Отчёт по лабораторной работе №7

Управление журналами событий в системе

Эзиз Хатамов

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Получить навыки работы с журналами мониторинга различных событий в системе.

2 Отчёт по выполнению работы

2.1 Мониторинг журнала системных событий в реальном времени

1. В трёх вкладках терминала были получены права администратора с помощью команды su -.
   После успешного ввода пароля был осуществлён вход под пользователем root.

2. Во второй вкладке был запущен мониторинг системных событий в реальном времени с помощью команды tail -f /var/log/messages.
   На экране начали отображаться текущие системные сообщения, включая информацию о работе служб и процессах пользователя.

```
root@ehatamov:/home/ehatamov# tail -f /var/log/messages
Oct  2 17:03:43 ehatamov systemd[1992]: Starting gvfs-metadata.service - Virtual filesystem metadata service...
Oct  2 17:03:43 ehatamov systemd[1992]: Started gvfs-metadata.service - Virtual filesystem metadata service.
Oct  2 17:03:44 ehatamov PackageKit[1407]: uid 1000 is trying to obtain org.freedesktop.packagekit.system-sources-refresh auth (onl
y_trusted:0)
Oct  2 17:03:44 ehatamov PackageKit[1407]: uid 1000 obtained auth for org.freedesktop.packagekit.system-sources-refresh
Oct  2 17:03:44 ehatamov systemd[1]: fprintd.service: Deactivated successfully.
Oct  2 17:03:45 ehatamov kernel: traps: VBoxClient[3408] trap int3 ip:41dd1b sp:7fdadef4fcd0 error:0 in VBoxClient[1dd1b,400000+bb0
00]
Oct  2 17:03:45 ehatamov systemd-coredump[3409]: Process 3405 (VBoxClient) of user 1000 terminated abnormally with signal 5/TRAP, p
rocessing...
Oct  2 17:03:45 ehatamov systemd[1]: Started systemd-coredump@12-3409-0.service - Process Core Dump (PID 3409/UID 0).
Oct  2 17:03:45 ehatamov systemd-coredump[3410]: Process 3405 (VBoxClient) of user 1000 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-1.2
3.0-2.el10.x86_64#012Stack trace of thread 3408:#012#0  0x00000000041dd1b n/a (n/a + 0x0)#012#1  0x00000000041dc94 n/a (n/a + 0x0
)#012#2  0x0000000000045041c n/a (n/a + 0x0)#012#3  0x00000000004355d0 n/a (n/a + 0x0)#012#4  0x00007fdaed5f211a start_thread (libc.
so.6 + 0x9511a)#012#5  0x00007fdaed662c3c __clone3 (libc.so.6 + 0x105c3c)#012#012Stack trace of thread 3405:#012#0  0x00007fdaed660
a3d syscall (libc.so.6 + 0x103a3d)#012#1  0x00000000004344e2 n/a (n/a + 0x0)#012#2  0x0000000000450066 n/a (n/a + 0x0)#012#3  0x000
0000000405123 n/a (n/a + 0x0)#012#4  0x00007fdaed58730e __libc_start_call_main (libc.so.6 + 0x2a30e)#012#5  0x00007fdaed5873c9 __li
bc_start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6  0x00000000004044aa n/a (n/a + 0x0)#012ELF object binary architecture: AMD x8
6-64
Oct  2 17:03:45 ehatamov systemd[1]: systemd-coredump@12-3409-0.service: Deactivated successfully.
Oct  2 17:03:50 ehatamov kernel: traps: VBoxClient[3430] trap int3 ip:41dd1b sp:7fdadef4fcd0 error:0 in VBoxClient[1dd1b,400000+bb0
```

Рис. 2.1: Мониторинг системных событий через tail -f /var/log/messages

3. В третьей вкладке из оболочки пользователя была выполнена попытка по-

лучить права администратора с помощью команды su -, однако был введён неверный пароль.

В результате в окне мониторинга (во второй вкладке) отобразилось сообщение об ошибке:

FAILED SU (to root) ehatamov on pts/2

Это свидетельствует о неудачной попытке аутентификации при переходе в режим суперпользователя.

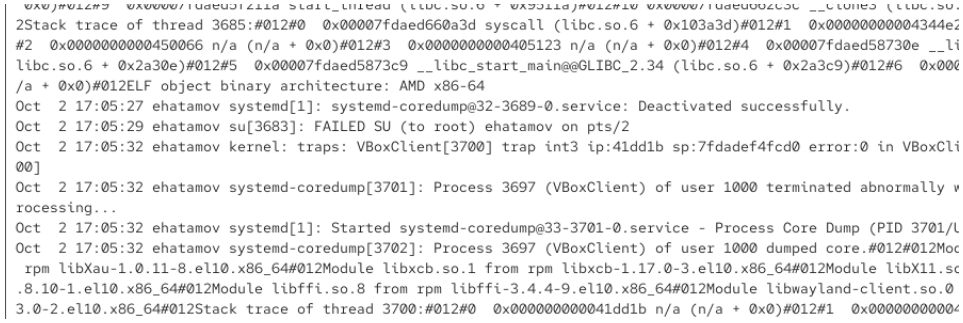

Рис. 2.2: Ошибка при попытке получить права root с неверным паролем

4. Затем пользователь ehatamov ввёл команду logger hello.

   После выполнения этой команды в терминале с мониторингом появилось сообщение ehatamov: hello, что подтверждает успешную запись пользовательского сообщения в системный журнал.

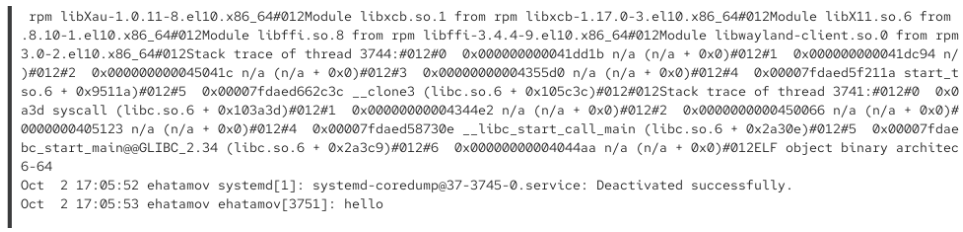

Рис. 2.3: Фиксация пользовательского сообщения в системном журнале через logger

5. Мониторинг системных сообщений был остановлен сочетанием клавиш Ctrl + C.

   Затем для анализа журнала безопасности были выведены последние 20

строк файла /var/log/secure с помощью команды tail -n 20 /var/log/secure.
В результате отобразились записи, подтверждающие предыдущие действия — успешные и неудачные попытки входа, а также обращения к системным службам.

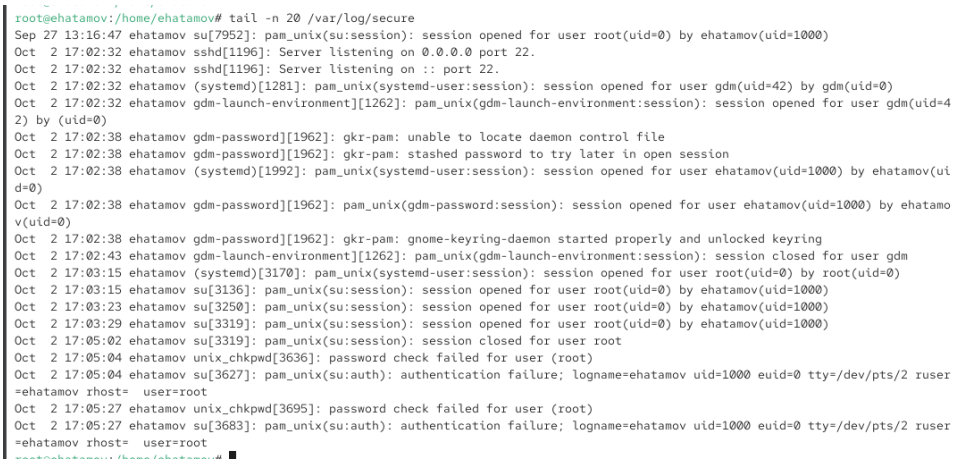

Рис. 2.4: Просмотр последних строк файла /var/log/secure

2.2 Изменение правил rsyslog.conf

1. В первой вкладке терминала был установлен веб-сервер Apache с помощью команды dnf -y install httpd.
   После завершения установки служба была запущена и добавлена в автозагрузку при помощи команд systemctl start httpd и systemctl enable httpd.
   Успешное выполнение команд подтвердило корректную установку и активацию сервиса.

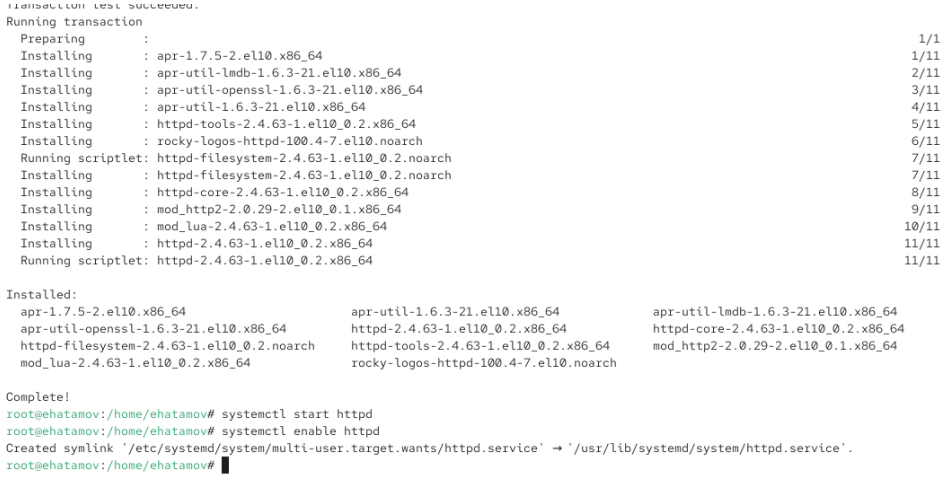

Рис. 2.5: Установка и запуск веб-сервера Apache

2. Во второй вкладке был запущен мониторинг журнала ошибок веб-службы Apache: tail -f /var/log/httpd/error_log.

   В журнале отобразились стандартные служебные сообщения о запуске Apache и активации модулей.

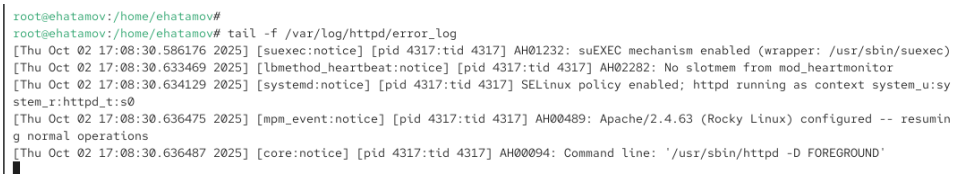

Рис. 2.6: Мониторинг журнала ошибок Apache

3. В третьей вкладке терминала в файле /etc/httpd/conf/httpd.conf была добавлена строка ErrorLog syslog:local1.

   Это позволило перенаправить сообщения об ошибках веб-службы в системный журнал через механизм syslog.

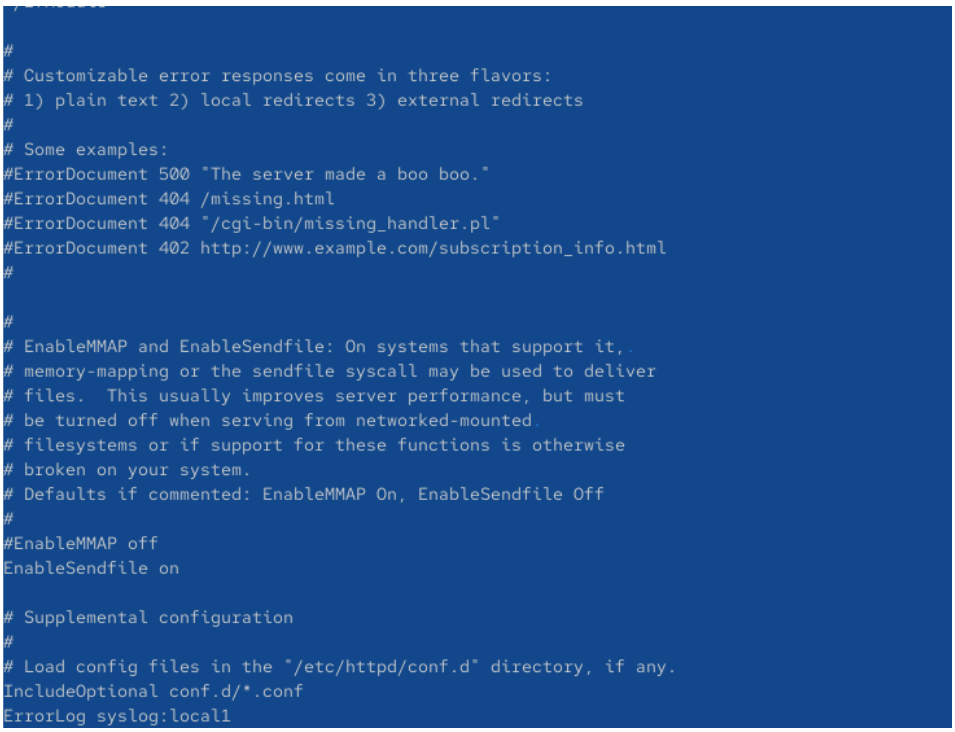

Рис. 2.7: Добавление строки ErrorLog syslog:local1 в конфигурацию Apache

4. Для настройки приёма сообщений от Apache в каталоге /etc/rsyslog.d был создан новый файл конфигурации httpd.conf.
   В него была добавлена строка local1.* -/var/log/httpd-error.log, направляющая все сообщения категории local1 в отдельный лог-файл /var/log/httpd-error.log.

Рис. 2.8: Создание конфигурационного файла httpd.conf в /etc/rsyslog.d

5. После внесения изменений служба rsyslog была перезапущена с помощью команды systemctl restart rsyslog.service для применения новых настроек.

6. Далее был создан отдельный файл конфигурации debug.conf для мониторинга отладочной информации.
   В этот файл была добавлена строка “*.debug /var/log/messages-debug”, направляющая все сообщения уровня debug в файл /var/log/messages-debug.
   После редактирования службы rsyslog была снова перезапущена.

```
root@ehatamov:/home/ehatamov# cd /etc/rsyslog.d/
root@ehatamov:/etc/rsyslog.d# touch httpd.conf
root@ehatamov:/etc/rsyslog.d# mcedit httpd.conf

root@ehatamov:/etc/rsyslog.d# touch debug.conf
root@ehatamov:/etc/rsyslog.d# echo "*.debug /var/log/messages-debug" > debug.conf
root@ehatamov:/etc/rsyslog.d# systemctl restart rsyslog.service
root@ehatamov:/etc/rsyslog.d#
```

Рис. 2.9: Создание файлов конфигурации httpd.conf и debug.conf, настройка вывода сообщений уровня debug

7. Для проверки работы механизма логирования был запущен мониторинг отладочного журнала командой tail -f /var/log/messages-debug.
   Затем было создано тестовое сообщение через команду logger -p

daemon.debug “Daemon debug message”.

В окне мониторинга появилось сообщение от root, подтверждающее успешную запись события в отладочный лог.

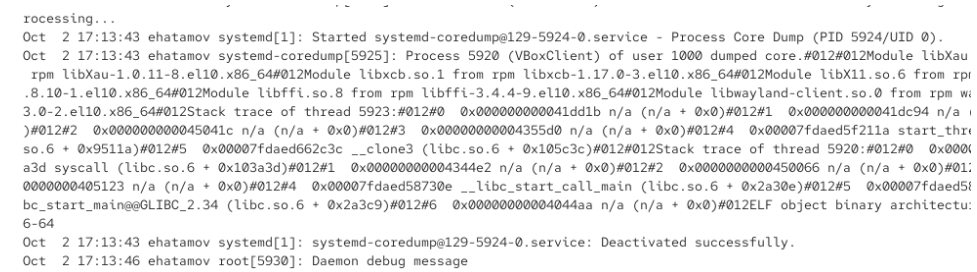

Рис. 2.10: Проверка работы отладочного логирования в messages-debug

2.3 Использование journalctl

1. Во второй вкладке терминала было выполнено отображение содержимого системного журнала с момента последнего запуска системы с помощью команды journalctl.

   В выводе отображались записи ядра и системных служб, включая данные о конфигурации виртуальной машины, аппаратных компонентах и запуске ядра.

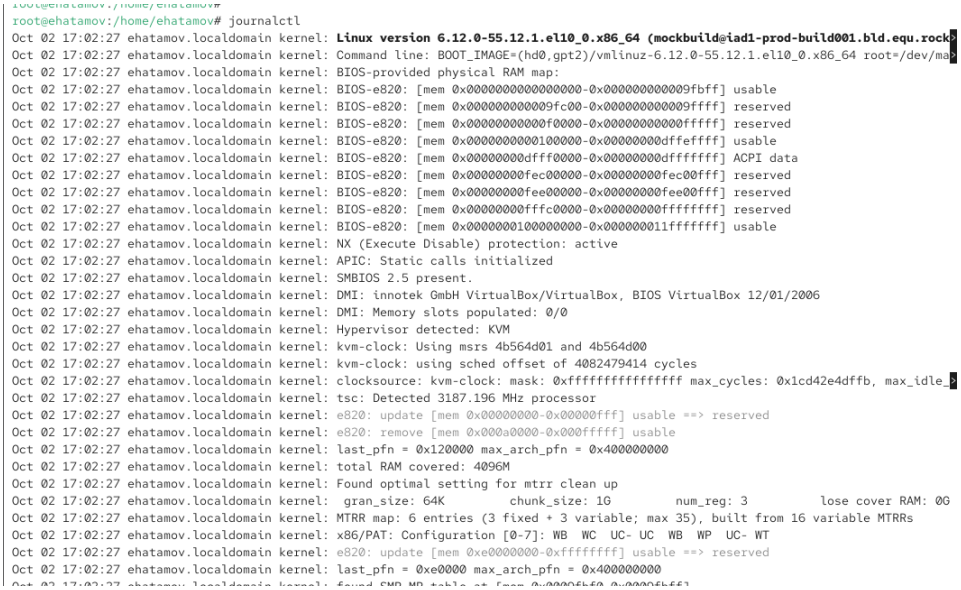

Рис. 2.11: Просмотр системного журнала с момента загрузки системы

2. Для демонстрации сообщений без постраничного просмотра использовалась команда journalctl –no-pager.

   В этом режиме весь вывод отображался сразу в окне терминала без ожидания действий пользователя.

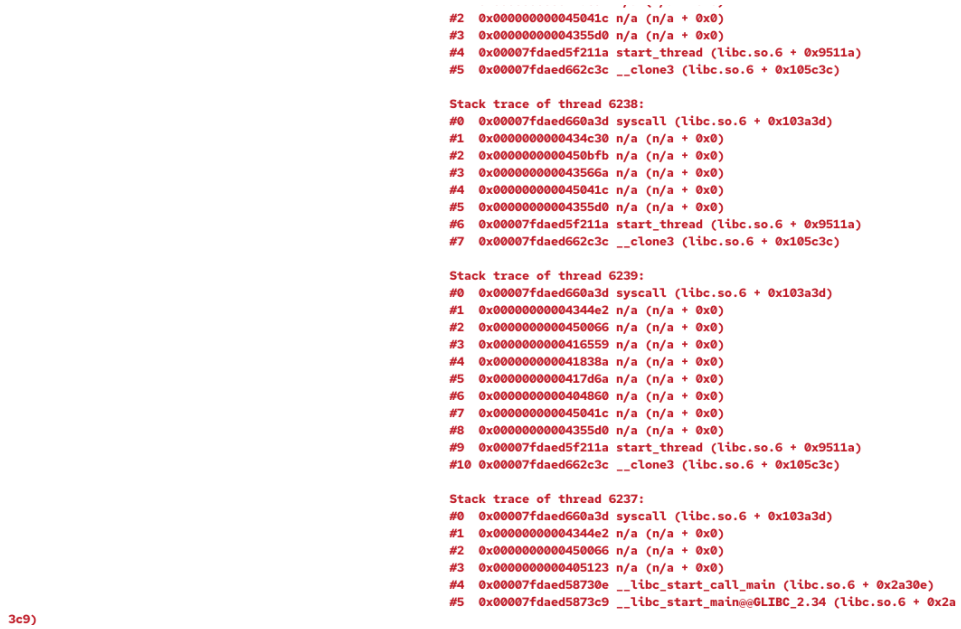

Рис. 2.12: Просмотр системного журнала

3. Далее был включён режим просмотра журнала в реальном времени с помощью journalctl -f.

   Система отображала новые записи по мере их поступления. Выход из режима был выполнен сочетанием клавиш Ctrl + C.

Рис. 2.13: Просмотр системного журнала

4. Для изучения параметров фильтрации командой journalctl дважды была нажата клавиша Tab.

   Терминал вывел список доступных фильтров, позволяющих отбирать события по различным полям — времени, пользователю, типу службы и другим параметрам.

```
root@ehatamov:/home/ehatamov#
root@ehatamov:/home/ehatamov# journalctl
Display all 128 possibilities? (y or n)
_AUDIT_LOGINUID=                    CURRENT_USE_PRETTY=                 PODMAN_TIME=
_AUDIT_SESSION=                     DBUS_BROKER_LOG_DROPPED=            PODMAN_TYPE=
AVAILABLE=                          DBUS_BROKER_METRICS_DISPATCH_AVG=   PRIORITY=
AVAILABLE_PRETTY=                   DBUS_BROKER_METRICS_DISPATCH_COUNT= REALMD_OPERATION=
_BOOT_ID=                           DBUS_BROKER_METRICS_DISPATCH_MAX=   _RUNTIME_SCOPE=
_CAP_EFFECTIVE=                     DBUS_BROKER_METRICS_DISPATCH_MIN=   SEAT_ID=
_CMDLINE=                           DBUS_BROKER_METRICS_DISPATCH_STDDEV= _SELINUX_CONTEXT=
CODE_FILE=                          DISK_AVAILABLE=                     SESSION_ID=
CODE_FUNC=                          DISK_AVAILABLE_PRETTY=              _SOURCE_BOOTTIME_TIMESTAMP=
CODE_LINE=                          DISK_KEEP_FREE=                     _SOURCE_MONOTONIC_TIMESTAMP=
_COMM=                              DISK_KEEP_FREE_PRETTY=              _SOURCE_REALTIME_TIMESTAMP=
CONFIG_FILE=                        ERRNO=                              SSSD_DOMAIN=
CONFIG_LINE=                        _EXE=                               SSSD_PRG_NAME=
COREDUMP_CGROUP=                    _GID=                               _STREAM_ID=
COREDUMP_CMDLINE=                   GLIB_DOMAIN=                        SYSLOG_FACILITY=
COREDUMP_COMM=                      GLIB_OLD_LOG_API=                   SYSLOG_IDENTIFIER=
COREDUMP_CWD=                       _HOSTNAME=                          SYSLOG_PID=
COREDUMP_ENVIRON=                   INITRD_USEC=                        SYSLOG_RAW=
COREDUMP_EXE=                       INVOCATION_ID=                      SYSLOG_TIMESTAMP=
COREDUMP_FILENAME=                  JOB_ID=                             _SYSTEMD_CGROUP=
COREDUMP_GID=                       JOB_RESULT=                         _SYSTEMD_INVOCATION_ID=
COREDUMP_HOSTNAME=                  JOB_TYPE=                           _SYSTEMD_OWNER_UID=
COREDUMP_OPEN_FDS=                  JOURNAL_NAME=                       _SYSTEMD_SESSION=
COREDUMP_OWNER_UID=                 JOURNAL_PATH=                       _SYSTEMD_SLICE=
COREDUMP_PACKAGE_JSON=              _KERNEL_DEVICE=                     _SYSTEMD_UNIT=
COREDUMP_PID=                       _KERNEL_SUBSYSTEM=                  _SYSTEMD_USER_SLICE=
```

Рис. 2.14: Просмотр доступных параметров фильтрации journalctl

5. Для анализа событий, связанных с пользователем root, использовалась команда journalctl _UID=0.

   В выводе отобразились записи, связанные с действиями суперпользователя и служб, запущенных от имени root.

```
root@ehatamov:/home/ehatamov# journalctl _UID=0
Oct 02 17:02:27 ehatamov.localdomain systemd-journald[282]: Collecting audit messages is disabled.
Oct 02 17:02:27 ehatamov.localdomain systemd-journald[282]: Journal started
Oct 02 17:02:27 ehatamov.localdomain systemd-journald[282]: Runtime Journal (/run/log/journal/b47de17405e044edba8d1cda0fbff070) is
Oct 02 17:02:27 ehatamov.localdomain systemd-modules-load[283]: Module 'msr' is built in
Oct 02 17:02:27 ehatamov.localdomain systemd-modules-load[283]: Inserted module 'fuse'
Oct 02 17:02:27 ehatamov.localdomain systemd-modules-load[283]: Module 'scsi_dh_alua' is built in
Oct 02 17:02:27 ehatamov.localdomain systemd-modules-load[283]: Module 'scsi_dh_emc' is built in
Oct 02 17:02:27 ehatamov.localdomain systemd-modules-load[283]: Module 'scsi_dh_rdac' is built in
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished systemd-tmpfiles-setup-dev-early.service - Create Static Device Nodes in
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Starting systemd-sysusers.service - Create System Users...
Oct 02 17:02:27 ehatamov.localdomain systemd-sysusers[298]: Creating group 'nobody' with GID 65534.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished systemd-sysctl.service - Apply Kernel Variables.
Oct 02 17:02:27 ehatamov.localdomain systemd-sysusers[298]: Creating group 'users' with GID 100.
Oct 02 17:02:27 ehatamov.localdomain systemd-sysusers[298]: Creating group 'systemd-journal' with GID 190.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished systemd-sysusers.service - Create System Users.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Starting systemd-tmpfiles-setup-dev.service - Create Static Device Nodes in /dev.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished systemd-vconsole-setup.service - Virtual Console Setup.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: dracut-cmdline-ask.service - dracut ask for additional cmdline parameters was ski
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Starting dracut-cmdline.service - dracut cmdline hook...
Oct 02 17:02:27 ehatamov.localdomain dracut-cmdline[308]: dracut-105-4.el10_0
Oct 02 17:02:27 ehatamov.localdomain dracut-cmdline[308]: Using kernel command line parameters:    BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished systemd-tmpfiles-setup-dev.service - Create Static Device Nodes in /dev.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished dracut-cmdline.service - dracut cmdline hook.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Starting dracut-pre-udev.service - dracut pre-udev hook...
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Finished dracut-pre-udev.service - dracut pre-udev hook.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Starting systemd-udevd.service - Rule-based Manager for Device Events and Files...
Oct 02 17:02:27 ehatamov.localdomain systemd-udevd[408]: Using default interface naming scheme 'rhel-10.0'.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: Started systemd-udevd.service - Rule-based Manager for Device Events and Files.
Oct 02 17:02:27 ehatamov.localdomain systemd[1]: dracut-pre-trigger.service - dracut pre-trigger hook was skipped because no trigg
```

Рис. 2.15: Просмотр событий для пользователя root (UID 0)

6. Для вывода последних двадцати строк журнала была выполнена команда journalctl -n 20.

В результате были показаны свежие записи системных событий, включая ошибки и отчёты ядра.

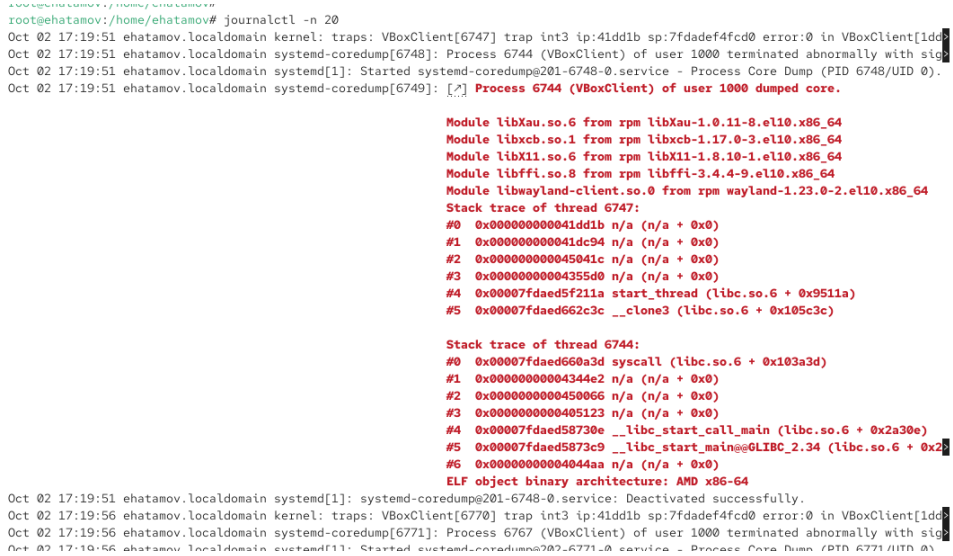

Рис. 2.16: Просмотр последних строк системного журнала

7. Для отображения только сообщений об ошибках применялась команда journalctl -p err.

   В результате вывелись сообщения уровня “error”, включая ошибки графического адаптера и сбои процессов VBoxClient.

Рис. 2.17: Вывод сообщений уровня ошибок

8. Для выборки сообщений за определённый период времени была использована команда journalctl –since yesterday.

   В журнале отобразились все события, зафиксированные со вчерашнего дня.

9. Для фильтрации только ошибок за этот же период применялась команда journalctl –since yesterday -p err.

   Отобразились сообщения, зарегистрированные со вчерашнего дня и имеющие уровень ошибки, включая сбои и предупреждения.

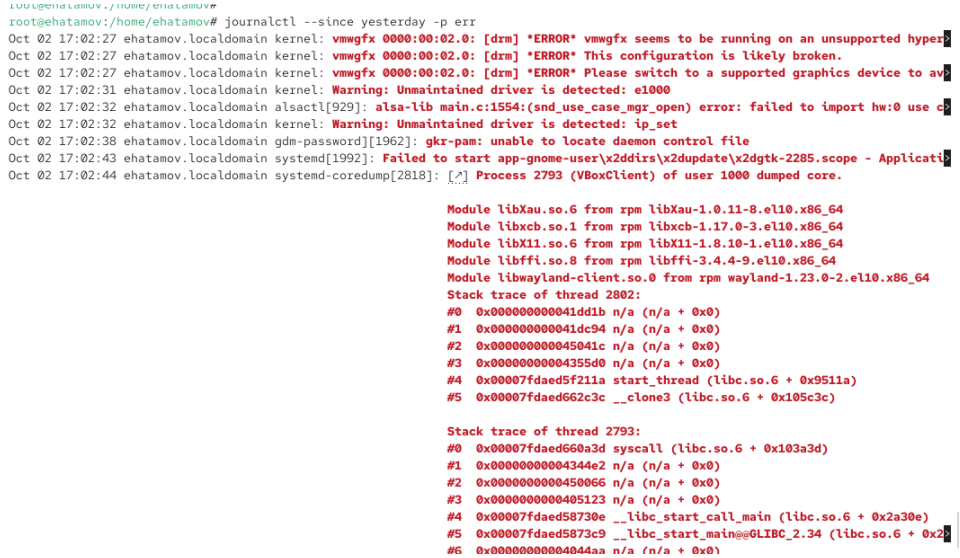

Рис. 2.18: Фильтрация сообщений об ошибках со вчерашнего дня

10. Для вывода подробной информации о событиях использовался параметр -o verbose, который позволяет просматривать расширенные метаданные каждой записи — идентификаторы процессов, временные метки и параметры системы.

Рис. 2.19: Просмотр подробной информации о событиях с параметром verbose

11. Для анализа работы службы sshd была выполнена команда journalctl _SYSTEMD_UNIT=sshd.service.

    В результате отображались записи, связанные с запуском и работой SSH-сервера, включая сообщения о прослушивании порта 22.

2.4 Постоянный журнал journald

1. В терминале были получены права администратора для выполнения системных операций.

2. Был создан каталог для хранения постоянного журнала systemd:
   /var/log/journal
   Этот каталог используется для записи системных событий на диск, обеспечивая их сохранение даже после перезагрузки системы.

3. Для корректной работы службы journald были изменены права доступа к каталогу:
   владельцем назначен root, а группой — systemd-journal.
   Также были установлены права 2755, что разрешает доступ на чтение и запись процессам, связанным с journald.

4. Для применения изменений была отправлена команда killall -USR1 systemd-journald, сигнализирующая службе journald о необходимости обновления конфигурации без полной перезагрузки системы.

5. После выполнения настроек журнал systemd стал постоянным.
   Для проверки отображения сообщений с момента последней перезагрузки использовалась команда journalctl -b.
   В выводе отобразились системные записи ядра, параметры запуска и аппаратная информация.

```
root@ehatamov:/home/ehatamov# mkdir -p /var/log/journal
root@ehatamov:/home/ehatamov# chown root:systemd-journal /var/log/journal/
root@ehatamov:/home/ehatamov# chmod 2755 /var/log/journal/
root@ehatamov:/home/ehatamov# killall -USR1 systemd-journald
root@ehatamov:/home/ehatamov# journalctl -b
Oct 02 17:02:27 ehatamov.localdomain kernel: Linux version 6.12.0-55.12.1.el10_0.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rock>
Oct 02 17:02:27 ehatamov.localdomain kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.12.1.el10_0.x86_64 root=/dev/ma>
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-provided physical RAM map:
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x0000000000100000-0x00000000dffeffff] usable
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x00000000dfff0000-0x00000000dfffffff] ACPI data
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: BIOS-e820: [mem 0x0000000100000000-0x000000011fffffff] usable
Oct 02 17:02:27 ehatamov.localdomain kernel: NX (Execute Disable) protection: active
Oct 02 17:02:27 ehatamov.localdomain kernel: APIC: Static calls initialized
Oct 02 17:02:27 ehatamov.localdomain kernel: SMBIOS 2.5 present.
Oct 02 17:02:27 ehatamov.localdomain kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Oct 02 17:02:27 ehatamov.localdomain kernel: DMI: Memory slots populated: 0/0
Oct 02 17:02:27 ehatamov.localdomain kernel: Hypervisor detected: KVM
Oct 02 17:02:27 ehatamov.localdomain kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
Oct 02 17:02:27 ehatamov.localdomain kernel: kvm-clock: using sched offset of 4082479414 cycles
Oct 02 17:02:27 ehatamov.localdomain kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_>
Oct 02 17:02:27 ehatamov.localdomain kernel: tsc: Detected 3187.196 MHz processor
Oct 02 17:02:27 ehatamov.localdomain kernel: e820: update [mem 0x00000000-0x00000fff] usable ==> reserved
Oct 02 17:02:27 ehatamov.localdomain kernel: e820: remove [mem 0x000a0000-0x000fffff] usable
Oct 02 17:02:27 ehatamov.localdomain kernel: last_pfn = 0x120000 max_arch_pfn = 0x400000000
Oct 02 17:02:27 ehatamov.localdomain kernel: total RAM covered: 4096M
Oct 02 17:02:27 ehatamov.localdomain kernel: Found optimal setting for mtrr clean up
Oct 02 17:02:27 ehatamov.localdomain kernel:  gran_size: 64K         chunk_size: 1G         num_reg: 3         lose cover RAM: 0G
```

Рис. 2.20: Настройка постоянного хранения журнала journald

3 Контрольные вопросы

1. **Какой файл используется для настройки rsyslogd?**

   - Основной конфигурационный файл — `/etc/rsyslog.conf`.

   - Дополнительные правила могут храниться в отдельных файлах в каталоге `/etc/rsyslog.d/`.

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

   - Сообщения, связанные с аутентификацией, хранятся в файле `/var/log/secure`.

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

   - По умолчанию ротация логов выполняется **еженедельно (weekly)**, а старые файлы обычно сохраняются до **4 недель**.

   - Эти параметры задаются в файле `/etc/logrotate.conf`.

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

   - Необходимо добавить строку:

     ```
     *.info /var/log/messages.info
     ```

- Это правило направляет все сообщения уровня **info** и выше в отдельный лог-файл.

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

   - Для просмотра сообщений в реальном времени используется команда:

     ```
     journalctl -f
     ```

   - Аналог команды `tail -f` для системного журнала.

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

   - Для выборки сообщений по идентификатору процесса и времени используется команда:

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

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

   - Для вывода сообщений, относящихся к текущему сеансу загрузки, используется команда:

     ```
     journalctl -b
     ```

8. **Какая процедура позволяет сделать журнал journald постоянным?**

   - Чтобы сделать журнал systemd постоянным, необходимо:

     1. Создать каталог `/var/log/journal`.

     2. Установить права доступа:

        ```
        chown root:systemd-journal /var/log/journal
        chmod 2755 /var/log/journal
        ```

3. Применить изменения без перезагрузки:

   ```
   killall -USR1 systemd-journald
   ```

   После этого логи journald будут сохраняться на диск и не теряться после перезагрузки.

4 Заключение

В ходе выполнения работы были изучены принципы функционирования системных журналов в Linux, а также механизмы их ведения и анализа с помощью служб **rsyslog** и **systemd-journald**.
Были рассмотрены способы настройки файлов конфигурации, перенаправления сообщений по уровням приоритета, создания собственных логов для приложений и организации постоянного хранения записей.
Кроме того, была освоена работа с утилитой **journalctl**, включая применение фильтров, просмотр сообщений в реальном времени и анализ событий по конкретным параметрам.