



## 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.

```
root@localhost:/home/manieyvazi# tail -f /var/log/messages-debug
Jan 17 17:47:03 localhost kernel: traps: VBoxClient[12493] trap int3 ip:41db4b sp:7f04dc
db4cd0 error:0 in VBoxClient[1db4b,400000+bb000]
Jan 17 17:47:03 localhost systemd-coredump[12494]: Process 12490 (VBoxClient) of user 10
00 terminated abnormally with signal 5/TRAP, processing...
Jan 17 17:47:03 localhost systemd[1]: Started systemd-coredump@592-12494-0.service - Pro
cess Core Dump (PID 12494/UID 0).
Jan 17 17:47:03 localhost systemd-coredump[12495]: Process 12490 (VBoxClient) of user 10
00 dumped core.#012#012Module libXau.so.6 from rpm libXau-1.0.11-8.el10.x86_64#012Module
libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64#012Module libX11.so.6 from rpm libX11-
1.8.10-1.el10.x86_64#012Module libffi.so.8 from rpm libffi-3.4.4-9.el10.x86_64#012Module
libwayland-client.so.0 from rpm wayland-1.23.0-2.el10.x86_64#012Stack trace of thread 1
2493:#012#0 0x000000000041db4b n/a (n/a + 0x0)#012#1 0x000000000041dac4 n/a (n/a + 0x0
)#012#2 0x0000000000450a7c n/a (n/a + 0x0)#012#3 0x0000000000435890 n/a (n/a + 0x0)#01
2#4 0x00007f04eb495b68 start_thread (libc.so.6 + 0x94b68)#012#5 0x00007f04eb5066bc __c
lone3 (libc.so.6 + 0x1056bc)#012#012Stack trace of thread 12490:#012#0 0x00007f04eb5044
bd syscall (libc.so.6 + 0x1034bd)#012#1 0x00000000004347a2 n/a (n/a + 0x0)#012#2 0x000
0000004506c6 n/a (n/a + 0x0)#012#3 0x0000000000405123 n/a (n/a + 0x0)#012#4 0x00007f0
4eb42b30e __libc_start_call_main (libc.so.6 + 0x2a30e)#012#5 0x00007f04eb42b3c9 __libc_
start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6 0x00000000004044aa n/a (n/a + 0x0)#0
12ELF object binary architecture: AMD x86-64
Jan 17 17:47:03 localhost systemd[1]: systemd-coredump@592-12494-0.service: Deactivated
successfully.
Jan 17 17:47:08 localhost kernel: traps: VBoxClient[12503] trap int3 ip:41db4b sp:7f04dc
db4cd0 error:0 in VBoxClient[1db4b,400000+bb000]
Jan 17 17:47:08 localhost systemd-coredump[12504]: Process 12500 (VBoxClient) of user 10
00 terminated abnormally with signal 5/TRAP, processing...
Jan 17 17:47:08 localhost systemd[1]: Started systemd-coredump@593-12504-0.service - Pro
cess Core Dump (PID 12504/UID 0).
Jan 17 17:47:08 localhost systemd-coredump[12505]: Process 12500 (VBoxClient) of user 10
```

**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.

```
manieyvazi@localhost:~$ su
Password:
root@localhost:/home/manieyvazi# logger hello
```

**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/`.

```
root@localhost:/home/manieyvazi# dnf -y install httpd
Extra Packages for Enterprise Linux 10 - x86_64      54 kB/s | 39 kB      00:00
Extra Packages for Enterprise Linux 10 - x86_64      48 kB/s | 5.6 MB     01:57
Rocky Linux 10 - BaseOS                             5.7 kB/s | 4.3 kB     00:00
Rocky Linux 10 - BaseOS                             4.3 MB/s | 7.6 MB     00:01
Rocky Linux 10 - AppStream                          14 kB/s | 4.3 kB     00:00
Rocky Linux 10 - AppStream                          2.7 MB/s | 2.1 MB     00:00
Rocky Linux 10 - Extras                             11 kB/s | 3.1 kB     00:00
Rocky Linux 10 - Extras                             6.8 kB/s | 5.9 kB     00:00
Dependencies resolved.
=====
Package           Architecture Version           Repository      Size
=====
Installing:
  httpd            x86_64        2.4.63-4.el10_1.3 appstream       52 k
Installing dependencies:
  apr              x86_64        1.7.5-2.el10     appstream       128 k
  apr-util         x86_64        1.6.3-21.el10    appstream       98 k
  apr-util-1mdb    x86_64        1.6.3-21.el10    appstream       14 k
```

`systemctl start httpd`

`systemctl enable httpd`

**Пятый шаг:** Изменение конфигурации Apache для отправки логов ошибок в системный логгер (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.

```
root@localhost:/home/manieyvazi# systemctl restart rsyslog.service
root@localhost:/home/manieyvazi# tail -f /var/log/messages-debug
Jan 17 17:47:03 localhost kernel: traps: VBoxClient[12493] trap int3 ip:41db4b sp:7f04dc
db4cd0 error:0 in VBoxClient[1db4b,400000+bb000]
Jan 17 17:47:03 localhost systemd-coredump[12494]: Process 12490 (VBoxClient) of user 10
00 terminated abnormally with signal 5/TRAP, processing...
Jan 17 17:47:03 localhost systemd[1]: Started systemd-coredump@592-12494-0.service - Pro
cess Core Dump (PID 12494/UID 0).
Jan 17 17:47:03 localhost systemd-coredump[12495]: Process 12490 (VBoxClient) of user 10
00 dumped core.#012#012Module libXau.so.6 from rpm libXau-1.0.11-8.el10.x86_64#012Module
libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64#012Module libX11.so.6 from rpm libX11-
1.8.10-1.el10.x86_64#012Module libffi.so.8 from rpm libffi-3.4.4-9.el10.x86_64#012Module
libwayland-client.so.0 from rpm wayland-1.23.0-2.el10.x86_64#012Stack trace of thread 1
2493:#012#0 0x000000000041db4b n/a (n/a + 0x0)#012#1 0x000000000041dac4 n/a (n/a + 0x0)
#012#2 0x0000000000450a7c n/a (n/a + 0x0)#012#3 0x0000000000435890 n/a (n/a + 0x0)#01
2#4 0x00007f04eb495b68 start_thread (libc.so.6 + 0x94b68)#012#5 0x00007f04eb5066bc __c
lone3 (libc.so.6 + 0x1056bc)#012#012Stack trace of thread 12490:#012#0 0x00007f04eb5044
bd syscall (libc.so.6 + 0x1034bd)#012#1 0x00000000004347a2 n/a (n/a + 0x0)#012#2 0x000
00000004506c6 n/a (n/a + 0x0)#012#3 0x0000000000405123 n/a (n/a + 0x0)#012#4 0x00007f0
4eb42b30e __libc_start_call_main (libc.so.6 + 0x2a30e)#012#5 0x00007f04eb42b3c9 __libc_
start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6 0x00000000004044aa n/a (n/a + 0x0)#0
12ELF object binary architecture: AMD x86-64
Jan 17 17:47:03 localhost systemd[1]: systemd-coredump@592-12494-0.service: Deactivated
successfully.
```

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

```
root@localhost:/home/manleyvazi# 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.ro
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/n
Jan 17 16:51:22 localhost kernel: BIOS-provided physical RAM map:
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbfff] usable
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009fffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000000100000-0x000000000000dfffffff] usable
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000000dfff0000-0x00000000000dffffffffff] ACPI data
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffffffff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x00000000100000000-0x000000002203fffff] 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 0x00000000-0x00000ffff] usable ==> reserved
```

**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.rocks)
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/mapper:
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 0x0000000000100000-0x0000000000dfffff] usable
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000000dffff000-0x0000000000dfffff] ACPI data
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x000000000fec0000-0x000000000fec00ff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x000000000fee0000-0x000000000fee00ff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x000000000fffc000-0x000000000fffcfff] reserved
Jan 17 16:51:22 localhost kernel: BIOS-e820: [mem 0x0000000010000000-0x000000002203ffff] 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 0x00000000-0x000000ff] usable ==> reserved
Jan 17 16:51:22 localhost kernel: e820: remove [mem 0x000a0000-0x0000ffff] usable
Jan 17 16:51:22 localhost kernel: last_pfn = 0x220400 max_arch_pfn = 0x40000000
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 = 0xe000 max_arch_pfn = 0x40000000
```



## закключение

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.