

# **Лабораторная работа №7**

**Расширенные настройки межсетевого экрана**

Демидова Екатерина Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Создание пользовательской службы firewalld . . . . .	6
3.2	Перенаправление портов . . . . .	9
3.3	Настройка Port Forwarding и Masquerading . . . . .	9
3.4	Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	10
<b>4</b>	<b>Контрольные вопросы</b>	<b>13</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>

## Список иллюстраций

3.1	Содержание файла службы ssh . . . . .	6
3.2	Редактирование файла службы SSh . . . . .	7
3.3	Список доступным FirewoIID служб . . . . .	8
3.4	Добавление новой службы и её активация . . . . .	8
3.5	Получение клиентом удаленного доступа по SSH к серверу через порт 2022 . . . . .	9
3.6	Настройка перенаправления IPv4-пакетов и включение маскардинга	10
3.7	Создание окружения для внесения изменений в настройки окружающей среды . . . . .	11
3.8	Содержание firewall.sh . . . . .	11
3.9	Изменение файла Vagrantfile . . . . .	12

# 1 Цель работы

Получить навыки настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.

## 2 Задание

1. Настройте межсетевой экран виртуальной машины `server` для доступа к серверу по протоколу SSH не через 22-й порт, а через порт 2022.
2. Настройте Port Forwarding на виртуальной машине `server`.
3. Настройте маскардинг на виртуальной машине `server` для организации доступа клиента к сети Интернет.
4. Напишите скрипт для Vagrant, фиксирующий действия по расширенной настройке межсетевого экрана. Соответствующим образом внести изменения в Vagrantfile

## 3 Выполнение лабораторной работы

### 3.1 Создание пользовательской службы firewalld

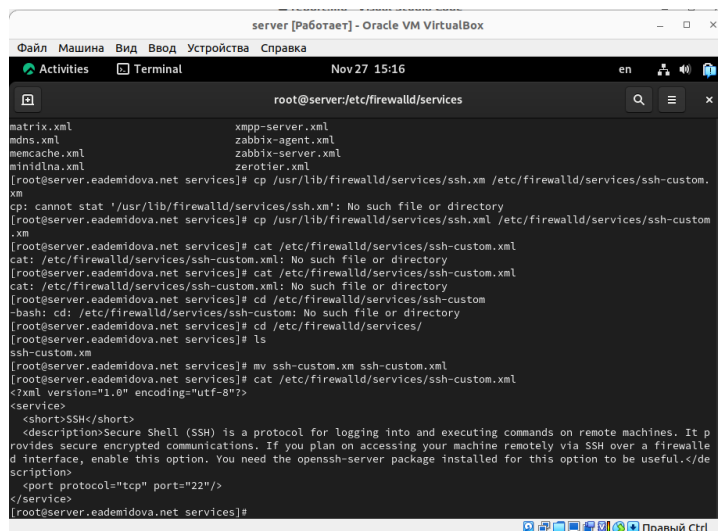
Загрузим нашу операционную систему и перейдем в рабочий каталог с проектом:

```
cd /var/tmp/eademidova/vagran
```

Затем запустим виртуальную машину server:

```
make server-up
```

На основе существующего файла описания службы ssh создадим файл с собственным описанием, посмотрим его содержимое(рис. 3.1):



```
server [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Activities  Terminal  Nov 27 15:16  en  [Icons]
root@server:/etc/firewalld/services

matrix.xml          xmpp-server.xml
mdns.xml            zabbix-agent.xml
memcached.xml       zabbix-server.xml
minidlna.xml        zerotier.xml
[root@server.eademidova.net services]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml
cp: cannot stat '/usr/lib/firewalld/services/ssh.xml': No such file or directory
[root@server.eademidova.net services]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml
cat: /etc/firewalld/services/ssh-custom.xml: No such file or directory
[root@server.eademidova.net services]# cat /etc/firewalld/services/ssh-custom.xml
cat: /etc/firewalld/services/ssh-custom.xml: No such file or directory
[root@server.eademidova.net services]# cd /etc/firewalld/services/ssh-custom
-bash: cd: /etc/firewalld/services/ssh-custom: No such file or directory
[root@server.eademidova.net services]# cd /etc/firewalld/services/
[root@server.eademidova.net services]# ls
ssh-custom.xml
[root@server.eademidova.net services]# mv ssh-custom.xml ssh-custom.xml
[root@server.eademidova.net services]# cat /etc/firewalld/services/ssh-custom.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote machines. It provides secure encrypted communications. If you plan on accessing your machine remotely via SSH over a firewall interface, enable this option. You need the openssh-server package installed for this option to be useful.</description>
  <port protocol="tcp" port="22"/>
</service>
[root@server.eademidova.net services]#
```

Рис. 3.1: Содержание файла службы ssh

В первой строчке этого файла указана версия xml и используемая кодировка - utf8. Затем указаны тег service, а внутри его тег-потомок short, внутри которого указано SSH. Также внутри указан тег description, внутри которого написано описание протокола ssh, и указан протокол передачи порта tcp и н номер порта 22.

Откроем файл описания службы на редактирование и заменим порт 22 на новый порт (2022) и скорректируем описание службы(рис. 3.2):

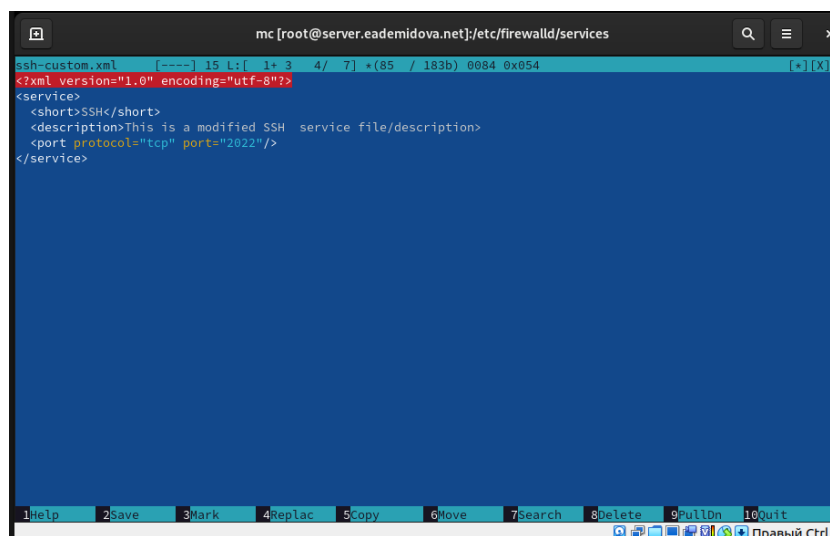


Рис. 3.2: Редактирование файла службы SSH

Посмотрим список доступных FirewallD служб(3.3):

```
root@server:/etc/firewalld/services

<short>SSH</short>
<description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote machines. It provides secure encrypted communications. If you plan on accessing your machine remotely via SSH over a firewall interface, enable this option. You need the openssh-server package installed for this option to be useful.</description>
<port protocol="tcp" port="22"/>
</service>
[root@server.eadimidova.net services]# mc

[root@server.eadimidova.net services]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon ceph-fengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-ov er-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gssd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc irc-cs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogon kpasswd kprop kshell kube-api kube-e-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker-ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-tcp llmnr-udp managesieve matrix mdns memcached mini-dlna mongod mosh mounstd mqtqt mqtqt-tls ms-wbt mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pcmd pmproxy pmwebapi pmwebapi-s pop3 pop3s postgresql proxysql prometheus prometheus-node-exporter proxy-dhcp ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquodad rsh rsyncd rtsp salt-master samba samba-client samba-ba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spidiroak-lansync spotify-sync squid ssdp ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vds vnc-server wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.eadimidova.net services]#
```

Рис. 3.3: Список доступным Firewalld служб

В этом списке нет новой службы. Теперь перезагрузим правила межмететевого экрана с сохранением информации о состоянии и вновь выведем на экран список служб, а также список активных служб. Новая служба отображается в списке доступных служб, но не активирована. Затем активируем новую службу в FirewallD и выведем на экран список активных служб(рис. 3.4):

```
root@server:/etc/firewalld/services

[root@server.eadimidova.net services]# firewall-cmd --reload
success
[root@server.eadimidova.net services]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2 bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon ceph-fengine checkmk-agent cockpit collectd condor-collector cratedb ctdb dhcp dhcpv6 dhcpv6-client distcc dns dns-ov er-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gssd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc irc-cs iscsi-target isns jellyfin jenkins kadmin kdeconnect kerberos kibana klogon kpasswd kprop kshell kube-api kube-e-apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker-ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-tcp llmnr-udp managesieve matrix mdns memcached mini-dlna mongod mosh mounstd mqtqt mqtqt-tls ms-wbt mssql murmur mysql nbd netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pcmd pmproxy pmwebapi pmwebapi-s pop3 pop3s postgresql proxysql prometheus prometheus-node-exporter proxy-dhcp ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquodad rsh rsyncd rtsp salt-master samba samba-client samba-ba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spidiroak-lansync spotify-sync squid ssdp ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-gui synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client vds vnc-server wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server zerotier
[root@server.eadimidova.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh
[root@server.eadimidova.net services]# firewall-cmd --add-service=ssh-custom
success
[root@server.eadimidova.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh ssh-custom
[root@server.eadimidova.net services]#
```

Рис. 3.4: Добавление новой службы и её активация



## 3.2 Перенаправление портов

Организуем на сервере переадресацию с порта 2022 на порт 22 с помощью команды:

```
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
```

На клиенте попробуем получить доступ по SSH к серверу через порт 2022(рис. 3.5):

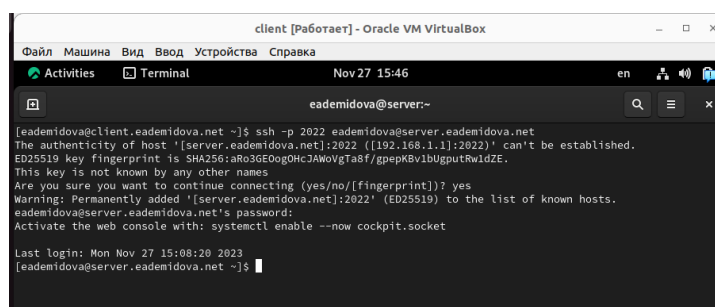
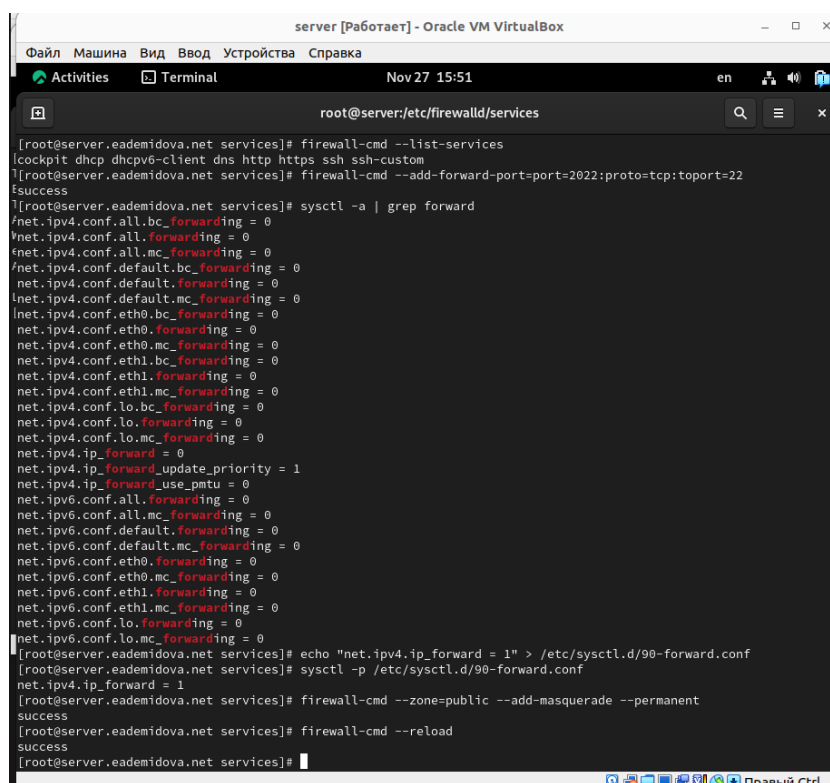


Рис. 3.5: Получение клиентом удаленного доступа по SSH к серверу через порт 2022

## 3.3 Настройка Port Forwarding и Masquerading

На сервере посмотрим, активирована ли в ядре системы возможность перенаправления IPv4-пакетов, затем включим пренаправление IPv4-пакетов на сервере и включим маскарадинг на сервере(3.6):



```
server [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Activities Terminal Nov 27 15:51 en
root@server:/etc/firewalld/services

[root@server.eademidova.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh ssh-custom
[root@server.eademidova.net services]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
success
[root@server.eademidova.net services]# sysctl -a | grep forward
net.ipv4.conf.all.bc_forwarding = 0
net.ipv4.conf.all.forwarding = 0
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.default.bc_forwarding = 0
net.ipv4.conf.default.forwarding = 0
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.eth0.bc_forwarding = 0
net.ipv4.conf.eth0.forwarding = 0
net.ipv4.conf.eth0.mc_forwarding = 0
net.ipv4.conf.eth1.bc_forwarding = 0
net.ipv4.conf.eth1.forwarding = 0
net.ipv4.conf.eth1.mc_forwarding = 0
net.ipv4.conf.lo.bc_forwarding = 0
net.ipv4.conf.lo.forwarding = 0
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.ip_forward = 0
net.ipv4.ip_forward_update_priority = 1
net.ipv4.ip_forward_use_pmtu = 0
net.ipv6.conf.all.forwarding = 0
net.ipv6.conf.all.mc_forwarding = 0
net.ipv6.conf.default.forwarding = 0
net.ipv6.conf.default.mc_forwarding = 0
net.ipv6.conf.eth0.forwarding = 0
net.ipv6.conf.eth0.mc_forwarding = 0
net.ipv6.conf.eth1.forwarding = 0
net.ipv6.conf.eth1.mc_forwarding = 0
net.ipv6.conf.lo.forwarding = 0
net.ipv6.conf.lo.mc_forwarding = 0
[root@server.eademidova.net services]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
[root@server.eademidova.net services]# sysctl -p /etc/sysctl.d/90-forward.conf
net.ipv4.ip_forward = 1
[root@server.eademidova.net services]# firewall-cmd --zone=public --add-masquerade --permanent
success
[root@server.eademidova.net services]# firewall-cmd --reload
success
[root@server.eademidova.net services]#
```

Рис. 3.6: Настройка перенаправления IPv4-пакетов и включение маскардинга

Теперь проверим доступность выхода в Интернет на клиенте.

### 3.4 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог firewall, в который поместим в соответствующие подкаталоги конфигурационные файлы FirewallD и создадим исполняемый файл firewall.sh(рис. 3.7)

```
success
[root@server.eademidova.net services]# cd /vagrant/provision/server
[root@server.eademidova.net server]# mkdir -p /vagrant/provision/server/firewall/etc/firewalld/services
[root@server.eademidova.net server]# mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
[root@server.eademidova.net server]# cp -r /etc/firewalld/services/ssh-custom.xml /vagrant/provision/server/firewall/etc/firewalld/services/
[root@server.eademidova.net server]# cp -r /etc/sysctl.d/90-forward.conf /vagrant/provision/server/firewall/etc/sysctl.d/
[root@server.eademidova.net server]# cd /vagrant/provision/server
[root@server.eademidova.net server]# touch firewall.sh
[root@server.eademidova.net server]# chmod +x firewall.sh
[root@server.eademidova.net server]#
```

Рис. 3.7: Создание окружения для внесения изменений в настройки окружающей среды

Открыв firewall.sh на редактирование, пропишем в нём следующий скрипт(3.8):

```
mc [root@server.eademidova.net]:/vagrant/provision/server
firewall.sh [-----] 19 L: [ 1+13 14/ 14] *(381 / 381b) <EOF>
#!/bin/bash

echo "Provisioning script $0"

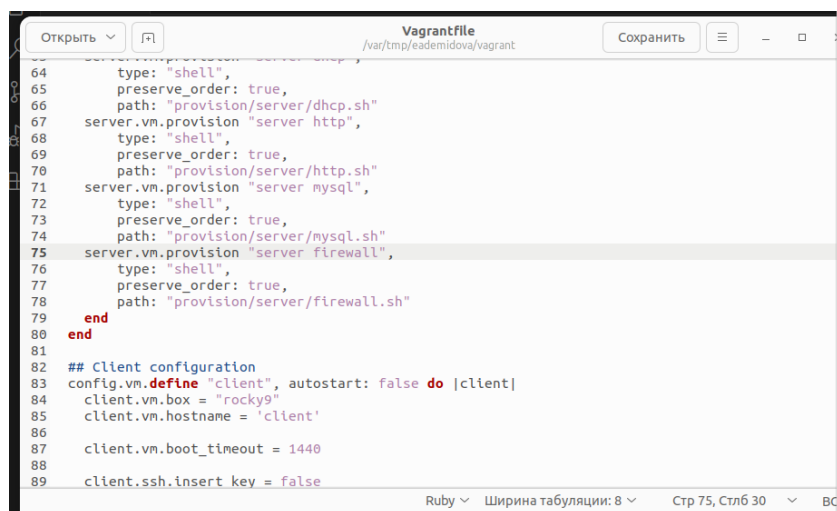
echo "Copy configuration files"
cp -R /vagrant/provision/server/firewall/etc/* /etc

echo "Configure masquerading"
firewall-cmd --add-service=ssh-custom --permanent
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22 --permanent
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload

restorecon -vR /etc
```

Рис. 3.8: Содержание firewall.sh

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера(3.9):



The image shows a code editor window titled "Vagrantfile" with the path "/var/tmp/esdemidova/vagrant". The editor contains Ruby code for Vagrant configuration. Lines 64-81 define a "server" VM with shell scripts for DHCP, HTTP, MySQL, and Firewall. Lines 82-89 define a "client" VM with configuration for Rocky Linux, hostname, boot timeout, and SSH settings. The status bar at the bottom indicates "Ruby", "Ширина табуляции: 8", "Стр 75, Стлб 30", and "BC".

```
64     type: "shell",
65     preserve_order: true,
66     path: "provision/server/dhcp.sh"
67   server.vm.provision "server http",
68     type: "shell",
69     preserve_order: true,
70     path: "provision/server/http.sh"
71   server.vm.provision "server mysql",
72     type: "shell",
73     preserve_order: true,
74     path: "provision/server/mysql.sh"
75   server.vm.provision "server firewall",
76     type: "shell",
77     preserve_order: true,
78     path: "provision/server/firewall.sh"
79   end
80 end
81
82 ## Client configuration
83 config.vm.define "client", autostart: false do |client|
84   client.vm.box = "rocky9"
85   client.vm.hostname = 'client'
86
87   client.vm.boot_timeout = 1440
88
89   client.ssh.insert_key = false
```

Рис. 3.9: Изменение файла Vagrantfile

## 4 Контрольные вопросы

1. Где хранятся пользовательские файлы firewalld?

`/usr/lib/firewalld/services/s`

2. Какую строку надо включить в пользовательский файл службы, чтобы указать порт TCP 2022?

`<port protocol="tcp" port="2022"/>`

3. Какая команда позволяет вам перечислить все службы, доступные в настоящее время на вашем сервере?

`firewall-cmd --get-services`

4. В чем разница между трансляцией сетевых адресов (NAT) и маскарadingом (masquerading)?

При маскарadingе вместо адреса отправителя(как делается это в NAT) динамически подставляется адрес назначенного интерфейса (сетевой адрес + порт).

5. Какая команда разрешает входящий трафик на порт 4404 и перенаправляет его в службу ssh по IP-адресу 10.0.0.10?

`sudo firewall-cmd --add-forward-port=port=4404:proto=tcp:toport=22:toaddr=10.0.0.`

6. Какая команда используется для включения маскарadingа IP-пакетов для всех пакетов, выходящих в зону public?

`firewall-cmd --zone=public --add-masquerade --permanent`

## 5 Выводы

В результате выполнения данной работы были приобретены практические навыки настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.