Лабораторная работа № 15

Настройка сетевого журналирования

Шубина София Антоновна

Содержание

Список иллюстраций

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
```

Рис. 3.1: создание файла

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

Рис. 3.2: Включение журналирования по TCP-порту 514

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

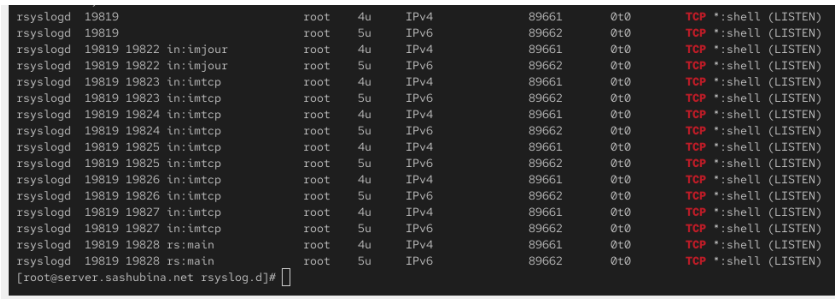

Рис. 3.3: Просмотр прослушиваемых портов, связанных с rsyslog

Порты 89661 и 89662 прослушиваттся rsyslogd - системный демон журналирования

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

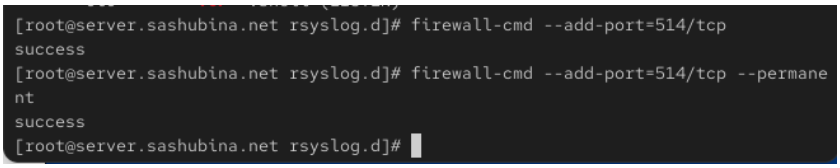

Рис. 3.4: Настройка межсетевого экрана

Проверим 514 порт

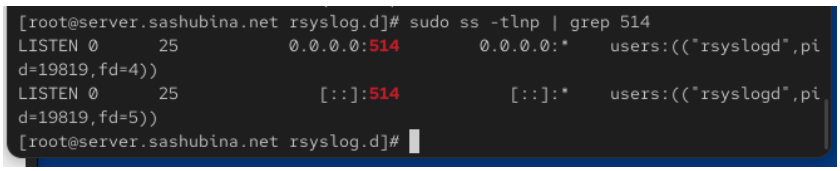

Рис. 3.5: Проверка 514 порта

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:

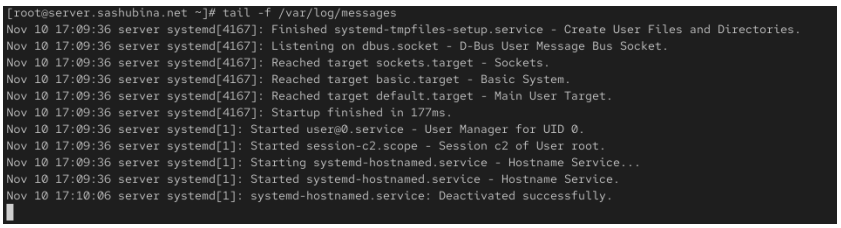

Рис. 3.9: Просмотр файла var/log/messages журнала

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`:

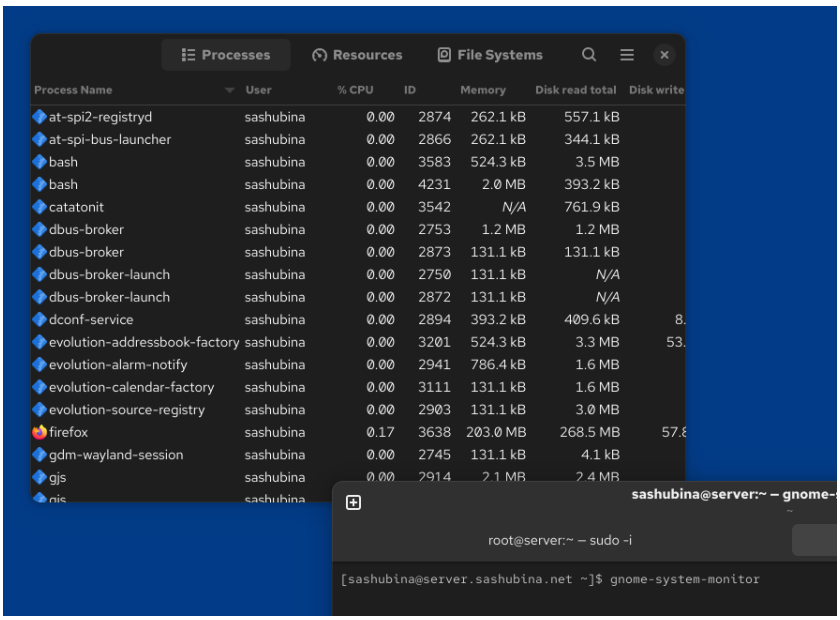

Рис. 3.10: Запуск графической программы для просмотра журналов

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 служб. 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
```

Рис. 3.11: загрузка

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

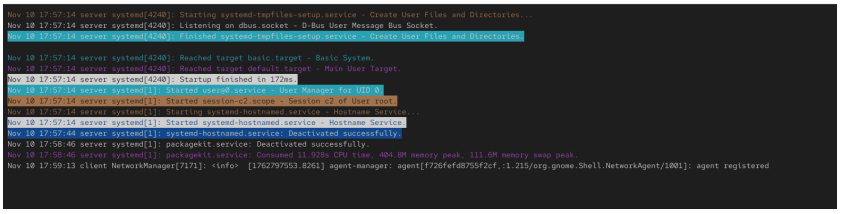

Рис. 3.12: Просмотр логов с сервера

Рис. 3.13: Просмотр логов с сервера

Рис. 3.14: Просмотр логов с клиента

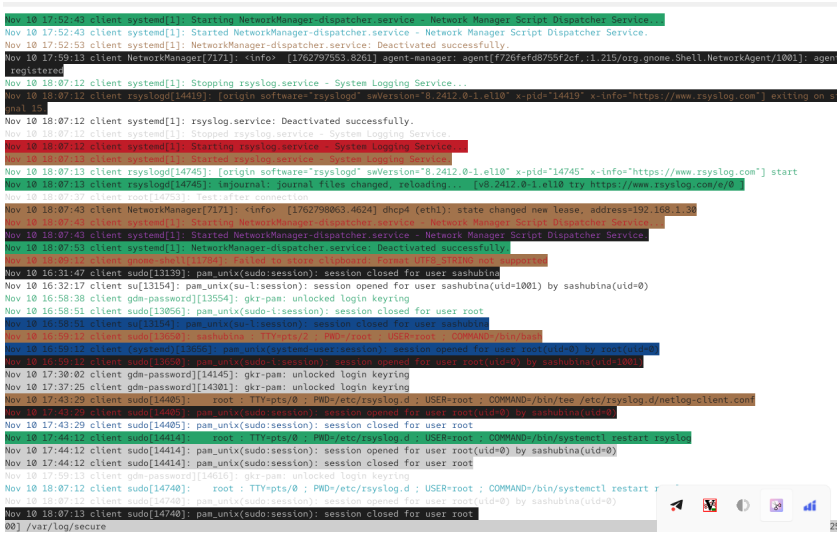

Рис. 3.15: Просмотр логов с клиента

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 машины

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/rsysl

touch netlog.sh
chmod +x netlog.sh
```

Рис. 3.16: создание файла

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

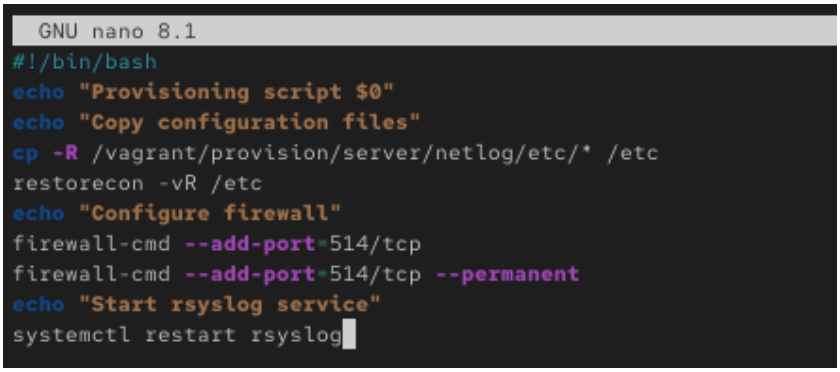

Рис. 3.17: Скрипта файла /vagrant/provision/server/netlog.sh

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"
```

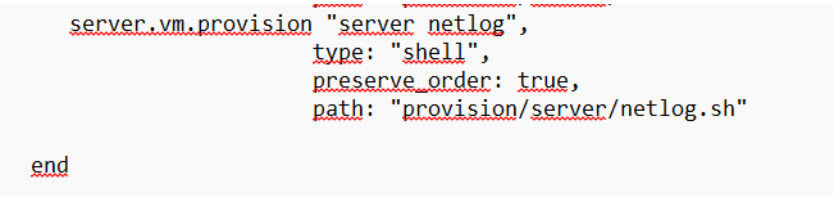

Рис. 3.20: Vagrantfile

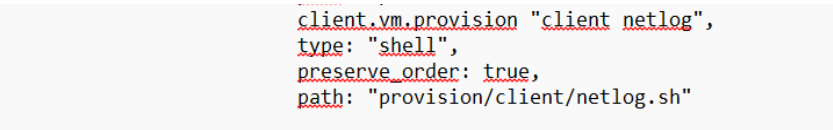

Рис. 3.21: Vagrantfile

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 управляется 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:omusrmsg.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.