РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук
Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

Управление журналами событий в системе

Студент: Эйвази Мани

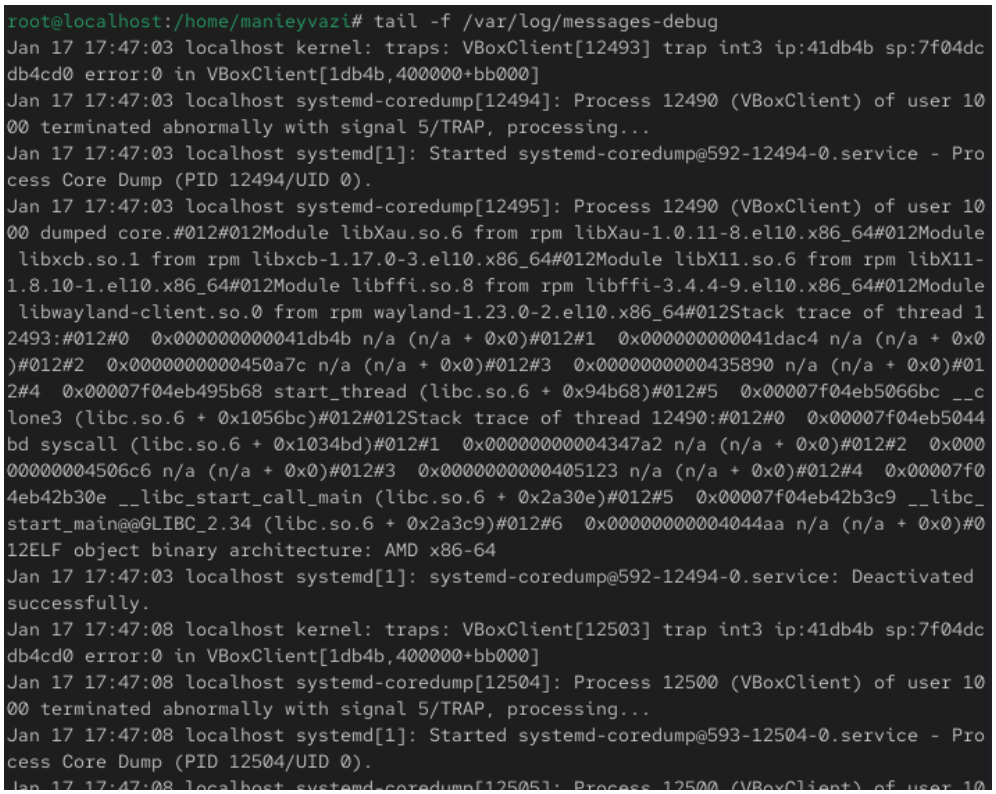
Группа: НПИбд-03-24

Студенческий билет №: 1032245107

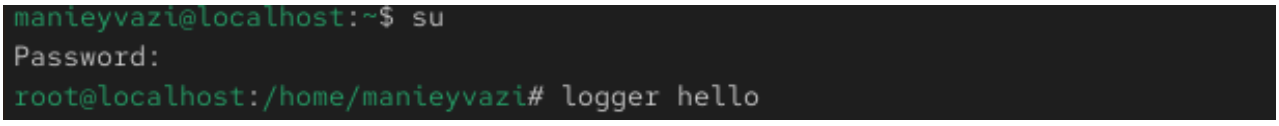
Цель работы

Получить практические навыки работы с системой журналирования событий в Linux. Освоить методы мониторинга логов в реальном времени, настройки маршрутизации сообщений через rsyslog, а также использования современных инструментов journalctl и journald для просмотра и управления журналами на основе systemd.

Первый шаг: Запуск трех терминалов, инициализация мониторинга основного лог-файла /var/log/messages. Команда tail-f начала вывод последних строк файла и продолжила отображать новые записи в реальном времени.



Второй шаг: Попытка неудачной авторизации su и отправка произвольного сообщения через logger. В окне мониторинга (tail-f) немедленно появились соответствующие записи: сообщение о неудачном su (появится в /var/log/secure или /var/log/messages, в зависимости от конфигурации) и строка с сообщением "hello" от пользователя.

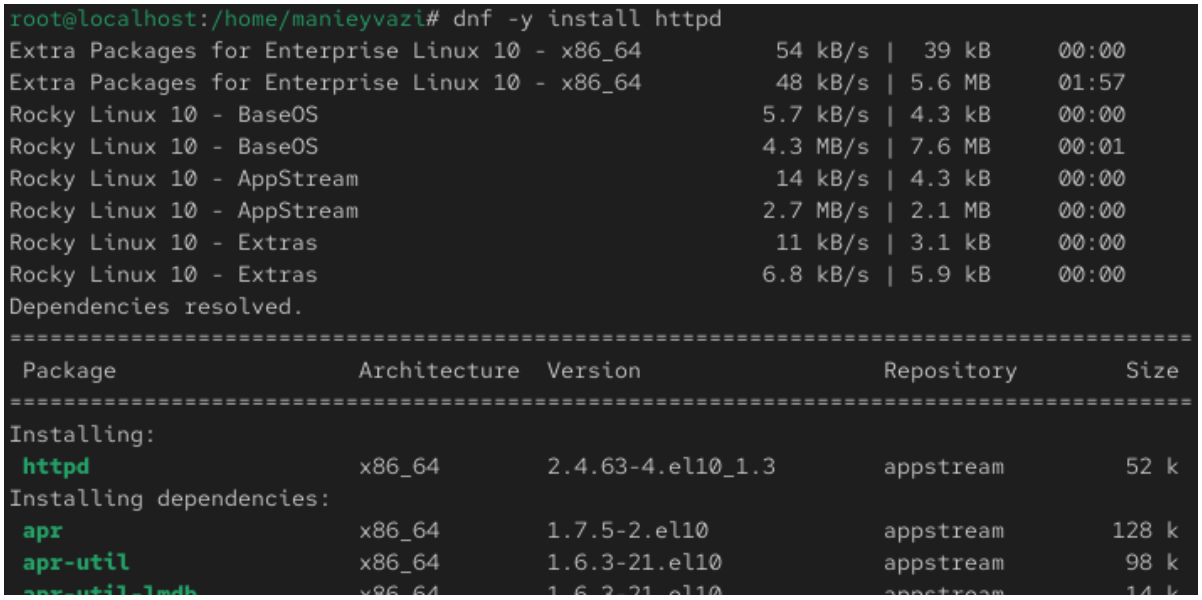


Третий шаг: Остановка предыдущего мониторинга и просмотр журнала аутентификации.

- Команда: tail-n 20 /var/log/secure

В выводе команды была обнаружена запись о неудачной попытке входа через su, что подтвердило запись событий безопасности в отдельный файл.

Четвертый шаг: Установка веб-сервера Apache и его активация. Веб-сервер Apache установлен и запущен. Логи по умолчанию пишутся в /var/log/httpd/.



systemctl start httpd

systemctl enable httpd

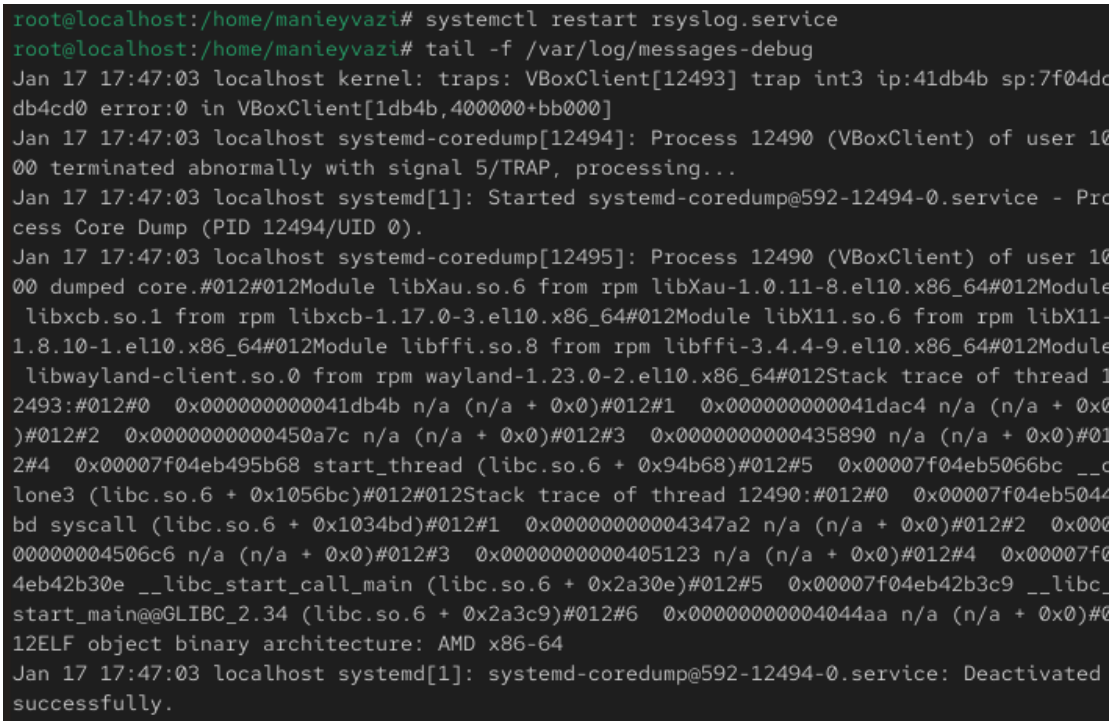
Пятый шаг: Изменение конфигурации Apache для отправки логов ошибок в системный логер (syslog), а не в отдельный файл. Apache сконфигурирован на использование средства local1 syslog для ошибок.

ErrorLog syslog:local1

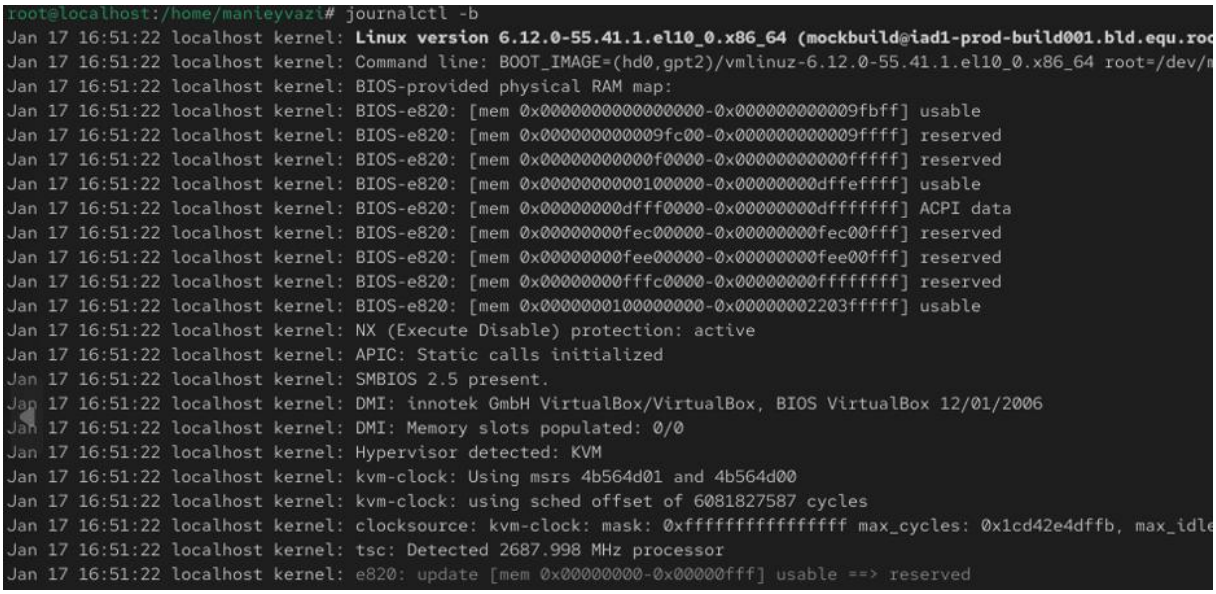
Шестой шаг: Настройка rsyslog для приема сообщений от средства local1 и записи их в выделенный файл. После перезапуска служб, ошибки Apache стали записываться в файл /var/log/httpd-error.log. Мониторинг командой tail-f /var/log/httpd-error.log подтвердил поступление записей.

```
root@localhost:/home/manieyvazi# systemctl restart rsyslog.service
root@localhost:/home/manieyvazi# systemctl restart httpd
```

Седьмой шаг: Настройка записи всех отладочных сообщений системы в отдельный файл. В файле /var/log/messages-debug появилось тестовое отладочное сообщение, что подтвердило корректность работы нового правила.



Восьмой шаг: Просмотр всего журнала с момента последней загрузки. Открылся постраничный просмотр всех записей журнала systemd.



Девятый шаг: Использование различных флагов для фильтрации вывода. Успешно опробованы ключевые параметры для фильтрации журнала по времени, приоритету и источнику (юниту). Вывод стал целенаправленным и удобным для анализа.

```
journalctl -f                  # Режим реального времени (аналог tail -f)
journalctl -n 20               # Последние 20 записей
journalctl -p err              # Только сообщения об ошибках (уровень err)
journalctl --since "2024-01-01" --until "2024-01-02" # За указанный период
journalctl _SYSTEMD_UNIT=sshd.service # Сообщения конкретной службы
```

десятый шаг: По умолчанию journald хранит логи только в оперативной памяти (/run/log/journal/). Необходимо создать директорию для их сохранения на диск. Создана директория /var/log/journal с правильными правами для службы systemd-journald.

```
manieyvazi@localhost:~$ su
Password:
root@localhost:/home/manieyvazi# mkdir -p /var/log/journal
root@localhost:/home/manieyvazi# chown root:systemd-journal /var/log/journal
root@localhost:/home/manieyvazi# chmod 2755 /var/log/journal
```

Одиннадцатый шаг: Перезапуск службы journald с сигналом для перечитывания конфигурации. После отправки сигнала USR1 служба systemd-journald начала сохранять журналы в новую директорию на диске.
Команда journalctl -b (просмотр журнала с последней загрузки) теперь будет работать и после перезагрузки системы.

```
root@localhost:/home/manieyvazi# killall -USR1 systemd-journald
root@localhost:/home/manieyvazi# journalctl -b
Jan 17 16:51:22 localhost kernel: Linux version 6.12.0-55.41.1.el10_0.x86_64 (mockbuild@iad1-prod-build001.bld.equ.roc>
Jan 17 16:51:22 localhost kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.41.1.el10_0.x86_64 root=/dev/m>
Jan 17 16:51:22 localhost kernel: BIOS-provided physical RAM map:
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000dffeffff] usable
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000dfff0000-0x00000000dfffffff] ACPI data
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000100000000-0x00000002203fffff] usable
Jan 17 16:51:22 localhost kernel: NX (Execute Disable) protection: active
Jan 17 16:51:22 localhost kernel: APIC: Static calls initialized
Jan 17 16:51:22 localhost kernel: SMBIOS 2.5 present.
Jan 17 16:51:22 localhost kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Jan 17 16:51:22 localhost kernel: DMI: Memory slots populated: 0/0
Jan 17 16:51:22 localhost kernel: Hypervisor detected: KVM
Jan 17 16:51:22 localhost kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
Jan 17 16:51:22 localhost kernel: kvm-clock: using sched offset of 6081827587 cycles
Jan 17 16:51:22 localhost kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle>
Jan 17 16:51:22 localhost kernel: tsc: Detected 2687.998 MHz processor
Jan 17 16:51:22 localhost kernel: e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
Jan 17 16:51:22 localhost kernel: e820: remove [mem 0x000a0000-0x000fffff] usable
Jan 17 16:51:22 localhost kernel: last_pfn = 0x220400 max_arch_pfn = 0x400000000
Jan 17 16:51:22 localhost kernel: MTRR map: 3 entries (3 fixed + 0 variable; max 19), built from 8 variable MTRRs
Jan 17 16:51:22 localhost kernel: x86/PAT: Configuration [0-7]: WB  WC  UC- UC  WB  WP  UC- WT
Jan 17 16:51:22 localhost kernel: CPU MTRRs all blank - virtualized system.
Jan 17 16:51:22 localhost kernel: last_pfn = 0xe0000 max_arch_pfn = 0x400000000
```

заключение

В ходе лабораторной работы были успешно освоены современные и классические методы работы с системой журналирования в Linux:

1. **Классическое журналирование (**rsyslog**):** Получены навыки настройки маршрутизации лог-сообщений на основе их источника (facility) и приоритета (priority). Практически настроено перенаправление логов веб-сервера Apache из его собственных файлов в централизованную систему rsyslog с записью в выделенный файл. Это повышает удобство централизованного сбора и анализа логов.
2. **Мониторинг в реальном времени:** Освоено использование утилиты tail -f для "слежения" за лог-файлами, что является ключевым навыком для оперативного реагирования на события в системе.
3. **Работа с** systemd-journald**:** Изучена мощная утилита journalctl для гибкого и структурированного просмотра журналов. Применены фильтры по времени, приоритету, системному юниту, что делает поиск нужных событий быстрым и эффективным.
4. **Организация постоянного хранения журналов:** Практически настроено постоянное хранение журналов journald на диске (в /var/log/journal/), что обеспечивает сохранность логов между перезагрузками системы, что критично для аудита и расследования инцидентов.
5. **Гибридный подход:** Продемонстрировано понимание сосуществования двух систем журналирования: классической (rsyslog) для совместимости и гибкой фильтрации, и современной (journald) для тесной интеграции с systemd и богатыми метаданными.

Работа показала, что грамотное владение инструментами журналирования является неотъемлемой частью компетенций системного администратора для обеспечения наблюдаемости, безопасности и стабильности работы серверов.

1. Какой файл используется для настройки rsyslogd?

   Основной файл: /etc/rsyslog.conf. Дополнительные конфигурации можно размещать в /etc/rsyslog.d/*.conf.

2. В каком файле журнала rsyslogd содержатся сообщения, связанные с аутентификацией?

   По умолчанию — /var/log/secure (или /var/log/auth.log в некоторых дистрибутивах).

3. Если вы ничего не настроите, то сколько времени потребуется для ротации файлов журналов?

   Ротацией логов обычно управляет logrotate. Стандартный конфигурационный файл /etc/logrotate.conf и скрипты в /etc/logrotate.d/ определяют периодичность (часто — еженедельно) и условия (размер, время).

4. Какую строку следует добавить в конфигурацию для записи всех сообщений с приоритетом info в файл /var/log/messages_info?

   *.info /var/log/messages_info (в файл /etc/rsyslog.conf или в /etc/rsyslog.d/my.conf).

5. Какая команда позволяет вам видеть сообщения журнала в режиме реального времени?

   Для текстовых файлов: tail-f /var/log/<имя_файла>. Для журнала systemd: journalctl-f.

6. Какая команда позволяет вам видеть все сообщения журнала, которые были написаны для PID 1 между 9:00 и 15:00?

   journalctl _PID=1--since "09:00"--until "15:00"

7. Какая команда позволяет вам видеть сообщения journald после последней перезагрузки системы?

   journalctl-b или journalctl-b 0 (для текущей загрузки). journalctl-b-1 — для предыдущей.

8. Какая процедура позволяет сделать журнал journald постоянным?

Создать директорию: mkdir-p /var/log/journal.

Назначить права: chown root:systemd-journal /var/log/journal && chmod 2755 /var/log/journal.

Отправить сигнал службе: killall-USR1 systemd-journald или перезагрузить систему.