сбои и улучшать общую безопасность. Использование Syslog в сочетании с мощными инструментами мониторинга и анализа помогает повысить защищенность инфраструктуры, а также оптимизировать её работу, что делает его незаменимым элементом в современной IT-среде.

Список литературы

1. Система Syslog и журналы логов в Linux. ИТ Проффи, 2023.
2. synergix. Контора пишет - syslog. unix.uz, 2010.
3. Differences between TCP and UDP. GeeksforGeeks, 2024.
4. What is syslog? sumo logic.
5. cryptoparty. Syslog : Полное руководство. 2022.
6. Syslog. sematext.
7. Просмотр и настройка логов Linux на Ubuntu, Debian и CentOS. beget.