

# **Отчёт по лабораторной работе 15**

## **Настройка сетевого журналирования**

Метвалли Ахмед Фарг Набеех

# **Содержание**

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение</b>	<b>6</b>
2.1 Настройка сервера сетевого журнала . . . . .	6
2.1.1 Создание конфигурации сервера . . . . .	6
2.1.2 Перезапуск службы и проверка портов . . . . .	6
2.1.3 Настройка межсетевого экрана . . . . .	7
2.2 Настройка клиента сетевого журнала . . . . .	7
2.3 Просмотр журнала . . . . .	8
2.4 Внесение изменений в Vagrant provisioning . . . . .	9
2.4.1 Настройка provisioning для сервера . . . . .	9
2.4.2 Настройка provisioning для клиента . . . . .	10
<b>3 Заключение</b>	<b>11</b>
<b>4 Контрольные вопросы</b>	<b>12</b>

# **Список иллюстраций**

2.1 Конфигурация netlog-server.conf . . . . .	6
2.2 Порты rsyslog и вывод команд firewall-cmd . . . . .	7
2.3 Конфигурация клиента netlog-client.conf . . . . .	7
2.4 Вывод системных сообщений . . . . .	8
2.5 Графическая утилита мониторинга . . . . .	9
2.6 Скрипт provisioning на сервере . . . . .	10
2.7 Скрипт provisioning на клиенте . . . . .	10

# **Список таблиц**

# **1 Цель работы**

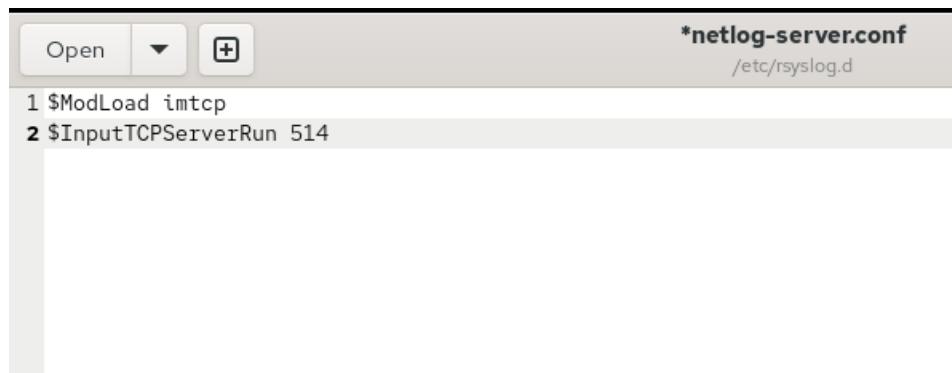
Получение навыков по работе с журналами системных событий.

## 2 Выполнение

### 2.1 Настройка сервера сетевого журнала

#### 2.1.1 Создание конфигурации сервера

- На сервере создан файл конфигурации сетевого приёма журналов:  
переходим в каталог /etc/rsyslog.d и создаём файл netlog-server.conf.
- В файл /etc/rsyslog.d/netlog-server.conf добавлены строки для включения TCP-приёма сообщений:  
загрузка модуля imtcp и запуск TCP-сервера на порту 514.



```
*netlog-server.conf  
/etc/rsyslog.d  
1 $ModLoad imtcp  
2 $InputTCPServerRun 514
```

Рис. 2.1: Конфигурация netlog-server.conf

#### 2.1.2 Перезапуск службы и проверка портов

- Служба rsyslog перезапущена.

2. Выполнена проверка прослушиваемых портов, связанных с rsyslog.

Видно, что процесс слушает TCP-порт 514.

```
rsyslogd 10224 10227 in:imtcp      root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10227 in:imtcp      root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10228 in:imtcp      root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10228 in:imtcp      root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10229 in:imtcp      root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10229 in:imtcp      root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10230 in:imtcp      root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10230 in:imtcp      root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10231 in:imtcp      root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10231 in:imtcp      root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10232 rs:main     root  4u   IPv4      52678    0t0      TCP *:shell (LISTEN)
rsyslogd 10224 10232 rs:main     root  5u   IPv6      52679    0t0      TCP *:shell (LISTEN)

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

Рис. 2.2: Порты rsyslog и вывод команд firewall-cmd

### 2.1.3 Настройка межсетевого экрана

3. На сервере открыт TCP-порт 514 для приёма сообщений.

## 2.2 Настройка клиента сетевого журнала

1. На клиенте создан файл /etc/rsyslog.d/netlog-client.conf.

2. В конфигурацию добавлена строка перенаправления всех сообщений на сервер по TCP-порту 514:

```
*.* @@server.ahmedfarg.net:514
```



Рис. 2.3: Конфигурация клиента netlog-client.conf

3. Служба rsyslog перезапущена.

## 2.3 Просмотр журнала

1. На сервере просмотрен файл `/var/log/messages`.

В логах появляются сообщения от клиента.

```
Nov  5 09:26:28 client systemd[1]: Started systemd-coredump@85-30245-0.service - Process Core Dump (PID 30245/UID 0).
Nov  5 09:26:28 client systemd-coredump[30246]: Process 30241 (VBoxClient) of user 1001 dumped core.#012#012Module libXau
u.so.6 from rpm libXau-1.0.11-8.el10.x86_64#012Module libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64#012Module libX11.
so.6 from rpm libX11-1.8.10-1.el10.x86_64#012Module libffi.so.8 from rpm libffi-3.4.4-9.el10.x86_64#012Module libwayland-
client.so.0 from rpm wayland-1.23.0-2.el10.x86_64#012Stack trace of thread 30244:#012#0 0x000000000041dd1b n/a (n/a +
0x0) #012#1 0x000000000041dc94 n/a (n/a + 0x0) #012#2 0x000000000045041c n/a (n/a + 0x0) #012#3 0x00000000004355d0 n/a (n/a +
0x0) #012#4 0x000007f2fd699b68 start_thread (libc.so.6 + 0x94b68) #012#5 0x000007f2fd670a6bc __clone3 (libc.so.6 +
0x1056bc) #012#012Stack trace of thread 30242:#012#0 0x000007f2fd67084bd syscall (libc.so.6 + 0x1034bd) #012#1 0x000000000
00434c30 n/a (n/a + 0x0) #012#2 0x0000000000450fb n/a (n/a + 0x0) #012#3 0x000000000043566a n/a (n/a + 0x0) #012#4 0x00
0000000045041c n/a (n/a + 0x0) #012#5 0x00000000004355d0 n/a (n/a + 0x0) #012#6 0x000007f2fd699b68 start_thread (libc.so.
6 + 0x94b68) #012#7 0x000007f2fd670a6bc __clone3 (libc.so.6 + 0x1056bc) #012#012Stack trace of thread 30243:#012#0 0x000
07f2fd67084bd syscall (libc.so.6 + 0x1034bd) #012#1 0x00000000004344e2 n/a (n/a + 0x0) #012#2 0x0000000000450066 n/a (n/
a + 0x0) #012#3 0x0000000000416559 n/a (n/a + 0x0) #012#4 0x0000000000418338a n/a (n/a + 0x0) #012#5 0x0000000000417d6a n
/a (n/a + 0x0) #012#6 0x0000000000404860 n/a (n/a + 0x0) #012#7 0x000007f2fd6699b68 start_thread (libc.so.6 + 0x94b68) #012#8 0x000000000043
55d0 n/a (n/a + 0x0) #012#9 0x000007f2fd6699b68 start_thread (libc.so.6 + 0x94b68) #012#10 0x000007f2fd670a6bc __clone3 (li
bc.so.6 + 0x1056bc) #012#012Stack trace of thread 30241:#012#0 0x000007f2fd67084bd syscall (libc.so.6 + 0x1034bd) #012#1
0x00000000004344e2 n/a (n/a + 0x0) #012#2 0x0000000000450066 n/a (n/a + 0x0) #012#3 0x0000000000405123 n/a (n/a + 0x0) #0
12#4 0x000007f2fd662f30# __libc_start_call_main (libc.so.6 + 0x2a30e) #012#5 0x000007f2fd662f3c9 __libc_start_main@GLIBC
_2.34 (libc.so.6 + 0x2a3c9) #012#6 0x00000000004044aa n/a (n/a + 0x0) #012ELF object binary architecture: AMD x86-64
Nov  5 09:26:28 client systemd[1]: traps: deactivate successfully.
Nov  5 09:26:31 server kernel[1]: traps: VBoxClient[10726] trap int3 ip:41dd1b sp:7f4ab3e35cd0 error:0 in VBoxClient[1dd1b,
400000+bb000]
Nov  5 09:26:31 server systemd-coredump[10727]: Process 10723 (VBoxClient) of user 1001 terminated abnormally with signa
l 5/TRAP, processing...
Nov  5 09:26:31 server systemd[1]: Started systemd-coredump@115-10727-0.service - Process Core Dump (PID 10727/UID 0).
```

Рис. 2.4: Вывод системных сообщений

2. Запущена графическая программа для просмотра системных журналов.

Показаны загрузка CPU, память и сетевые показатели.



Рис. 2.5: Графическая утилита мониторинга

3. На сервер установлен просмотрщик системных логов lnav.

## 2.4 Внесение изменений в Vagrant provisioning

### 2.4.1 Настройка provisioning для сервера

- В каталоге /vagrant/provision/server создан каталог netlog, содержащий подкаталог etc/rsyslog.d, куда помещён конфигурационный файл netlog-server.conf.
- Создан исполняемый файл netlog.sh, содержащий:  
копирование конфигурации, восстановление контекстов SELinux, настройку firewall и перезапуск rsyslog.

```
1 #!/bin/bash
2 echo "Provisioning script $0"
3 echo "Copy configuration files"
4 cp -R /vagrant/provision/server/netlog/etc/* /etc
5 restorecon -vR /etc
6 echo "Configure firewall"
7 firewall-cmd --add-port=514/tcp
8 firewall-cmd --add-port=514/tcp --permanent
9 echo "Start rsyslog service"
10 systemctl restart rsyslog
```

Рис. 2.6: Скрипт provisioning на сервере

#### 2.4.2 Настройка provisioning для клиента

1. В каталоге `/vagrant/provision/client` создан каталог `netlog`, куда помещён файл `netlog-client.conf`.
2. Создан исполняемый файл `netlog.sh`, обеспечивающий копирование конфигураций, восстановление контекстов и перезапуск `rsyslog`.

```
1 #!/bin/bash
2 echo "Provisioning script $0"
3 echo "Install needed packages"
4 #dnf -y install lnav
5 echo "Copy configuration files"
6 cp -R /vagrant/provision/client/netlog/etc/* /etc
7 restorecon -vR /etc
8 echo "Start rsyslog service"
9 systemctl restart rsyslog
10
```

Рис. 2.7: Скрипт provisioning на клиенте

## **3 Заключение**

Сетевой журнал был успешно настроен: сервер принял конфигурацию для получения сообщений по TCP, клиент корректно перенаправляет логи, взаимодействие проверено просмотром системных сообщений. Дополнительно подготовлены provisioning-скрипты для автоматизации развёртывания обеих виртуальных машин.

## 4 Контрольные вопросы

1. **Какой модуль rsyslog вы должны использовать для приёма сообщений от journald?** Для интеграции journald с rsyslog используется модуль **imjournal**.
2. **Как называется устаревший модуль, который можно использовать для включения приёма сообщений журнала в rsyslog?** Устаревший модуль – **imuxsock**.
3. **Чтобы убедиться, что устаревший метод приёма сообщений из journald в rsyslog не используется, какой дополнительный параметр следует применить?** Нужно добавить параметр: **UseUxSock off**
4. **В каком конфигурационном файле содержатся настройки, которые позволяют настраивать работу журнала?** Основные параметры работы journald находятся в файле **/etc/systemd/journald.conf**.
5. **Каким параметром управляется пересылка сообщений из journald в rsyslog?** За пересылку отвечает параметр journald: **ForwardToSyslog=**
6. **Какой модуль rsyslog можно использовать для включения сообщений из файла журнала, не созданного rsyslog?** Для чтения любых лог-файлов используется модуль **imfile**.
7. **Какой модуль rsyslog нужно использовать для пересылки сообщений в базу данных MariaDB?** Для записи сообщений в MariaDB применяют модуль **ommysql**.

- 8. Какие две строки нужно включить в rsyslog.conf, чтобы позволить текущему журнальному серверу принимать сообщения через TCP?**

```
$ModLoad imtcp  
$InputTCPServerRun 514
```

- 9. Как настроить локальный брандмауэр, чтобы разрешить приём сообщений журнала через порт TCP 514?** Нужно открыть порт командой firewall-cmd:

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