

Отчёт по лабораторной работе №7

Управление журналами событий в системе

Амина Аджигалиева

Содержание

1	Цель работы	5
2	Ход выполнения работы	6
2.1	Анализ системных журналов	6
2.2	Настройка регистрации сообщений через rsyslog	7
2.3	Использование journalctl	10
2.4	Настройка постоянного журнала journald	15
3	Контрольные вопросы	17
4	Заключение	19

Список иллюстраций

2.1	Сообщения о сбоях VBoxClient и неудачной попытке входа под root	6
2.2	Системные сообщения и повторные ошибки VBoxClient	6
2.3	Просмотр журнала /var/log/secure	7
2.4	Установка и запуск Apache	7
2.5	Просмотр error_log веб-сервера	8
2.6	Редактирование httpd.conf	8
2.7	Создание правила для регистрации ошибок Apache	9
2.8	Создание файла debug.conf	9
2.9	Проверка работы отладочного логирования	9
2.10	Просмотр системного журнала	10
2.11	Отображение доступных фильтров journalctl	12
2.12	Просмотр событий для UID=0	12
2.13	Вывод последних строк журнала	13
2.14	Фильтрация сообщений по приоритету ошибок	13
2.15	События со вчерашнего дня	14
2.16	Ошибки со вчерашнего дня	14
2.17	Подробный вывод журнала	15
2.18	Просмотр журнала sshd	15
2.19	Создание постоянного журнала journald	16

Список таблиц

1 Цель работы

Получить навыки работы с журналами мониторинга различных событий в системе.

2 Ход выполнения работы

2.1 Анализ системных журналов

Сначала я просмотрела логи сбоя процессов, связанных с **VBoxClient**, и сообщения о неудачных попытках получения привилегий root.

На скриншоте видно, что при выполнении команды **su** вход был отклонён, а также зафиксированы ошибки в работе клиента VirtualBox.

```
1c#3 0x0000000000000000 n/a (n/a + 0x0)#012#4 0x0000000000000000 stat_thread (libc.so.6 + 0x0000000000000000)
7f3200435c3c __clone3 (libc.so.6 + 0x105c3c)#012#012Stack trace of thread 3630:#012#0 0x0000000000000000 syscall (
libc.so.6 + 0x103a3d)#012#1 0x00000000000000004344e2 n/a (n/a + 0x0)#012#2 0x0000000000000000450066 n/a (n/a + 0x0)#012#3 0x
0000000000000000405123 n/a (n/a + 0x0)#012#4 0x0000000000000000f320035a30e __libc_start_call_main (libc.so.6 + 0x2a30e)#012#5 0x00
007f320035a3c9 __libc_start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6 0x0000000000004044aa n/a (n/a + 0x0)#012ELF
object binary architecture: AMD x86-64
Sep 29 08:13:21 aradzhigalieva systemd[1]: systemd-coredump@27-3634-0.service: Deactivated successfully.
Sep 29 08:13:25 aradzhigalieva su[3621]: FAILED SU (to root) aradzhigalieva on pts/2
Sep 29 08:13:26 aradzhigalieva kernel: traps: VBoxClient[3645] trap int3 ip:41dd1b sp:7f31fd24cd0 error:0 in VBoxC
lient[1dd1b,400000+bb000]
Sep 29 08:13:26 aradzhigalieva systemd-coredump[3646]: Process 3642 (VBoxClient) of user 1000 terminated abnormally
with signal 5/TRAP, processing...
Sep 29 08:13:26 aradzhigalieva systemd[1]: Started systemd-coredump@28-3646-0.service - Process Core Dump (PID 3646
/UID 0).
Sep 29 08:13:26 aradzhigalieva systemd-coredump[3647]: Process 3642 (VBoxClient) of user 1000 dumped core.#012#012M
```

Рис. 2.1: Сообщения о сбоях VBoxClient и неудачной попытке входа под root

Далее я проанализировала системный журнал, где видно серию сообще-ний «hello» от пользователя **aradzhlgalieva**, а также повторяющиеся ошибки VBoxClient, сопровождающиеся завершением процессов с генерацией дампа памяти.

```
007f320035a3c9 __libc_start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6 0x0000000000004044aa n/a (n/a + 0x0)
object binary architecture: AMD x86-64
Sep 29 08:13:51 aradzhigalieva systemd[1]: systemd-coredump@33-3701-0.service: Deactivated successfully.
Sep 29 08:13:53 aradzhigalieva aradzhigalieva[3707]: hello
Sep 29 08:13:53 aradzhigalieva aradzhigalieva[3712]: hello
Sep 29 08:13:54 aradzhigalieva aradzhigalieva[3714]: hello
Sep 29 08:13:55 aradzhigalieva aradzhigalieva[3716]: hello
Sep 29 08:13:56 aradzhigalieva kernel: traps: VBoxClient[3721] trap int3 ip:41dd1b sp:7f31fd24cd0 error:0
lient[1dd1b,400000+bb000]
Sep 29 08:13:56 aradzhigalieva svstemd-coredump[3722]: Process 3718 (VBoxClient) of user 1000 terminated at
```

Рис. 2.2: Системные сообщения и повторные ошибки VBoxClient

Затем я обратилась к журналу **/var/log/secure**, чтобы отследить авторизации пользователей.

В выводе отчётливо видно открытие сессий для пользователей **gdm**, **aradzhigalieva** и **root**. Также зафиксированы ошибки проверки паролей и события, связанные с PAM (Pluggable Authentication Modules).

```
root@aradzhigalieva:~# tail -n 20 /var/log/secure
Sep 29 08:10:34 aradzhigalieva sshd[1193]: Server listening on :: port 22.
Sep 29 08:10:34 aradzhigalieva (systemd)[1258]: pam_unix(systemd-user:session): session opened for user gdm(uid=42)
by gdm(uid=0)
Sep 29 08:10:34 aradzhigalieva gdm-launch-environment[1237]: pam_unix(gdm-launch-environment:session): session ope
ned for user gdm(uid=42) by (uid=0)
Sep 29 08:10:40 aradzhigalieva unix_chkpwd[1980]: password check failed for user (aradzhigalieva)
Sep 29 08:10:40 aradzhigalieva gdm-password[1973]: pam_unix(gdm-password:auth): authentication failure; logname= u
id=0 euid=0 tty=/dev/tty1 ruser= rhost= user=aradzhigalieva
Sep 29 08:10:40 aradzhigalieva gdm-password[1973]: gkr-pam: unable to locate daemon control file
Sep 29 08:10:40 aradzhigalieva gdm-password[1973]: gkr-pam: stashed password to try later in open session
Sep 29 08:10:52 aradzhigalieva gdm-password[1992]: gkr-pam: unable to locate daemon control file
Sep 29 08:10:52 aradzhigalieva gdm-password[1992]: gkr-pam: stashed password to try later in open session
Sep 29 08:10:52 aradzhigalieva (systemd)[2004]: pam_unix(systemd-user:session): session opened for user aradzhigali
eva(uid=1000) by aradzhigalieva(uid=0)
Sep 29 08:10:52 aradzhigalieva gdm-password[1992]: pam_unix(gdm-password:session): session opened for user aradzh
igalieva(uid=1000) by aradzhigalieva(uid=0)
Sep 29 08:10:52 aradzhigalieva gdm-password[1992]: gkr-pam: gnome-keyring-daemon started properly and unlocked key
ring
Sep 29 08:11:02 aradzhigalieva gdm-launch-environment[1237]: pam_unix(gdm-launch-environment:session): session clo
sed for user gdm
Sep 29 08:12:33 aradzhigalieva (systemd)[3365]: pam_unix(systemd-user:session): session opened for user root(uid=0)
by root(uid=0)
Sep 29 08:12:33 aradzhigalieva su[3350]: pam_unix(su:session): session opened for user root(uid=0) by aradzhigali
eva(uid=1000)
Sep 29 08:12:42 aradzhigalieva su[3445]: pam_unix(su:session): session opened for user root(uid=0) by aradzhigali
eva(uid=1000)
Sep 29 08:12:47 aradzhigalieva su[3508]: pam_unix(su:session): session opened for user root(uid=0) by aradzhigali
eva(uid=1000)
Sep 29 08:13:19 aradzhigalieva su[3508]: pam_unix(su:session): session closed for user root
Sep 29 08:13:22 aradzhigalieva unix_chkpwd[3640]: password check failed for user (root)
```

Рис. 2.3: Просмотр журнала /var/log/secure

2.2 Настройка регистрации сообщений через rsyslog

Сначала я установила веб-сервер Apache и запустила его службу. Также я на-строила автоматический запуск httpd при старте системы.

```
Installed:
apr-1.7.5-2.el10.x86_64
apr-util-lmdb-1.6.3-21.el10.x86_64
httpd-2.4.63-1.el10_0.2.x86_64
httpd-filesystem-2.4.63-1.el10_0.2.noarch
mod_http2-2.0.29-2.el10_0.1.x86_64
rocky-logos-httpd-100.4-7.el10.noarch

apr-util-1.6.3-21.el10.x86_64
apr-util-openssl-1.6.3-21.el10.x86_64
httpd-core-2.4.63-1.el10_0.2.x86_64
httpd-tools-2.4.63-1.el10_0.2.x86_64
mod_lua-2.4.63-1.el10_0.2.x86_64

Complete!
root@aradzhigalieva:~# systemctl start httpd
root@aradzhigalieva:~# systemctl enable httpd
Created symlink '/etc/systemd/system/multi-user.target.wants/httpd.service' → '/usr/lib/systemd/system/httpd.servic
e'.
root@aradzhigalieva:~#
```

Рис. 2.4: Установка и запуск Apache

Затем я просмотрела журнал ошибок веб-службы, чтобы убедиться в корректности её работы.

На скриншоте видно сообщения о запуске Apache и успешной конфигурации.

```
root@aradzhigaliev:~/home/aradzhigaliev#  
root@aradzhigaliev:~/home/aradzhigaliev# tail -f /var/log/httpd/error_log  
[Mon Sep 29 08:15:23.823745 2025] [suexec:notice] [pid 4153:tid 4153] AH01232: suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)  
[Mon Sep 29 08:15:23.867208 2025] [lbmethod_heartbeat:notice] [pid 4153:tid 4153] AH02282: No slotmem from mod_heartmonitor  
[Mon Sep 29 08:15:23.867764 2025] [systemd:notice] [pid 4153:tid 4153] SELinux policy enabled; httpd running as context system_u:system_r:httpd_t:s0  
[Mon Sep 29 08:15:23.870631 2025] [mpm_event:notice] [pid 4153:tid 4153] AH00489: Apache/2.4.63 (Rocky Linux) configured -- resuming normal operations  
[Mon Sep 29 08:15:23.870642 2025] [core:notice] [pid 4153:tid 4153] AH00094: Command line: '/usr/sbin/httpd -D FOREGROUND'
```

Рис. 2.5: Просмотр error_log веб-сервера

После этого я изменила конфигурацию веб-сервера, добавив в файл /etc/httpd/conf/httpd.conf строку:

ErrorLog syslog:local1.

Эта настройка позволяет перенаправлять сообщения об ошибках Apache в системный журнал.



```
GNU nano 8.1 /etc/httpd/conf/httpd.conf Modified  
# Customizable error responses come in three flavors:  
# 1) plain text 2) local redirects 3) external redirects  
#  
# Some examples:  
#ErrorDocument 500 "The server made a boo boo."  
#ErrorDocument 404 /missing.html  
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"  
#ErrorDocument 402 http://www.example.com/subscription_info.html  
#  
#  
# EnableMMAP and EnableSendfile: On systems that support it,  
# memory-mapping or the sendfile syscall may be used to deliver  
# files. This usually improves server performance, but must  
# be turned off when serving from networked-mounted  
# filesystems or if support for these functions is otherwise  
# broken on your system.  
# Defaults if commented: EnableMMAP On, EnableSendfile Off  
#  
#EnableMMAP off  
EnableSendfile on  
  
# Supplemental configuration  
#  
# Load config files in the "/etc/httpd/conf.d" directory, if any.  
IncludeOptional conf.d/*.conf  
ErrorLog syslog:local1
```

Рис. 2.6: Редактирование httpd.conf

В каталоге /etc/rsyslog.d я создала новый файл конфигурации и добавила пра-

вило, которое направляет все сообщения объекта local1 в файл /var/log/httpd-error.log.



Рис. 2.7: Создание правила для регистрации ошибок Apache

Затем я создала отдельный файл debug.conf для регистрации отладочных сообщений и указала в нём правило:

*.debug /var/log/messages-debug.

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

Рис. 2.8: Создание файла debug.conf

После перезапуска служб rsyslog и httpd я проверила работу конфигурации. Для этого выполнила команду `logger -p daemon.debug "Daemon Debug Message"`. На скриншоте видно, что отладочные сообщения корректно записываются в файл /var/log/messages-debug, а ошибки VBoxClient продолжают фиксироваться.

```
007f320035a3c9 __libc_start_main@GLIBC_2.34 (libc.so.6 + 0x2a3c9)#012#6 0x0000000004044aa n/a (n/a + 0x0)#012ELF  
object binary architecture: AMD x86-64  
Sep 29 08:22:26 aradzhigalieva systemd[1]: systemd-coredump@134-5888-0.service: Deactivated successfully.  
Sep 29 08:22:26 aradzhigalieva root[5894]: Daemon Debug Message  
Sep 29 08:22:27 aradzhigalieva root[5899]: Daemon Debug Message  
Sep 29 08:22:28 aradzhigalieva root[5901]: Daemon Debug Message  
Sep 29 08:22:31 aradzhigalieva kernel: traps: VBoxClient[5906] trap int3 ip:41dd1b sp:7f31fd24cd0 error:0 in VBoxC  
lient[1dd1b,400000+bb000]  
Sep 29 08:22:31 aradzhigalieva systemd-coredump[5907]: Process 5903 (VBoxClient) of user 1000 terminated abnormally  
with signal 5/TRAP, processing...  
Sep 29 08:22:31 aradzhigalieva systemd[1]: Started systemd-coredump@135-5907-0.service - Process Core Dump (PID 590  
7/UID 0).  
Sep 29 08:22:31 aradzhigalieva systemd-coredump[5908]: Process 5903 (VBoxClient) of user 1000 dumped core.#012#012M  
odules: libXv.so.6 from /usr/lib64: 0 11 0 110 006 640012Modules: libXv.so.6 from /usr/lib64: 1 17 0 2 110 006 6400
```

Рис. 2.9: Проверка работы отладочного логирования

2.3 Использование journalctl

Сначала я просмотрела журнал системы, чтобы получить список событий, начиная с последней загрузки.

Команда `journalctl` показала записи ядра и системных служб, включая параметры запуска и информацию о виртуальной машине.

```
root@aradzhigalieva:/home/aradzhigalieva# journalctl
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Linux version 6.12.0-55.12.1.el10_0.x86_64 (mockbuild@iad1-prod
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.12.1.el10_0
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-provided physical RAM map:
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000009fbff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000000009fc00-0x0000000000000fffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000ff000-0x0000000000000fffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x000000000000100000-0x000000000000dfffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000dffff0000-0x00000000dffff000] ACPI data
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000ffc0000-0x00000000ffffffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000100000000-0x000000011fffffffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: NX (Execute Disable) protection: active
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: APIC: Static calls initialized
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: SMBIOS 2.5 present.
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: DMI: Memory slots populated: 0/0
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Hypervisor detected: KVM
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: kvm-clock: using sched offset of 4075594194 cycles
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: tsc: Detected 3187.200 MHz processor
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: e820: update [mem 0x00000000-0x000000fff] usable ==> reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: e820: remove [mem 0x000a0000-0x0000ffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: last_pfn = 0x120000 max_arch_pfn = 0x400000000
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: total RAM covered: 4096M
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Found optimal setting for mtrr clean up
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: gran_size: 64K chunk_size: 1G num_reg: 3
```

Рис. 2.10: Просмотр системного журнала

Затем я проанализировала ошибки, связанные с библиотеками и аварийным завершением процессов.

Скриншоты фиксируют трассировки стека (stack trace), полученные при падении процессов.

```
x86_64
.23.0-2.el10.x86_64

0x9511a)
05c3c)

3a3d)

libc.so.6 + 0x2a30e)
2.34 (libc.so.6 + 0x2a3c9)

Sep 29 08:23:58 aradzhigalieva.localdomain systemd[1]: systemd-coredump@152-6138-0.service: Deactivated successfully.
root@aradzhigalieva:/home/aradzhigalieva#

Module libwayland-client.so.0 from rpm wayland-1

Stack trace of thread 6137:
#0 0x00000000041dd1b n/a (n/a + 0x0)
#1 0x00000000041dc94 n/a (n/a + 0x0)
#2 0x00000000045041c n/a (n/a + 0x0)
#3 0x0000000004355d0 n/a (n/a + 0x0)
#4 0x00007f32003c511a start_thread (libc.so.6 +
#5 0x00007f3200435c3c __clone3 (libc.so.6 + 0x1

Stack trace of thread 6134:
#0 0x00007f3200433a3d syscall (libc.so.6 + 0x10

#1 0x0000000004344e2 n/a (n/a + 0x0)
#2 0x000000000450066 n/a (n/a + 0x0)
#3 0x000000000405123 n/a (n/a + 0x0)
#4 0x00007f320035a30e __libc_start_call_main (l
#5 0x00007f320035a3c9 __libc_start_main@@GLIBC_
#6 0x0000000004044aa n/a (n/a + 0x0)
ELF object binary architecture: AMD x86-64

Stack trace of thread 6169:
#0 0x00007f3200433a3d syscall (libc.so.6 + 0x10

#1 0x000000000434c30 n/a (n/a + 0x0)
#2 0x000000000450bfb n/a (n/a + 0x0)
#3 0x00000000043566a n/a (n/a + 0x0)
#4 0x00000000045041c n/a (n/a + 0x0)
#5 0x0000000004355d0 n/a (n/a + 0x0)
#6 0x00007f32003c511a start_thread (libc.so.6 +
#7 0x00007f3200435c3c __clone3 (libc.so.6 + 0x1

Stack trace of thread 6168:
#0 0x00007f3200433a3d syscall (libc.so.6 + 0x10

#1 0x0000000004344e2 n/a (n/a + 0x0)
#2 0x000000000450066 n/a (n/a + 0x0)
#3 0x000000000405123 n/a (n/a + 0x0)
#4 0x00007f320035a30e __libc_start_call_main (l
#5 0x00007f320035a3c9 __libc_start_main@@GLIBC_
#6 0x0000000004044aa n/a (n/a + 0x0)
ELF object binary architecture: AMD x86-64

Sep 29 08:24:13 aradzhigalieva.localdomain systemd[1]: systemd-coredump@155-6172-0.service: Deactivated successfully.
y.
█
```

Для фильтрации событий я использовала встроенные параметры. При вводе команды `journalctl` и двойном нажатии Tab был отображён список доступных фильтров.

```

root@aradzhigaliev:/home/aradzhigaliev# 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_ICON=  KERNEL_DEVIC=                _SYSTEMD_UNIT=

```

Рис. 2.11: Отображение доступных фильтров journalctl

Затем я просмотрела только те события, которые связаны с пользователем root (UID=0).

В журнале отображаются действия системных служб и процессы, запускаемые от имени суперпользователя.

```

root@aradzhigaliev:/home/aradzhigaliev# journalctl_UID=0
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-journald[281]: Collecting audit messages is disabled.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-journald[281]: Journal started
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-journald[281]: Runtime Journal (/run/log/journal/343a350182dc428
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-modules-load[282]: Module 'msr' is built in
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-modules-load[282]: Inserted module 'fuse'
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-modules-load[282]: Module 'scsi_dh_alua' is built in
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-modules-load[282]: Module 'scsi_dh_emc' is built in
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-modules-load[282]: Module 'scsi_dh_rdac' is built in
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished systemd-vconsole-setup.service - Virtual Console Se
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: dracut-cmdline-ask.service - dracut ask for additional cmdlin
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Starting dracut-cmdline.service - dracut cmdline hook...
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-sysusers[295]: Creating group 'nobody' with GID 65534.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-sysusers[295]: Creating group 'users' with GID 100.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished systemd-sysctl.service - Apply Kernel Variables.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd-sysusers[295]: Creating group 'systemd-journal' with GID 190.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished systemd-sysusers.service - Create System Users.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Starting systemd-tmpfiles-setup-dev.service - Create Static
Sep 29 08:10:29 aradzhigaliev.localdomain dracut-cmdline[304]: dracut-105-4.el10_0
Sep 29 08:10:29 aradzhigaliev.localdomain dracut-cmdline[304]: Using kernel command line parameters: BOOT_IMAG
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished systemd-tmpfiles-setup-dev.service - Create Static
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished dracut-cmdline.service - dracut cmdline hook.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Starting dracut-pre-udev.service - dracut pre-udev hook...
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Finished dracut-pre-udev.service - dracut pre-udev hook.
Sep 29 08:10:29 aradzhigaliev.localdomain systemd[1]: Starting systemd-udev.service - Rule-based Manager for Devic

```

Рис. 2.12: Просмотр событий для UID=0

Для анализа последних записей я вывела последние 20 строк журнала с помощью команды `journalctl -n 20`.

В сообщениях видны ошибки клиента VirtualBox и созданные дампы процессов.

```

root@aradzhigalieva: /home/aradzhigalieva# journalctl -n 20
Sep 29 08:25:14 aradzhigalieva.localdomain systemd-coredump[6337]: [P] Process 6332 (VBoxClient) of user 1000 dump>
Module libXau.so.6 from rpm libXau-1.0.11-8.el1>
Module libxcb.so.1 from rpm libxcb-1.17.0-3.el1>
Module libX11.so.6 from rpm libX11-1.8.10-1.el1>
Module libffi.so.8 from rpm libffi-3.4.4-9.el10>
Module libwayland-client.so.0 from rpm wayland>
Stack trace of thread 6335:
#0 0x00000000041dd1b n/a (n/a + 0x0)
#1 0x00000000041dc94 n/a (n/a + 0x0)
#2 0x00000000045041c n/a (n/a + 0x0)
#3 0x0000000004355d0 n/a (n/a + 0x0)
#4 0x00007f32003c511a start_thread (libc.so.6 >
#5 0x00007f3200435c3c __clone3 (libc.so.6 + 0x>
Stack trace of thread 6332:
#0 0x00007f3200433a3d syscall (libc.so.6 + 0x1>
#1 0x0000000004344e2 n/a (n/a + 0x0)
#2 0x000000000450066 n/a (n/a + 0x0)
#3 0x000000000405123 n/a (n/a + 0x0)
#4 0x00007f320035a30e __libc_start_call_main (>
#5 0x00007f320035a3c9 __libc_start_main@@GLIBC>
#6 0x0000000004044aa n/a (n/a + 0x0)
ELF object binary architecture: AMD x86-64

```

Рис. 2.13: Вывод последних строк журнала

Далее я просмотрела только сообщения уровня ошибок с помощью команды `journalctl -p err`.

В выводе содержатся ошибки графического драйвера `vmwgfx`, предупреждения ядра и сбой процессов `VBoxClient`.

```

root@aradzhigalieva: /home/aradzhigalieva# journalctl -p err
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* vmwgfx seems to be running o>
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* This configuration is likely>
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] *ERROR* Please switch to a supported>
Sep 29 08:10:32 aradzhigalieva.localdomain kernel: Warning: Unmaintained driver is detected: e1000
Sep 29 08:10:33 aradzhigalieva.localdomain alsactl[932]: alsa-lib main.c:1554:(snd_use_case_mgr_open) error: failed>
Sep 29 08:10:34 aradzhigalieva.localdomain kernel: Warning: Unmaintained driver is detected: ip_set
Sep 29 08:10:40 aradzhigalieva.localdomain gdm-password[1973]: gkr-pam: unable to locate daemon control file
Sep 29 08:10:52 aradzhigalieva.localdomain gdm-password[1992]: gkr-pam: unable to locate daemon control file
Sep 29 08:11:03 aradzhigalieva.localdomain systemd-coredump[2813]: [P] Process 2795 (VBoxClient) of user 1000 dump>
Module libXau.so.6 from rpm libXau-1.0.11-8.el1>
Module libxcb.so.1 from rpm libxcb-1.17.0-3.el1>
Module libX11.so.6 from rpm libX11-1.8.10-1.el1>
Module libffi.so.8 from rpm libffi-3.4.4-9.el10>
Module libwayland-client.so.0 from rpm wayland>
Stack trace of thread 2798:
#0 0x00000000041dd1b n/a (n/a + 0x0)
#1 0x00000000041dc94 n/a (n/a + 0x0)
#2 0x00000000045041c n/a (n/a + 0x0)
#3 0x0000000004355d0 n/a (n/a + 0x0)
#4 0x00007f32003c511a start_thread (libc.so.6 >
#5 0x00007f3200435c3c __clone3 (libc.so.6 + 0x>
Stack trace of thread 2797:
#0 0x00007f3200433a3d syscall (libc.so.6 + 0x1>
#1 0x0000000004344e2 n/a (n/a + 0x0)
#2 0x000000000450066 n/a (n/a + 0x0)
#3 0x000000000416559 n/a (n/a + 0x0)

```

Рис. 2.14: Фильтрация сообщений по приоритету ошибок

Используя параметр `--since yesterday`, я просмотрела все события, начиная со вчерашнего дня.

Журнал снова показал последовательность загрузки системы и параметры ядра.

```

root@aradzhigalieva:/home/aradzhigalieva# journalctl --since yesterday
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Linux version 6.12.0-55.12.1.el10_0.x86_64 (mockbuild@iad1-prod)
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.12.1.el10_0
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-provided physical RAM map:
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000000-0x00000000000009fbff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000000009fc00-0x00000000000009ffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000f0000-0x0000000000000fffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000001000000-0x000000000dffffffffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000dffff0000-0x00000000dffffffffff] ACPI data
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00ffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00ffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000100000000-0x000000011fffffffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: NX (Execute Disable) protection: active
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: APIC: Static calls initialized
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: SMBIOS 2.5 present.
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: DMI: Memory slots populated: 0/0
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Hypervisor detected: KVM
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: kvm-clock: Using msrs 4b564d01 and 4b564d00
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: kvm-clock: using sched offset of 4075594194 cycles
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: clocksource: mask: 0xffffffffffffffff max_cycles: 0x1
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: tsc: Detected 3187.200 MHz processor
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: e820: update [mem 0x00000000-0x000000ffff] usable ==> reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: e820: remove [mem 0x00000000-0x000000ffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: last_pfn = 0x120000 max_arch_pfn = 0x400000000
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: total RAM covered: 4096M
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Found optimal setting for mtrr clean up
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: gran_size: 64K chunk_size: 1G num_reg: 3

```

Рис. 2.15: События со вчерашнего дня

Затем я объединила фильтры и вывела только ошибки, зафиксированные со вчерашнего дня: `journalctl --since yesterday -p err`.

Результат включает сообщения о проблемах с драйвером `vmwgfx`, РАМ-авторизацией и сбоями `VBoxClient`.

```

root@aradzhigalieva:/home/aradzhigalieva# journalctl --since yesterday -p err
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] "ERROR" vmwgfx seems to be running on
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] "ERROR" This configuration is likely
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: vmwgfx 0000:00:02.0: [drm] "ERROR" Please switch to a supported
Sep 29 08:10:32 aradzhigalieva.localdomain kernel: Warning: Unmaintained driver is detected: e1000
Sep 29 08:10:33 aradzhigalieva.localdomain alsactl[932]: alsa-lib main.c:1554:(snd_use_case_mgr_open) error: failed
Sep 29 08:10:34 aradzhigalieva.localdomain kernel: Warning: Unmaintained driver is detected: ip_set
Sep 29 08:10:40 aradzhigalieva.localdomain gdm-password[1973]: gkr-pam: unable to locate daemon control file
Sep 29 08:10:52 aradzhigalieva.localdomain gdm-password[1992]: gkr-pam: unable to locate daemon control file
Sep 29 08:11:03 aradzhigalieva.localdomain systemd-coredump[2813]: Process 2795 (VBoxClient) of user 1000 dump:

Module libXau.so.6 from rpm libXau-1.0.11-8.el10
Module libxcb.so.1 from rpm libxcb-1.17.0-3.el10
Module libX11.so.6 from rpm libX11-1.8.10-1.el10
Module libffi.so.8 from rpm libffi-3.4.4-9.el10
Module libwayland-client.so.0 from rpm wayland-1.22.0-2.el10
Stack trace of thread 2798:
#0  0x000000000041dd1b n/a (n/a + 0x0)
#1  0x000000000041dc94 n/a (n/a + 0x0)
#2  0x000000000045041c n/a (n/a + 0x0)
#3  0x00000000004355d0 n/a (n/a + 0x0)
#4  0x00007f32003c511a start_thread (libc.so.6 + 0x7f32003c511a)
#5  0x00007f3200435c3c __clone3 (libc.so.6 + 0x7f3200435c3c)

Stack trace of thread 2797:
#0  0x00007f3200433a3d syscall (libc.so.6 + 0x7f3200433a3d)
#1  0x00000000004344e2 n/a (n/a + 0x0)
#2  0x0000000000450066 n/a (n/a + 0x0)
#3  0x0000000000416559 n/a (n/a + 0x0)

```

Рис. 2.16: Ошибки со вчерашнего дня

Для получения более подробных сведений я использовала режим `journalctl`

-o verbose.

На скриншоте видно, что каждая запись сопровождается расширенными атрибутами: временем, хостом, идентификатором процесса и приоритетом.

```
Mon 2025-09-29 08:10:29.274409 MSK [s=b982d14e533447768cfd92e9877c7bb;i=1;b=1e229539ae9f4de9aac453a58766d518;m=12>
  _SOURCE_BOOTTIME_TIMESTAMP=0
  _SOURCE_MONOTONIC_TIMESTAMP=0
  _TRANSPORT=kernel
  PRIORITY=5
  SYSLOG_FACILITY=0
  SYSLOG_IDENTIFIER=kernel
  MESSAGE=Linux version 6.12.0-55.12.1.el10_0.x86_64 (mockbuild@iad1-prod-build001.bld.equ.rockylinux.org) (gcc
  _BOOT_ID=1e229539ae9f4de9aac453a58766d518
  _MACHINE_ID=343a350182dc4246a2a53c31ed9190c4
  _HOSTNAME=aradzhigalieva.localdomain
  _RUNTIME_SCOPE=initrd
Mon 2025-09-29 08:10:29.274422 MSK [s=b982d14e533447768cfd92e9877c7bb;i=2;b=1e229539ae9f4de9aac453a58766d518;m=12>
  _SOURCE_BOOTTIME_TIMESTAMP=0
  _SOURCE_MONOTONIC_TIMESTAMP=0
  _TRANSPORT=kernel
  SYSLOG_FACILITY=0
  SYSLOG_IDENTIFIER=kernel
  _BOOT_ID=1e229539ae9f4de9aac453a58766d518
  _MACHINE_ID=343a350182dc4246a2a53c31ed9190c4
  _HOSTNAME=aradzhigalieva.localdomain
  _RUNTIME_SCOPE=initrd
  PRIORITY=6
  MESSAGE=Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.12.1.el10_0.x86_64 root=/dev/mapper/rl_vbox-root
Mon 2025-09-29 08:10:29.274427 MSK [s=b982d14e533447768cfd92e9877c7bb;i=3;b=1e229539ae9f4de9aac453a58766d518;m=12>
  _SOURCE_BOOTTIME_TIMESTAMP=0
  _SOURCE_MONOTONIC_TIMESTAMP=0
  _TRANSPORT=kernel
  SYSLOG_FACILITY=0
  SYSLOG_IDENTIFIER=kernel
lines 1-30
```

Рис. 2.17: Подробный вывод журнала

Я использовала команду `journalctl _SYSTEMD_UNIT=sshd.service`, чтобы посмотреть сообщения, относящиеся к службе `sshd`.

В выводе отразились предупреждение об `unset`-переменной окружения, а также информация о том, что сервер слушает порт 22.

```
root@aradzhigalieva:/home/aradzhigalieva#
root@aradzhigalieva:/home/aradzhigalieva# journalctl _SYSTEMD_UNIT=sshd.service
Sep 29 08:10:34 aradzhigalieva.localdomain (sshd)[1193]: sshd.service: Referenced but unset environment variable
Sep 29 08:10:34 aradzhigalieva.localdomain sshd[1193]: Server listening on 0.0.0.0 port 22.
Sep 29 08:10:34 aradzhigalieva.localdomain sshd[1193]: Server listening on :: port 22.
lines 1-3/3 (END)
```

Рис. 2.18: Просмотр журнала `sshd`

2.4 Настройка постоянного журнала `journald`

По умолчанию система хранит журнал `journald` во временном каталоге `/run/log/journal`, и после перезагрузки записи теряются.

Чтобы сделать журнал постоянным, я создала каталог `/var/log/journal`, назначила владельца и права доступа, а затем отправила сигнал `USR1` процессу `systemd-journald`.

```
-----
root@aradzhigalieva:/home/aradzhigalieva# mkdir -p /var/log/journal
root@aradzhigalieva:/home/aradzhigalieva# chown root:systemd-journal /var/log/journal/
root@aradzhigalieva:/home/aradzhigalieva# chmod 755 /var/log/journal/
root@aradzhigalieva:/home/aradzhigalieva# killall -USR1 systemd-journald
root@aradzhigalieva:/home/aradzhigalieva# journalctl -b
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Linux version 6.12.0-55.12.1.el10_0.x86_64 (mockbuild@iad1-prod)
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: Command line: BOOT_IMAGE=(hd0,gpt2)/vmlinuz-6.12.0-55.12.1.el10_0
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-provided physical RAM map:
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000009fc00-0x0000000000009ffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x00000000000f0000-0x000000000000ffff] reserved
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000001000000-0x000000000000ffff] usable
Sep 29 08:10:29 aradzhigalieva.localdomain kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] reserved
```

Рис. 2.19: Создание постоянного журнала `journald`

После этого я выполнила команду `journalctl -b`, чтобы убедиться, что журнал сохраняет записи с момента последней загрузки системы.

В выводе отразились стандартные сообщения ядра и параметры запуска.

3 Контрольные вопросы

1. Какой файл используется для настройки rsyslogd?

Основной файл конфигурации — `/etc/rsyslog.conf`. Дополнительные правила могут храниться в каталоге `/etc/rsyslog.d/`.

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

Сообщения об аутентификации обычно записываются в файл `/var/log/secure`.

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

По умолчанию ротация журналов выполняется еженедельно (настраивается через `logrotate`).

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

В конфигурацию нужно добавить строку:

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

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

Для этого используется команда:

```
journalctl -f
```

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

Используется команда:

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

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

Для этого используется команда:

```
journalctl -b
```

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

Нужно создать каталог `/var/log/journal`, назначить владельца и права:

```
mkdir -p /var/log/journal
```

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

```
chmod 2755 /var/log/journal
```

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

4 Заключение

В ходе выполнения лабораторной работы я изучила принципы работы системных журналов Linux и научилась настраивать их хранение и просмотр.

Я освоила использование команд `journalctl` и различных параметров фильтрации, научилась отслеживать ошибки и события в режиме реального времени, а также работать с приоритетами сообщений.

Кроме того, я настроила службу `rsyslogd` для перенаправления логов веб-сервера Apache и добавила правила для записи отладочной информации в отдельный файл.

Я также сделала журнал `journald` постоянным, чтобы записи сохранялись после перезагрузки системы.

Все выполненные действия позволили закрепить знания о механизмах логирования в Linux, а также получить практические навыки администрирования, которые необходимы для обеспечения контроля и диагностики работы системы.