

# **Лабораторная работа № 15**

**Настройка сетевого журналирования**

**Шубина София Антоновна**

# **Содержание**

<b>1 Цель работы</b>	<b>4</b>
<b>2 Задание</b>	<b>5</b>
<b>3 Выполнение лабораторной работы</b>	<b>6</b>
3.1 Настройка сервера сетевого журнала . . . . .	6
3.2 Настройка клиента сетевого журнала . . . . .	8
3.3 Просмотр журнала . . . . .	8
3.4 Внесение изменений в настройки внутреннего окружения виртуальных машин . . . . .	14
<b>4 Контрольные вопросы</b>	<b>17</b>
<b>5 Выводы</b>	<b>19</b>

# Список иллюстраций

3.1	создание файла	6
3.2	Включение журналирования по TCP-порту 514	6
3.3	Просмотр прослушиваемых портов, связанных с rsyslog	7
3.4	Настройка межсетевого экрана	7
3.5	Проверка 514 порта	7
3.6	создание файла	8
3.7	Включение перенаправления сообщений журнала на 514 TCP-порт сервера	8
3.8	перезапустим службу	8
3.9	Просмотр файла var/log/messages журнала	9
3.10	Запуск графической программы для просмотра журналов	10
3.11	загрузка	11
3.12	Просмотр логов с сервера	11
3.13	Просмотр логов с сервера	11
3.14	Просмотр логов с клиента	12
3.15	Просмотр логов с клиента	12
3.16	создание файла	14
3.17	Скрипта файла /vagrant/provision/server/netlog.sh	14
3.18	создание файла	15
3.19	Скрипта файла /vagrant/provision/client/ netlog.sh	15
3.20	Vagrantfile	16
3.21	Vagrantfile	16

# **1 Цель работы**

Получение навыков по работе с журналами системных событий.

## **2 Задание**

1. Настройте сервер сетевого журналирования событий.
2. Настройте клиент для передачи системных сообщений в сетевой журнал на сервере.
3. Просмотрите журналы системных событий с помощью нескольких программ. При наличии сообщений о некорректной работе сервисов исправьте ошибки в настройках соответствующих служб.
4. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сетевого сервера журналирования

# 3 Выполнение лабораторной работы

## 3.1 Настройка сервера сетевого журнала

Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:

```
cd /work/sashubina/vagrant
```

Затем запустим виртуальную машину server:

```
vagrant up server
```

На сервере создадим файл конфигурации сетевого хранения журналов:

```
cd /etc/rsyslog.d  
touch netlog-server.conf
```

```
[root@server.sashubina.net ~]# cd /etc/rsyslog.d  
[root@server.sashubina.net rsyslog.d]# touch netlog-server.conf  
[root@server.sashubina.net rsyslog.d]#
```

Рис. 3.1: создание файла

В файле конфигурации /etc/rsyslog.d/netlog-server.conf включим приём записей журнала по TCP-порту 514:

```
GNU nano 8.1          /etc/rsyslog.d/netlog-server.conf  
$ModLoad imtcp  
$InputTCPServerRun 514
```

Рис. 3.2: Включение журналирования по TCP-порту 514

Перезапустим службу rsyslog и посмотрим, какие порты, связанные с rsyslog, прослушиваются:

```
[root@server.sashubina.net rsyslog.d]# netstat -an | grep :89661
rsyslogd 19819          root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819          root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19822 in:imjour   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19822 in:imjour   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19823 in:imtcp   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19823 in:imtcp   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19824 in:imtcp   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19824 in:imtcp   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19825 in:imtcp   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19825 in:imtcp   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19826 in:imtcp   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19826 in:imtcp   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19827 in:imtcp   root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19827 in:imtcp   root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19828 rs:main    root  4u    IPv4          89661      0t0      TCP *:shell (LISTEN)
rsyslogd 19819 19828 rs:main    root  5u    IPv6          89662      0t0      TCP *:shell (LISTEN)
[root@server.sashubina.net rsyslog.d]#
```

Рис. 3.3: Просмотр прослушиваемых портов, связанных с rsyslog

Порты 89661 и 89662 прослушиваются rsyslogd - системный демон журналирования

На сервере настроим межсетевой экран для приёма сообщений по TCP-порту 514:

```
[root@server.sashubina.net rsyslog.d]# firewall-cmd --add-port=514/tcp
success
[root@server.sashubina.net rsyslog.d]# firewall-cmd --add-port=514/tcp --permanent
success
[root@server.sashubina.net rsyslog.d]#
```

Рис. 3.4: Настройка межсетевого экрана

Проверим 514 порт

```
[root@server.sashubina.net rsyslog.d]# sudo ss -tlnp | grep 514
LISTEN 0      25          0.0.0.0:514          0.0.0.0:*      users:(("rsyslogd",pi
d=19819,fd=4))
LISTEN 0      25          [::]:514           [::]:*      users:(("rsyslogd",pi
d=19819,fd=5))
[root@server.sashubina.net rsyslog.d]#
```

Рис. 3.5: Проверка 514 порта

Rsyslog демон работает и принимает логи по TCP на порту 514. Слушает как IPv4, так и IPv6 подключения. Это нормальная конфигурация для централизованного сбора логов

## 3.2 Настройка клиента сетевого журнала

На клиенте создадим файл конфигурации сетевого хранения журналов:

```
cd /etc/rsyslog.d  
touch netlog-client.conf
```

```
[sashubina@client.sashubina.net ~]$ sudo -i  
[sudo] password for sashubina:  
[root@client.sashubina.net ~]# cd /etc/rsyslog.d  
[root@client.sashubina.net rsyslog.d]# touch netlog-client.conf  
[root@client.sashubina.net rsyslog.d]#
```

Рис. 3.6: создание файла

На клиенте в файле конфигурации /etc/rsyslog.d/netlog-client.conf включим перенаправление сообщений журнала на 514 TCP-порт сервера:

```
GNU nano 8.1      /etc/rsyslog.d/netlog-client.conf  
.* @@server.sashubina.net:514
```

Рис. 3.7: Включение перенаправления сообщений журнала на 514 TCP-порт сервера

Перезапустим службу rsyslog:

```
systemctl restart rsyslog
```

```
[root@client.sashubina.net rsyslog.d]# systemctl restart rsyslog  
[root@client.sashubina.net rsyslog.d]#
```

Рис. 3.8: перезапустим службу

## 3.3 Просмотр журнала

На сервере просмотрим один из файлов журнала:

```
[root@server sashubina.net ~]# tail -f /var/log/messages
Nov 10 17:09:36 server systemd[4167]: Finished systemd-tmpfiles-setup.service - Create User Files and Directories.
Nov 10 17:09:36 server systemd[4167]: Listening on dbus.socket - D-Bus User Message Bus Socket.
Nov 10 17:09:36 server systemd[4167]: Reached target sockets.target - Sockets.
Nov 10 17:09:36 server systemd[4167]: Reached target basic.target - Basic System.
Nov 10 17:09:36 server systemd[4167]: Reached target default.target - Main User Target.
Nov 10 17:09:36 server systemd[4167]: Startup finished in 177ms.
Nov 10 17:09:36 server systemd[1]: Started user@.service - User Manager for UID 0.
Nov 10 17:09:36 server systemd[1]: Started session-c2.scope - Session c2 of User root.
Nov 10 17:09:36 server systemd[1]: Starting systemd-hostnamed.service - Hostname Service...
Nov 10 17:09:36 server systemd[1]: Started systemd-hostnamed.service - Hostname Service.
Nov 10 17:10:06 server systemd[1]: systemd-hostnamed.service: Deactivated successfully.
```

Рис. 3.9: Просмотр файла var/log/messages журнала

rsyslogd с PID 19819 прослушивает как стандартный TCP-порт 514 для приема логов, так и дополнительные порты 89661 и 89662 на IPv4 и IPv6 интерфейсах. При этом необычным является то, что эти же эфемерные порты также прослушиваются процессом shell, что может указывать на перезапуск службы или конфигурацию с совместным использованием портов. Параллельно в системе ведется живой мониторинг системных логов через tail -f /var/log/messages, где отображаются нормальные операционные события: инициализация пользовательской среды systemd для root, запуск служб управления временными файлами, D-Bus и сокетов, а также периодическая активация и деактивация сервиса hostname, что является типичной активностью для работающего Linux-сервера.

На сервере под пользователем sashubina запустим графическую программу для просмотра журналов с помощью команды gnome-system-monitor:

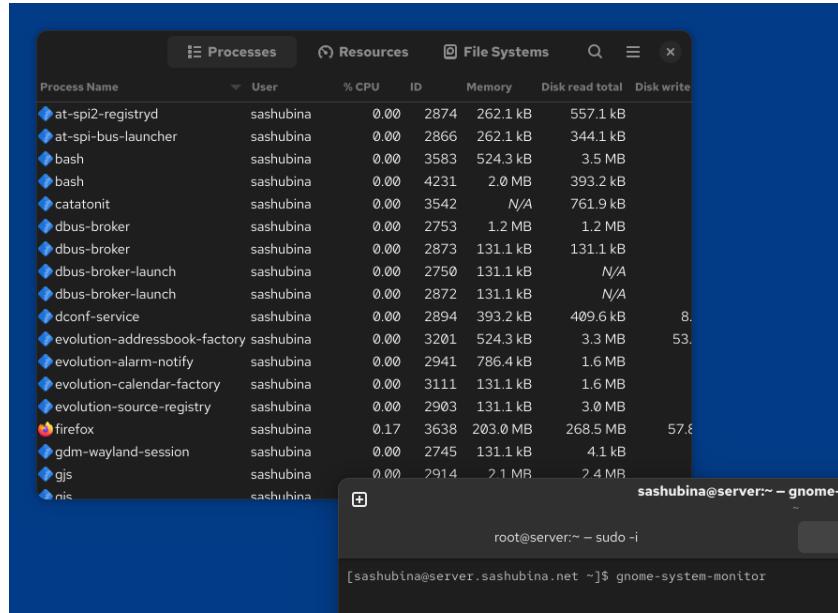


Рис. 3.10: Запуск графической программы для просмотра журналов

демон системного журналирования rsyslogd (PID 19819) прослушивает стандартный TCP-порт 514 для приема логов, а также дополнительные порты 89661 и 89662 на всех сетевых интерфейсах. Параллельно эти же порты заняты процессом shell, что может указывать на перезапуск службы или особую конфигурацию. В системе ведется мониторинг логов, где фиксируются штатные события инициализации пользовательской среды и работы службы. Одновременно в графической сессии GNOME пользователя sashubina активны различные приложения, включая Firefox, почтовый клиент Evolution и системные службы, потребляющие умеренные ресурсы без значительной нагрузки на CPU.

На сервере установите просмотрщик журналов системных сообщений lnav или его аналог. Я установила аналог, потому что lnav у меня не скачался

```
dnf -y install multitail
```

```
root@rhel7-1: ~]# curl -s https://mirrors.tuna.tsinghua.edu.cn/multilib/x86_64/y/install.multilib | bash
Last metadata expiration check: 1:58:56 ago on Mon Mar 18 Nov 2025 03:26:23 PM UTC.
Metadata files received:
-----
```

Package	Architecture	Version	Repository	Size
multilib-7.1.3-2.el10.x86_64	x86_64	7.1.3-2.el10.x86_64	epel	148 kB

```
Transaction Summary
install 1 Package

Total download size: 148 kB
Is this ok [y/N]: y
Downloaded Packages:
multilib-7.1.3-2.el10.x86_64.rpm
```

1.5 MB/s   148 kB 00:00
00:00:00   148 kB 00:01

```
Running transaction check
Transaction check succeeded
Running transaction test
Transaction test succeeded
Preparing to unpack.../multilib-7.1.3-2.el10.x86_64
Running scriptlet: multilib-7.1.3-2.el10.x86_64
Installed:
  multilib-7.1.3-2.el10.x86_64
```

```
Complete!
curl: (35) curl: (35) curl: (35)
```

Рис. 3.11: загрузка

Просмотрим логи с помощью multitail на клиенте и на сервере. посмотрим /var/log/messages и /var/log/secure

Рис. 3.12: Просмотр логов с сервера

Рис. 3.13: Просмотр логов с сервера

```

Nov 10 17:49:36 client systemd[1]: setroublehood-service: Consumed 660ms CPU time, 76.8M memory peak
Nov 10 17:49:44 client gnome-shell[11784]: Cursor update failed: drModeAtomicCommit: Invalid argument
Nov 10 17:52:43 client NetworkManager[7171]: <info> [1762797163.4056] dhcpc4 (eth1): state changed new lease, address=192.168.1.30
Nov 10 17:52:43 client NetworkManager[7171]: Starting NetworkManager-dispatcher service - Network Manager Script Dispatcher Service.
Nov 10 17:52:43 client systemd[1]: Started NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 17:52:53 client systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
Nov 10 17:59:13 client NetworkManager[7171]: <info> [1762797553.8261] agent-manager: agent[f720fef8755f2cf,1.215/org.gnome.Shell.NetworkAgent/1001]: agent registered
Nov 10 18:07:12 client systemd[1]: Stopping rsyslog.service - System Logging Service...
Nov 10 18:07:12 client systemd[1]: Stopped rsyslog.service - System Logging Service.
Nov 10 18:07:13 client rsyslogd[410]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] exiting on signal 15
Nov 10 18:07:12 client systemd[1]: rsyslog.service: Deactivated successfully.
Nov 10 18:07:12 client systemd[1]: Stopping rsyslog.service - System Logging Service...
Nov 10 18:07:12 client systemd[1]: Stopped rsyslog.service - System Logging Service.
Nov 10 18:07:13 client rsyslogd[14745]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] start
Nov 10 18:07:13 client rsyslogd[14745]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] start
Nov 10 18:07:43 client NetworkManager[7171]: <info> [1762798063.4624] dhcpc4 (eth1): state changed new lease, address=192.168.1.38
Nov 10 18:07:43 client systemd[1]: Starting NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 18:07:43 client systemd[1]: Started NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 18:07:53 client systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.

```

Рис. 3.14: Просмотр логов с клиента

```

Nov 10 17:52:43 client systemd[1]: Starting NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 17:52:43 client systemd[1]: Started NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 17:59:13 client NetworkManager[7171]: <info> [1762797553.8261] agent-manager: agent[f720fef8755f2cf,1.215/org.gnome.Shell.NetworkAgent/1001]: agent registered
Nov 10 18:07:12 client systemd[1]: Stopping rsyslog.service - System Logging Service...
Nov 10 18:07:12 client rsyslogd[410]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] exiting on signal 15
Nov 10 18:07:12 client systemd[1]: rsyslog.service: Deactivated successfully.
Nov 10 18:07:12 client systemd[1]: Stopping rsyslog.service - System Logging Service...
Nov 10 18:07:12 client systemd[1]: Stopped rsyslog.service - System Logging Service.
Nov 10 18:07:13 client rsyslogd[14745]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] start
Nov 10 18:07:13 client rsyslogd[14745]: [origin software="rsyslogd" swVersion="8.2412.0-1.el10" x-pid="14745" x-info="https://www.rsyslog.com"] start
Nov 10 18:07:43 client NetworkManager[7171]: <info> [1762798063.4624] dhcpc4 (eth1): state changed new lease, address=192.168.1.38
Nov 10 18:07:43 client systemd[1]: Starting NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 18:07:43 client systemd[1]: Started NetworkManager-dispatcher.service - Network Manager Script Dispatcher Service.
Nov 10 18:07:53 client systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
Nov 10 16:21:47 client sudo[12139]: pam_unix(sudo:session): session closed for user sashubina
Nov 10 16:23:17 client sudo[12154]: pam_unix(sudo:session): session opened for user sashubina(uid=1001) by sashubina(uid=0)
Nov 10 16:58:38 client gdm-password[123554]: gkr-pam: unlocked login keyring
Nov 10 16:58:51 client sudo[13056]: pam_unix(sudo:session): session closed for user root
Nov 10 16:59:01 client sudo[13056]: pam_unix(sudo:session): session opened for user root(uid=0) by sashubina(uid=0)
Nov 10 16:59:12 client sudo[13056]: sashubina : TTY-pts/2 : PWD=/root : USER=root : COMMAND=/bin/systemctl restart rsyslog
Nov 10 16:59:12 client sudo[13056]: pam_unix(sudo:session): session opened for user root(uid=0) by sashubina(uid=0)
Nov 10 17:30:02 client gdm-password[14445]: gkr-pam: unlocked login keyring
Nov 10 17:37:25 client gdm-password[14430]: gkr-pam: unlocked login keyring
Nov 10 17:44:29 client sudo[14085]: pam_unix(sudo:session): session closed for user root
Nov 10 17:44:29 client sudo[14085]: pam_unix(sudo:session): session closed for user root
Nov 10 17:44:32 client sudo[14085]: pam_unix(sudo:session): session opened for user root(uid=0) by sashubina(uid=0)
Nov 10 17:44:42 client sudo[14441]: pam_unix(sudo:session): session opened for user root(uid=0) by sashubina(uid=0)
Nov 10 17:44:42 client sudo[14441]: pam_unix(sudo:session): session closed for user root
Nov 10 17:59:11 client gdm-password[14616]: gkr-pam: unlocked login keyring
Nov 10 18:07:12 client sudo[14740]: root : TTY-pts/2 : PWD=/etc/rsyslog.d : USER=root : COMMAND=/bin/systemctl restart rsyslog
Nov 10 18:07:13 client sudo[14740]: pam_unix(sudo:session): session closed for user root
Nov 10 18:07:13 client sudo[14740]: pam_unix(sudo:session): session closed for user root

```

Рис. 3.15: Просмотр логов с клиента

видны логи сервера, где успешно запускаются системные службы, включая `systemd-tmpfiles-setup.service`, работает демон D-Bus через `dbus.socket`, и достигаются системные цели `basic.target` и `default.target`. Важным доказательством работоспособности сетевого журналирования является наличие сообщения от клиента с меткой времени Nov 10 17:59:13, где клиентский NetworkManager регистрирует агента - это прямо подтверждает, что логи передаются с клиента на сервер корректно.

продолжение логов сервера показывает стабильную работу пользовательских сессий для пользователей `sashubina` и `gdm`, функционирование аутентификации PAM, успешный графический вход через `gdm-password`, а также работу SSH-сервера на порту 22. Критически важным является наличие в конце файла

сообщений от клиента, что дополнительно доказывает настройку сетевого журналирования.

демонстрируются логи клиентской машины, где видно, что клиент успешно получает IP-адрес 192.168.1.30 через DHCP, работает NetworkManager, происходит перезапуск службы rsyslog - что указывает на применение изменений конфигурации, и все сетевые службы функционируют штатно.

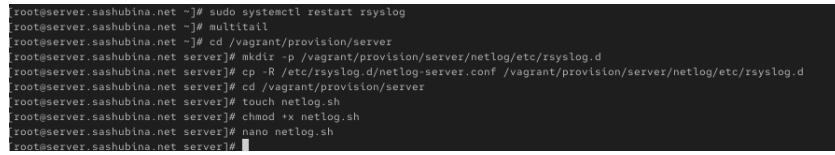
Журнал системы отражает штатную работу многопользовательской среды Linux с графической оболочкой GNOME/GDM, где зафиксированы ключевые события сетевой активности, управления системой и пользовательской работы. Клиент успешно получил сетевые настройки через DHCP с присвоением адреса 192.168.1.30, что инициировало корректную работу сетевых диспетчеров. Административная деятельность включала перезапуск службы журналирования rsyslog пользователем sashubina через sudo для применения изменений конфигурации, а также регулярные операции с правами root. В графической подсистеме отмечена незначительная ошибка буфера обмена в GNOME, не оказывающая влияния на общую работоспособность системы. Анализ логов подтверждает стабильное функционирование всех основных служб, отсутствие критических сбоев и корректную настройку сетевого журналирования, что свидетельствует о успешном выполнении задач лабораторной работы

Критических ошибок в работе сетевого журналирования не обнаружено. Имеются лишь незначительные проблемы, такие как искажение текста в multitail из-за проблем кодировки, мелкие ошибки GNOME, связанные с работой буфера обмена и курсора, временные DNS-ошибки на сервере и единичные сбои распаковки сообщений в rsyslog. Однако основная функция работает корректно: логи успешно передаются от клиента к серверу, сообщения клиента четко видны в журналах сервера, а сетевое подключение остается стабильным. Для целей лабораторной работы задание выполнено успешно, так как все ключевые требования по настройке сетевого журналирования реализованы.

## 3.4 Внесение изменений в настройки внутреннего окружения виртуальных машин

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог netlog, в который поместим в соответствующие подкаталоги конфигурационные файлы, а также создадим исполняемый файл netlog.sh:

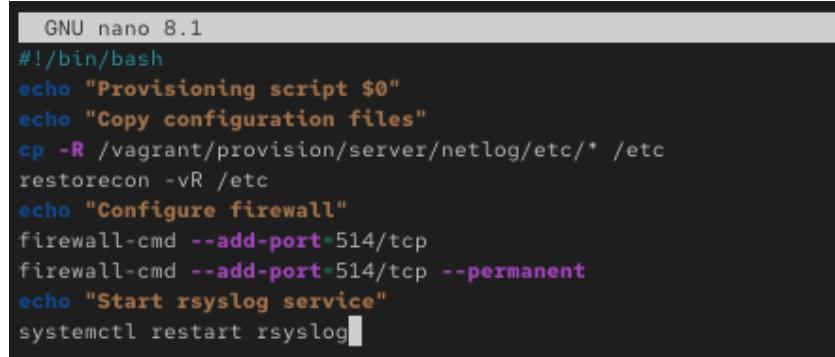
```
cd /vagrant/provision/server  
mkdir -p /vagrant/provision/server/netlog/etc/rsyslog.d  
cp -R /etc/rsyslog.d/netlog-server.conf /vagrant/provision/server/netlog/etc/rsyslog.d  
  
touch netlog.sh  
chmod +x netlog.sh
```



```
[root@server.sashubina.net ~]# sudo systemctl restart rsyslog  
[root@server.sashubina.net ~]# multitail  
[root@server.sashubina.net ~]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# mkdir -p /vagrant/provision/server/netlog/etc/rsyslog.d  
[root@server.sashubina.net server]# cp -R /etc/rsyslog.d/netlog-server.conf /vagrant/provision/server/netlog/etc/rsyslog.d  
[root@server.sashubina.net server]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# touch netlog.sh  
[root@server.sashubina.net server]# chmod +x netlog.sh  
[root@server.sashubina.net server]# nano netlog.sh  
[root@server.sashubina.net server]#
```

Рис. 3.16: создание файла

В каталоге /vagrant/provision/server создадим исполняемый файл netlog.sh и внесем скрипт:



```
GNU nano 8.1  
#!/bin/bash  
echo "Provisioning script $0"  
echo "Copy configuration files"  
cp -R /vagrant/provision/server/netlog/etc/* /etc  
restorecon -vR /etc  
echo "Configure firewall"  
firewall-cmd --add-port 514/tcp  
firewall-cmd --add-port 514/tcp --permanent  
echo "Start rsyslog service"  
systemctl restart rsyslog
```

Рис. 3.17: Скрипта файла /vagrant/provision/server/netlog.sh

На виртуальной машине client перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/client/, создадим в нём каталог netlog, в который поместим в соответствующие подкаталоги конфигурационные файлы, а также создадим исполняемый файл netlog.sh:

```
cd /vagrant/provision//client  
mkdir -p /vagrant/provision//client/netlog/etc/rsyslog.d  
cp -R /etc/rsyslog.d/netlog-/client.conf /vagrant/provision//client/netlog/etc/rsy  
  
touch netlog.sh  
chmod +x netlog.sh  
  
[root@client.sashubina.net rsyslog.d]# cd /vagrant/provision/client  
[root@client.sashubina.net client]# mkdir -p /vagrant/provision/client/n  
etlog/etc/rsyslog.d  
[root@client.sashubina.net client]# cp -R /etc/rsyslog.d/netlog-client.c  
onf /vagrant/provision/client/netlog/etc/rsyslog.d/  
[root@client.sashubina.net client]# cd /vagrant/provision/client  
[root@client.sashubina.net client]# touch netlog.sh  
[root@client.sashubina.net client]# chmod +x netlog.sh
```

Рис. 3.18: создание файла

В каталоге /vagrant/provision/client создадим исполняемый файл netlog.sh и внесем скрипт:

```
GNU nano 8.1          netlog.sh  
#!/bin/bash  
echo "Provisioning script $0"  
echo "Install needed packages"  
dnf -y install multitail  
echo "Copy configuration files"  
cp -R /vagrant/provision/client/netlog/etc/* /etc  
restorecon -vR /etc  
echo "Start rsyslog service"  
systemctl restart rsyslog
```

Рис. 3.19: Скрипта файла /vagrant/provision/client/ netlog.sh

Затем для отработки созданных скриптов в конфигурационном файле Vagrantfile необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

```
server.vm.provision "server netlog",
  type: "shell",
  preserve_order: true,
  path: "provision/server/netlog.sh"

client.vm.provision "client netlog",
  type: "shell",
  preserve_order: true,
  path: "provision/client/netlog.sh"
```

```
server.vm.provision "server netlog",
  type: "shell",
  preserve_order: true,
  path: "provision/server/netlog.sh"
```

```
end
```

Рис. 3.20: Vagrantfile

```
client.vm.provision "client netlog",
  type: "shell",
  preserve_order: true,
  path: "provision/client/netlog.sh"
```

Рис. 3.21: Vagrantfile

## 4 Контрольные вопросы

1. Какой модуль rsyslog вы должны использовать для приёма сообщений от journald?
2. Как называется устаревший модуль, который можно использовать для включения приёма сообщений журнала в rsyslog?
3. Чтобы убедиться, что устаревший метод приёма сообщений из journald в rsyslog не используется, какой дополнительный параметр следует использовать?
4. В каком конфигурационном файле содержатся настройки, которые позволяют вам настраивать работу журнала?
5. Каким параметром управляет пересылка сообщений из journald в rsyslog?
6. Какой модуль rsyslog вы можете использовать для включения сообщений из файла журнала, не созданного rsyslog?
7. Какой модуль rsyslog вам нужно использовать для пересылки сообщений в базу данных MariaDB?
8. Какие две строки вам нужно включить в rsyslog.conf, чтобы позволить текущему журнальному серверу получать сообщения через TCP?
9. Как настроить локальный брандмауэр, чтобы разрешить приём сообщений журнала через порт TCP 514?

10. Для приёма сообщений от journald вам следует использовать модуль imjournal.
11. Устаревший модуль, который можно использовать для включения приёма сообщений журнала в rsyslog, называется imklog.
12. Чтобы убедиться, что устаревший метод приёма сообщений из journald не используется, следует использовать параметр “SystemCallFilter [include:omusrmmsg.conf?]” в конфигурационном файле rsyslog.conf.
13. Настройки, позволяющие настраивать работу журнала, содержатся в конфигурационном файле rsyslog.conf.
14. Пересылка сообщений из journald в rsyslog управляется параметром “ForwardToSyslog” в файле конфигурации journald.conf.
15. Модуль rsyslog, который можно использовать для включения сообщений из файла журнала, не созданного rsyslog, называется imfile.
16. Для пересылки сообщений в базу данных MariaDB вам следует использовать модуль ommysql.
17. Для позволения текущему журнальному серверу получать сообщения через TCP, вам нужно включить две строки в rsyslog.conf:  
`$ModLoad imtcp $InputTCPServerRun 514`

18. Чтобы разрешить приём сообщений журнала через порт TCP 514 можно использовать следующую команду:

```
firewall-cmd --add-port=514/tcp  
firewall-cmd --add-port=514/tcp --permanent
```

## **5 Выводы**

В результате выполнения данной работы были приобретены практические навыки по работе с журналами системных событий.