

Лабораторная работа №15

Презентация

Мосолов А.Д.

13 декабря 2025

Российский университет дружбы народов, Москва, Россия

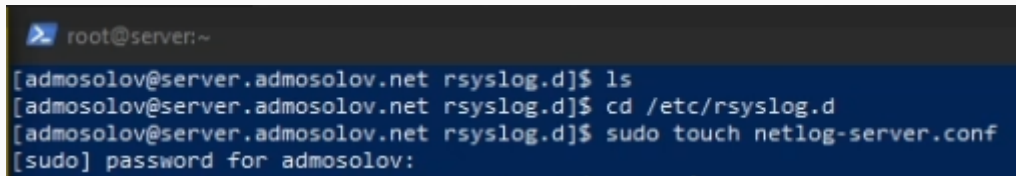
- Мосолов Александр Денисович
- Студент, НПИбд02-23
- Российский университет дружбы народов
- 1132236128@pfur.ru

Получение навыков по работе с журналами системных событий.

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.
4. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сетевого сервера журналирования.

Создание файла конфигурации на сервере

Начинаю настройку сервера сетевого журнала. Для этого перехожу в каталог конфигурации rsyslog и создаю файл `netlog-server.conf`, который будет содержать настройки для приема логов по сети.

A terminal window with a dark background. The prompt is 'root@server:~'. The user runs 'ls' in the directory '/etc/rsyslog.d', then 'cd /etc/rsyslog.d', and finally 'sudo touch netlog-server.conf'. A password prompt '[sudo] password for admosolov:' is shown.

```
root@server:~  
[admosolov@server.admosolov.net rsyslog.d]$ ls  
[admosolov@server.admosolov.net rsyslog.d]$ cd /etc/rsyslog.d  
[admosolov@server.admosolov.net rsyslog.d]$ sudo touch netlog-server.conf  
[sudo] password for admosolov:
```

Рис. 1: Создание файла конфигурации на сервере

Настройка приема логов по TCP

Редактирую созданный файл конфигурации `/etc/rsyslog.d/netlog-server.conf`. Включаю модуль `imtcp` для поддержки протокола TCP и указываю прослушивание порта 514. Это стандартный порт для службы `syslog` при использовании надежного транспортного протокола.

A screenshot of a terminal window. The top bar shows the prompt 'root@server:/etc/rsyslog.d' and the file name 'netlog-server.conf'. The main area shows the GNU nano 8.1 editor with the following content: '\$ModLoad imtcp' and '\$InputTCPServerRun 514'. The cursor is at the end of the second line.

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

Рис. 2: Настройка приема логов по TCP

Перезапуск службы rsyslog

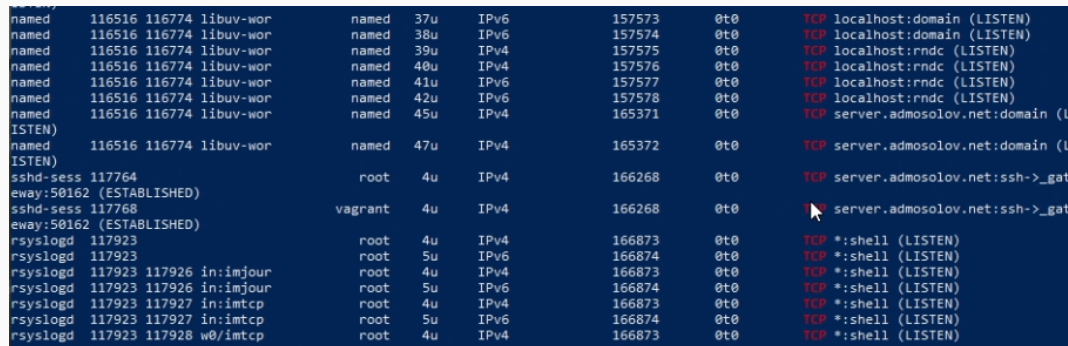
После внесения изменений в конфигурацию необходимо перезапустить службу rsyslog, чтобы новые параметры вступили в силу.

```
[root@server.admosolov.net rsyslog.d]# systemctl restart rsyslog
```

Рис. 3: Перезапуск службы rsyslog

Проверка открытых портов

Проверяю, что служба действительно начала слушать нужный порт. Использую команду `lsof` с фильтрацией по TCP. Видно, что процесс `rsyslogd` прослушивает порт 514 (обозначен как `shell` в выводе `lsof` для этого порта).

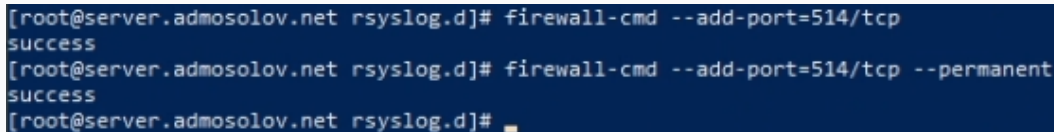


named	116516	116774	libuv-wor	named	37u	IPv6	157573	0t0	TCP	localhost:domain (LISTEN)
named	116516	116774	libuv-wor	named	38u	IPv6	157574	0t0	TCP	localhost:domain (LISTEN)
named	116516	116774	libuv-wor	named	39u	IPv4	157575	0t0	TCP	localhost:rndc (LISTEN)
named	116516	116774	libuv-wor	named	40u	IPv4	157576	0t0	TCP	localhost:rndc (LISTEN)
named	116516	116774	libuv-wor	named	41u	IPv6	157577	0t0	TCP	localhost:rndc (LISTEN)
named	116516	116774	libuv-wor	named	42u	IPv6	157578	0t0	TCP	localhost:rndc (LISTEN)
named	116516	116774	libuv-wor	named	45u	IPv4	165371	0t0	TCP	server.admosolov.net:domain (LISTEN)
named	116516	116774	libuv-wor	named	47u	IPv4	165372	0t0	TCP	server.admosolov.net:domain (LISTEN)
sshd-session	117764			root	4u	IPv4	166268	0t0	TCP	server.admosolov.net:ssh->_gate
sshd-session	117768			vagrant	4u	IPv4	166268	0t0	TCP	server.admosolov.net:ssh->_gate
rsyslogd	117923			root	4u	IPv4	166873	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923			root	5u	IPv6	166874	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923	117926	in:imjour	root	4u	IPv4	166873	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923	117926	in:imjour	root	5u	IPv6	166874	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923	117927	in:imtcp	root	4u	IPv4	166873	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923	117927	in:imtcp	root	5u	IPv6	166874	0t0	TCP	*:shell (LISTEN)
rsyslogd	117923	117928	w0/imtcp	root	4u	IPv4	166873	0t0	TCP	*:shell (LISTEN)

Рис. 4: Проверка открытых портов

Настройка Firewall на сервере

Чтобы сервер мог принимать соединения извне, настраиваю межсетевой экран (firewall). Добавляю правило, разрешающее входящий трафик на порт 514/tcp, и закрепляю его как постоянное (--permanent).

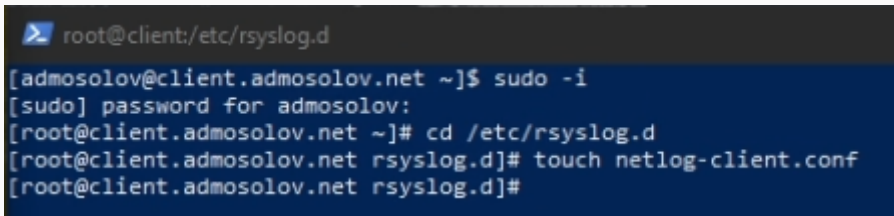
A terminal window with a dark blue background and white text. It shows three lines of commands and their outputs. The first line is '[root@server.admosolov.net rsyslog.d]# firewall-cmd --add-port=514/tcp' followed by 'success'. The second line is '[root@server.admosolov.net rsyslog.d]# firewall-cmd --add-port=514/tcp --permanent' followed by 'success'. The third line is '[root@server.admosolov.net rsyslog.d]#' followed by a cursor.

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

Рис. 5: Настройка Firewall на сервере

Создание конфигурации на клиенте

Перехожу к настройке клиента. На виртуальной машине клиента захожу под суперпользователем, перехожу в каталог конфигурации rsyslog и создаю файл netlog-client.conf.

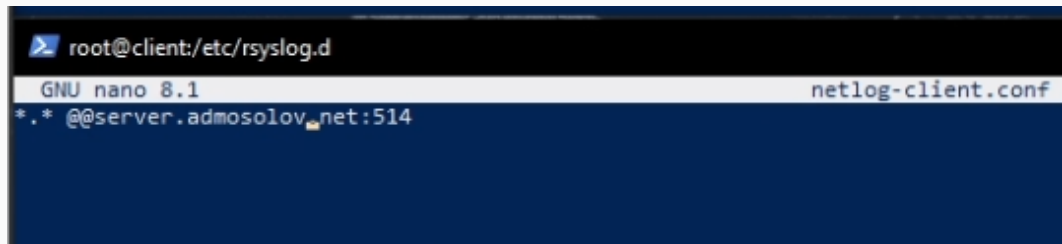
A terminal window with a dark blue background and white text. The prompt is root@client:/etc/rsyslog.d. The user runs 'sudo -i', enters the password for 'admosolov', and then runs 'cd /etc/rsyslog.d' and 'touch netlog-client.conf'.

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

Рис. 6: Создание конфигурации на клиенте

Настройка пересылки логов на сервер

В файле конфигурации клиента прописываю правило пересылки всех логов (*.*) на удаленный сервер. Использую формат `@server.admosolov.net:514` (две «собачки» означают использование протокола TCP).

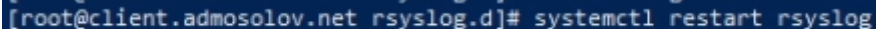


```
root@client:/etc/rsyslog.d
GNU nano 8.1 netlog-client.conf
*. * @@server.admosolov.net:514
```

Рис. 7: Настройка пересылки логов на сервер

Перезапуск rsyslog на клиенте

Для применения настроек перезапускаю службу журналирования на клиенте командой `systemctl restart rsyslog`.

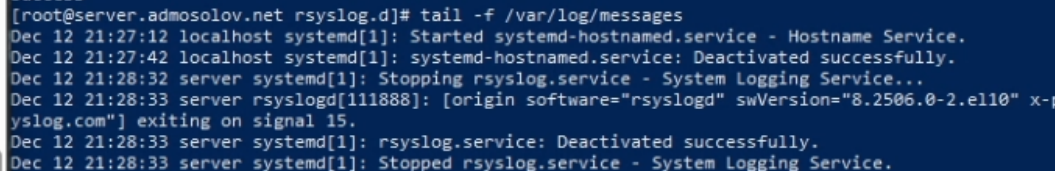
A terminal window with a dark blue background and white text. The prompt is [root@client.admosolov.net rsyslog.d]# and the command entered is systemctl restart rsyslog.

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

Рис. 8: Перезапуск rsyslog на клиенте

Мониторинг логов на сервере

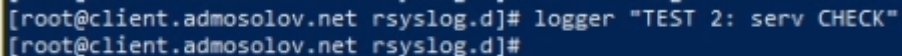
Возвращаюсь на сервер для проверки получения логов. Запускаю команду `tail -f /var/log/messages` для просмотра сообщений в реальном времени. В логе видны записи о перезапуске службы.



```
[root@server.admosolov.net rsyslog.d]# tail -f /var/log/messages
Dec 12 21:27:12 localhost systemd[1]: Started systemd-hostnamed.service - Hostname Service.
Dec 12 21:27:42 localhost systemd[1]: systemd-hostnamed.service: Deactivated successfully.
Dec 12 21:28:32 server systemd[1]: Stopping rsyslog.service - System Logging Service...
Dec 12 21:28:33 server rsyslogd[111888]: [origin software="rsyslogd" swVersion="8.2506.0-2.el10" x-p
yslog.com"] exiting on signal 15.
Dec 12 21:28:33 server systemd[1]: rsyslog.service: Deactivated successfully.
Dec 12 21:28:33 server systemd[1]: Stopped rsyslog.service - System Logging Service.
```

Рис. 9: Мониторинг логов на сервере

Чтобы убедиться в работоспособности сетевого журналирования, отправляю тестовое сообщение с клиента с помощью утилиты `logger`. Текст сообщения: “TEST 2: serv CHECK”.

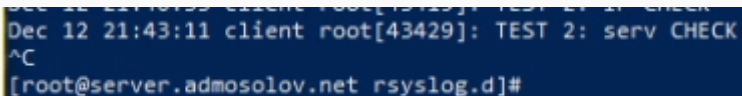
A terminal window with a dark blue background and light blue text. The prompt is [root@client.admosolov.net rsyslog.d]#. The command entered is logger "TEST 2: serv CHECK". The prompt is repeated on the next line.

```
[root@client.admosolov.net rsyslog.d]# logger "TEST 2: serv CHECK"  
[root@client.admosolov.net rsyslog.d]#
```

Рис. 10: Отправка тестового сообщения

Получение тестового сообщения на сервере

Проверяю журнал на сервере. Вижу, что сообщение, отправленное с хоста `client`, успешно записалось в файл `/var/log/messages` на сервере. Это подтверждает корректность настройки.

A screenshot of a terminal window with a dark blue background and light blue text. The text shows a log entry: 'Dec 12 21:43:11 client root[43429]: TEST 2: serv CHECK'. Below this, there is a carriage return and the prompt '[root@server.admosolov.net rsyslog.d]#'.

```
Dec 12 21:43:11 client root[43429]: TEST 2: serv CHECK  
^C  
[root@server.admosolov.net rsyslog.d]#
```

Рис. 11: Получение тестового сообщения на сервере

Запуск `gnome-system-monitor`

Перехожу к заданию по использованию графических утилит. На клиенте запускаю `gnome-system-monitor` (Системный монитор) для визуального наблюдения за процессами.

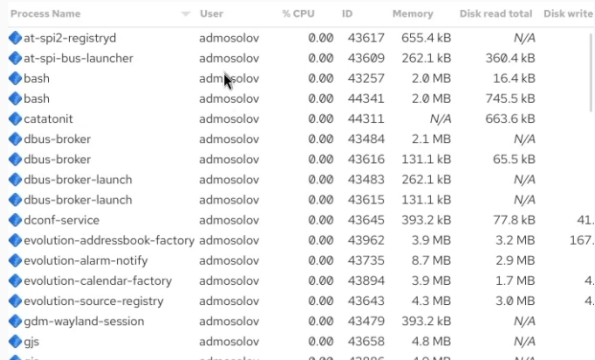
A terminal window titled "admosolov@client:~ – gnome-system-monitor". The terminal shows the command "[admosolov@client.admosolov.net ~]\$ gnome-system-monitor" being entered, with a cursor at the end of the line.

```
admosolov@client:~ – gnome-system-monitor  
[admosolov@client.admosolov.net ~]$ gnome-system-monitor
```

Рис. 12: Запуск `gnome-system-monitor`

Интерфейс системного монитора

В открывшемся окне системного монитора просматриваю список запущенных процессов, потребление ресурсов (CPU, память). Это позволяет отслеживать активность системы в графическом режиме.



The screenshot shows the 'System Monitor' window with the 'Processes' tab selected. It displays a table of running processes with columns for Process Name, User, % CPU, ID, Memory, Disk read total, and Disk write. The processes listed include system services like at-spi2-registryd, dbus-broker, and gdm-wayland-session, as well as user applications like bash and evolution-addressbook-factory. A mouse cursor is visible over the 'bash' process row.

Process Name	User	% CPU	ID	Memory	Disk read total	Disk write
at-spi2-registryd	admosolov	0.00	43617	655.4 kB	N/A	
at-spi-bus-launcher	admosolov	0.00	43609	262.1 kB	360.4 kB	
bash	admosolov	0.00	43257	2.0 MB	16.4 kB	
bash	admosolov	0.00	44341	2.0 MB	745.5 kB	
catatonit	admosolov	0.00	44311	N/A	663.6 kB	
dbus-broker	admosolov	0.00	43484	2.1 MB	N/A	
dbus-broker	admosolov	0.00	43616	131.1 kB	65.5 kB	
dbus-broker-launch	admosolov	0.00	43483	262.1 kB	N/A	
dbus-broker-launch	admosolov	0.00	43615	131.1 kB	N/A	
dconf-service	admosolov	0.00	43645	393.2 kB	77.8 kB	41.
evolution-addressbook-factory	admosolov	0.00	43962	3.9 MB	3.2 MB	167.
evolution-alarm-notify	admosolov	0.00	43735	8.7 MB	2.9 MB	
evolution-calendar-factory	admosolov	0.00	43894	3.9 MB	1.7 MB	4.
evolution-source-registry	admosolov	0.00	43643	4.3 MB	3.0 MB	4.
gdm-wayland-session	admosolov	0.00	43479	393.2 kB	N/A	
gjs	admosolov	0.00	43658	4.8 MB	N/A	
nautilus	admosolov	0.00	43885	4.0 MB	N/A	

Рис. 13: Интерфейс системного монитора

Просмотр логов через journalctl

Также просматриваю журналы событий на сервере с помощью утилиты `journalctl`. Использую флаг `-f` для отслеживания новых записей. В логе отображаются действия менеджера пакетов `PackageKit`.

```
[admosolov@server.admosolov.net rsyslog.d]$ sudo journalctl -f
Dec 12 21:49:17 server.admosolov.net PackageKit[117987]: search-file transaction /6_bddcadda from uid 1001 finished with success after 39ms
Dec 12 21:49:21 server.admosolov.net PackageKit[117987]: new install-packages transaction /7_cd9bbceeb scheduled from uid 1001
Dec 12 21:49:21 server.admosolov.net PackageKit[117987]: in /7_cd9bbceeb for install-packages package squashfs-tools;4.6.1-6.el10;x86_64;baseos was installing for uid 1001
Dec 12 21:49:21 server.admosolov.net PackageKit[117987]: in /7_cd9bbceeb for install-packages package snap-confine;2.70-1.el10_1;x86_64;epel was installing for uid 1001
Dec 12 21:49:21 server.admosolov.net PackageKit[117987]: in /7_cd9bbceeb for install-packages package snapd;2.70-1.el10_1;x86_64;epel was installing for uid 1001
```

Рис. 14: Просмотр логов через `journalctl`

Подготовка каталогов для провижининга сервера

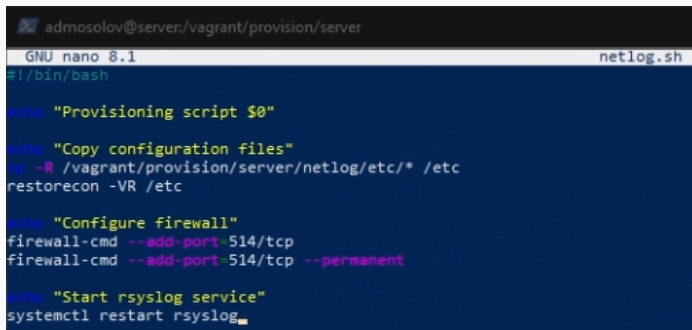
Приступаю к автоматизации процесса настройки через Vagrant. На сервере создаю структуру каталогов внутри `/vagrant/provision/server/` для хранения конфигурационных файлов и копирую туда текущий конфиг `netlog-server.conf`.

```
[admosolov@server.admosolov.net rsyslog.d]$ cd /vagrant/provision/server
[admosolov@server.admosolov.net server]$ mkdir -p /vagrant/provision/server/netlog/etc/rsyslog.d
[admosolov@server.admosolov.net server]$ cp /etc/rsyslog.d/netlog-server.conf /vagrant/provision/server/netlog/etc/rsyslog.d/
[admosolov@server.admosolov.net server]$ cd /vagrant/provision/server
[admosolov@server.admosolov.net server]$ touch netlog.sh
[admosolov@server.admosolov.net server]$ chmod +x netlog.sh
[admosolov@server.admosolov.net server]$
```

Рис. 15: Подготовка каталогов для провижининга сервера

Скрипт автоматизации для сервера

Создаю скрипт `netlog.sh` для сервера. В нем прописываю команды копирования конфигурации в `/etc/`, восстановления контекста безопасности SELinux, настройки firewall и перезапуска службы `rsyslog`.



```
admosolov@server:/vagrant/provision/server
GNU nano 8.1 netlog.sh
#!/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. 16: Скрипт автоматизации для сервера

Подготовка каталогов для провижининга клиента

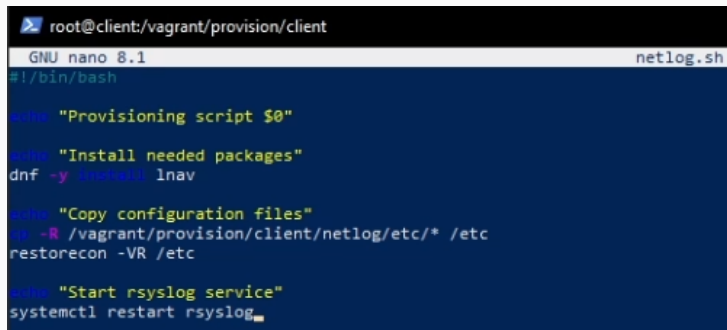
Аналогичные действия выполняю для клиента. Создаю структуру папок в `/vagrant/provision/client/` и копирую туда настроенный файл `netlog-client.conf`.

```
[root@client.admosolov.net rsyslog.d]# cd /vagrant/provision/client
[root@client.admosolov.net client]# mkdir -p /vagrant/provision/client/netlog/etc/rsyslog.d
[root@client.admosolov.net client]# cp /etc/rsyslog.d/netlog-client.conf /vagrant/provision/client/netlog/etc/rsyslog.d/
[root@client.admosolov.net client]# touch netlog.sh
[root@client.admosolov.net client]# chmod +x netlog.sh
[root@client.admosolov.net client]# nano netlog.sh
```

Рис. 17: Подготовка каталогов для провижининга клиента

Скрипт автоматизации для клиента

Создаю скрипт `netlog.sh` для клиента. Он включает установку утилиты `lnav` (для удобного просмотра логов), копирование конфигурации, настройку SELinux и перезапуск службы.



```
root@client:/vagrant/provision/client
GNU nano 8.1 netlog.sh
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
dnf -y install lnav

echo "Copy configuration files"
cp -R /vagrant/provision/client/netlog/etc/* /etc
restorecon -VR /etc

echo "Start rsyslog service"
systemctl restart rsyslog
```

Рис. 18: Скрипт автоматизации для клиента

Настройка Vagrantfile для сервера

В завершение, вношу изменения в основной файл Vagrantfile для виртуальной машины server. Добавляю секцию provision типа shell, указывающую на созданный скрипт provision/server/netlog.sh.

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

Рис. 19: Настройка Vagrantfile для сервера

Настройка Vagrantfile для клиента

Также добавляю секцию `provision` в Vagrantfile для виртуальной машины `client`, указывая путь к скрипту `provision/client/netlog.sh`. Теперь настройка журналирования будет выполняться автоматически при развертывании машин.

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

Рис. 20: Настройка Vagrantfile для клиента

В ходе выполнения лабораторной работы я освоил принципы централизованного сбора логов в Linux. Я настроил сервер rsyslog для приема сообщений по протоколу TCP и клиент для их отправки. Также я научился анализировать логи с помощью `tail`, `journalctl` и графических утилит, а также написал скрипты автоматизации настройки окружения для Vagrant.