

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Администрирование сетевых подсистем

Студент: Бансимба Клодели Дьегра

Студ. билет № 1032215651

Группа: НПИбд-02-22

МОСКВА

2024 г.

Цель работы:

Целью данной работы является получение навыков настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.

Выполнение работы:

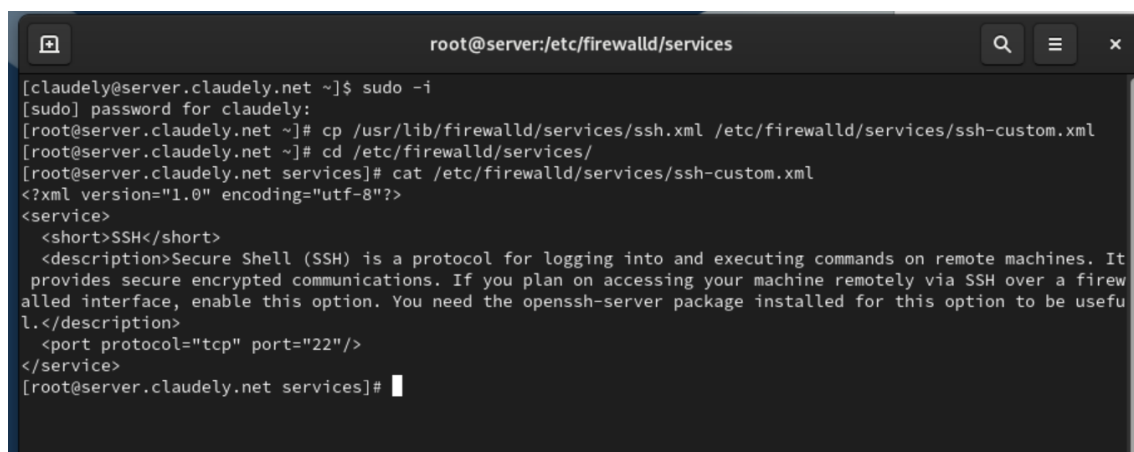
На основе существующего файла описания службы ssh создадим файл с собственным описанием:

```
cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml
```

```
cd /etc/firewalld/services/
```

Теперь посмотрим содержимое файла службы (Рис. 1.1):

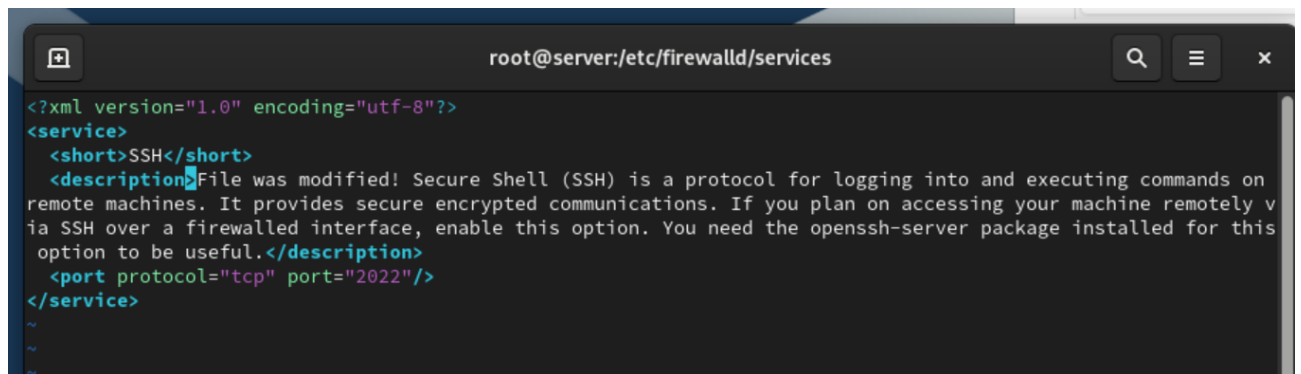
```
cat /etc/firewalld/services/ssh-custom.xml
```



```
root@server:/etc/firewalld/services
[claudely@server.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@server.claudely.net ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml
[root@server.claudely.net ~]# cd /etc/firewalld/services/
[root@server.claudely.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 firew
alled interface, enable this option. You need the openssh-server package installed for this option to be usefu
l.</description>
  <port protocol="tcp" port="22"/>
</service>
[root@server.claudely.net services]#
```

Рис. 1.1. Создание файла с собственным описанием на основе существующего файла описания службы ssh. Просмотр содержимого файла службы.

Откроем файл описания службы на редактирование и заменим порт 22 на новый порт (2022). В этом же файле скорректируем описание службы для демонстрации, что это модифицированный файл службы (Рис. 1.2):



```
root@server:/etc/firewalld/services
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>File was modified! 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 v
ia SSH over a firewalled interface, enable this option. You need the openssh-server package installed for this
option to be useful.</description>
  <port protocol="tcp" port="2022"/>
</service>
```

Рис. 1.2. Открытие файла описания службы на редактирование и замена порта 22 на новый порт (2022), корректирование описания службы для демонстрации, что это модифицированный файл службы.

Просмотрим список доступных FirewallD служб (Рис. 1.3):

`firewall-cmd --get-services`



```
[root@server.claudely.net services]#
[root@server.claudely.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 bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testn
et bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd cond
or-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls dock
er-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy fr
eeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master
git gpssd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs is
csi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver
kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nodep
ort-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap lda
ps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcac
he minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboar
d nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy
pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3ne
tsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt
-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snm
ptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui syncthin
g-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-client v
dsm vnc-server warpinator 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 zerot
ier
[root@server.claudely.net services]#
```

Рис. 1.3. Просмотр списка доступных FirewallD служб.

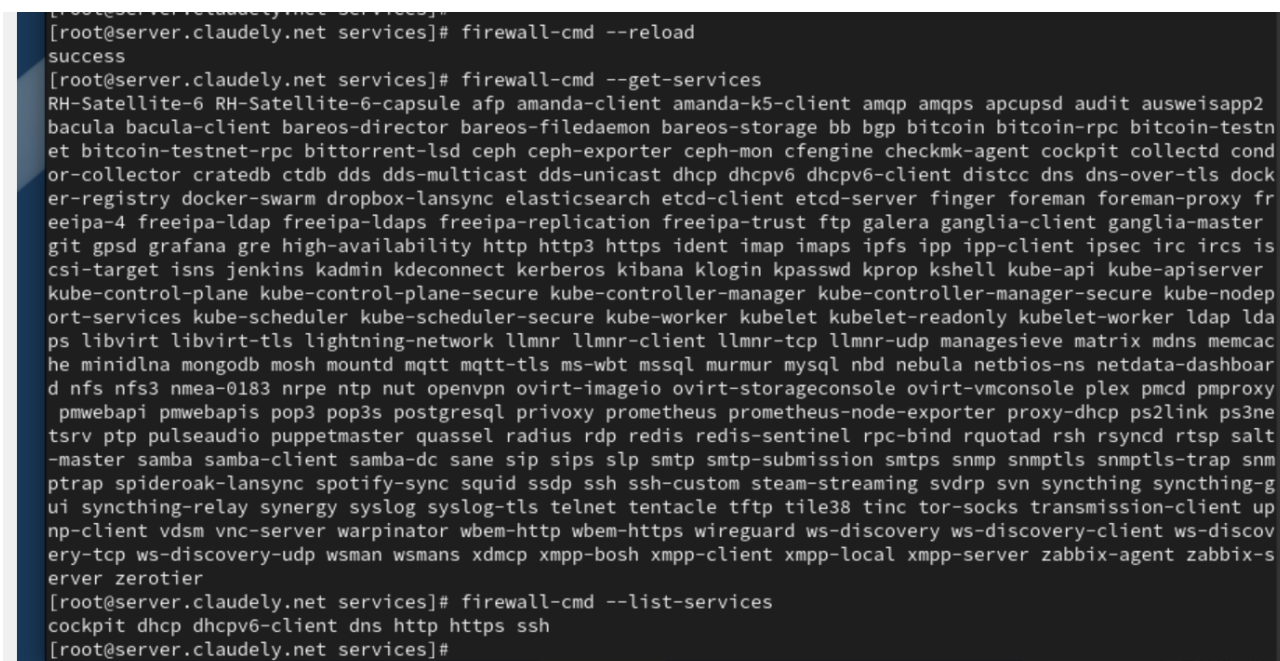
Перезагрузим правила межсетевого экрана с сохранением информации о состоянии и вновь выведем на экран список служб, а также список активных служб:

`firewall-cmd --reload`

```
firewall-cmd --get-services
```

```
firewall-cmd --list-services
```

Убедимся, что созданная нами служба отображается в списке доступных для FirewallD служб, но не активирована (Рис. 1.4):



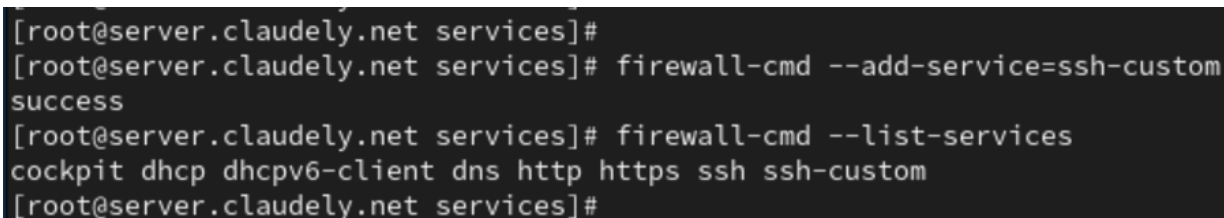
```
[root@server.claudely.net services]# firewall-cmd --reload
success
[root@server.claudely.net services]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2
bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testn
et bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd cond
or-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls dock
er-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy fr
eeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master
git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs is
csi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver
kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-nod
eport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap l
da ps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcac
he minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-dashboar
d nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd pmproxy
pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3ne
tsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp salt
-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap snm
ptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom steam-streaming svdrp svn syncthing syncthing-g
ui syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client up
np-client vdsm vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discov
ery-tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-s
erver zerotier
[root@server.claudely.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh
[root@server.claudely.net services]#
```

Рис. 1.4. Перегрузка правил межсетевого экрана с сохранением информации о состоянии, вывод на экран списка служб, а также списка активных служб.

Добавим новую службу в FirewallD и выведем на экран список активных служб (Рис. 1.5):

```
firewall-cmd --add-service=ssh-custom
```

```
firewall-cmd --list-services
```



```
[root@server.claudely.net services]#
[root@server.claudely.net services]# firewall-cmd --add-service=ssh-custom
success
[root@server.claudely.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh ssh-custom
[root@server.claudely.net services]#
```

Рис. 1.5. Добавление новой службы в FirewallD и вывод на экран списка активных служб.

Организуем на сервере переадресацию с порта 2022 на порт 22 (Рис. 2.1):

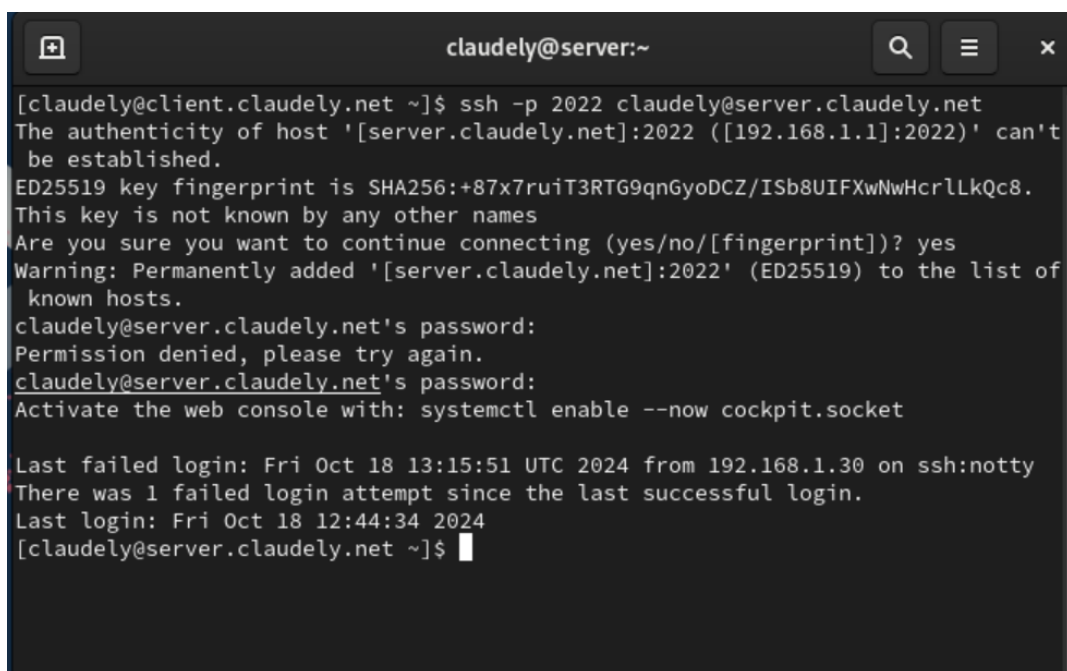
```
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
```

```
[root@server.claudely.net services]#  
[root@server.claudely.net services]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22  
success  
[root@server.claudely.net services]#  
[root@server.claudely.net services]#
```

Рис. 2.1. Организация переадресации на сервере с порта 2022 на порт 22.

На клиенте попробуем получить доступ по SSH к серверу через порт 2022 (Рис. 2.2):

```
ssh -p 2022 claudely@server.claudely.net
```



```
claudely@server:~  
[claudely@client.claudely.net ~]$ ssh -p 2022 claudely@server.claudely.net  
The authenticity of host '[server.claudely.net]:2022 ([192.168.1.1]:2022)' can't  
be established.  
ED25519 key fingerprint is SHA256:+87x7ruiT3RTG9qnGyoDCZ/ISb8UIFXwNwHcrLLkQc8.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '[server.claudely.net]:2022' (ED25519) to the list of  
known hosts.  
claudely@server.claudely.net's password:  
Permission denied, please try again.  
claudely@server.claudely.net's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Last failed login: Fri Oct 18 13:15:51 UTC 2024 from 192.168.1.30 on ssh:notty  
There was 1 failed login attempt since the last successful login.  
Last login: Fri Oct 18 12:44:34 2024  
[claudely@server.claudely.net ~]$
```

Рис. 2.2. Попытка получить на клиенте доступ по SSH к серверу через порт 2022.

На сервере посмотрим, активирована ли в ядре системы возможность перенаправления IPv4-пакетов (Рис. 3.1):

```
sysctl -a | grep forward
```

```
root@server:~  
[root@server.claudely.net ~]# 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.claudely.net ~]#
```

Рис. 3.1. Просмотр на сервере, активирована ли в ядре системы возможность перенаправления IPv4-пакетов.

Включим перенаправление IPv4-пакетов на сервере:

```
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
```

```
sysctl -p /etc/sysctl.d/90-forward.conf
```

Затем включим маскардинг на сервере (Рис. 3.2):

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

```
firewall-cmd --reload
```

```
net.ipv4.conf.lo.mc_forwarding = 0  
[root@server.claudely.net ~]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf  
[root@server.claudely.net ~]# sysctl -p /etc/sysctl.d/90-forward.conf  
net.ipv4.ip_forward = 1  
[root@server.claudely.net ~]# firewall-cmd --zone=public --add-masquerade --permanent  
success  
[root@server.claudely.net ~]# firewall-cmd --reload  
success  
[root@server.claudely.net ~]#
```

Рис. 3.2. Включение перенаправления IPv4-пакетов на сервере и маскарadingа на сервере.

На клиенте проверим доступность выхода в Интернет (Рис. 3.3):

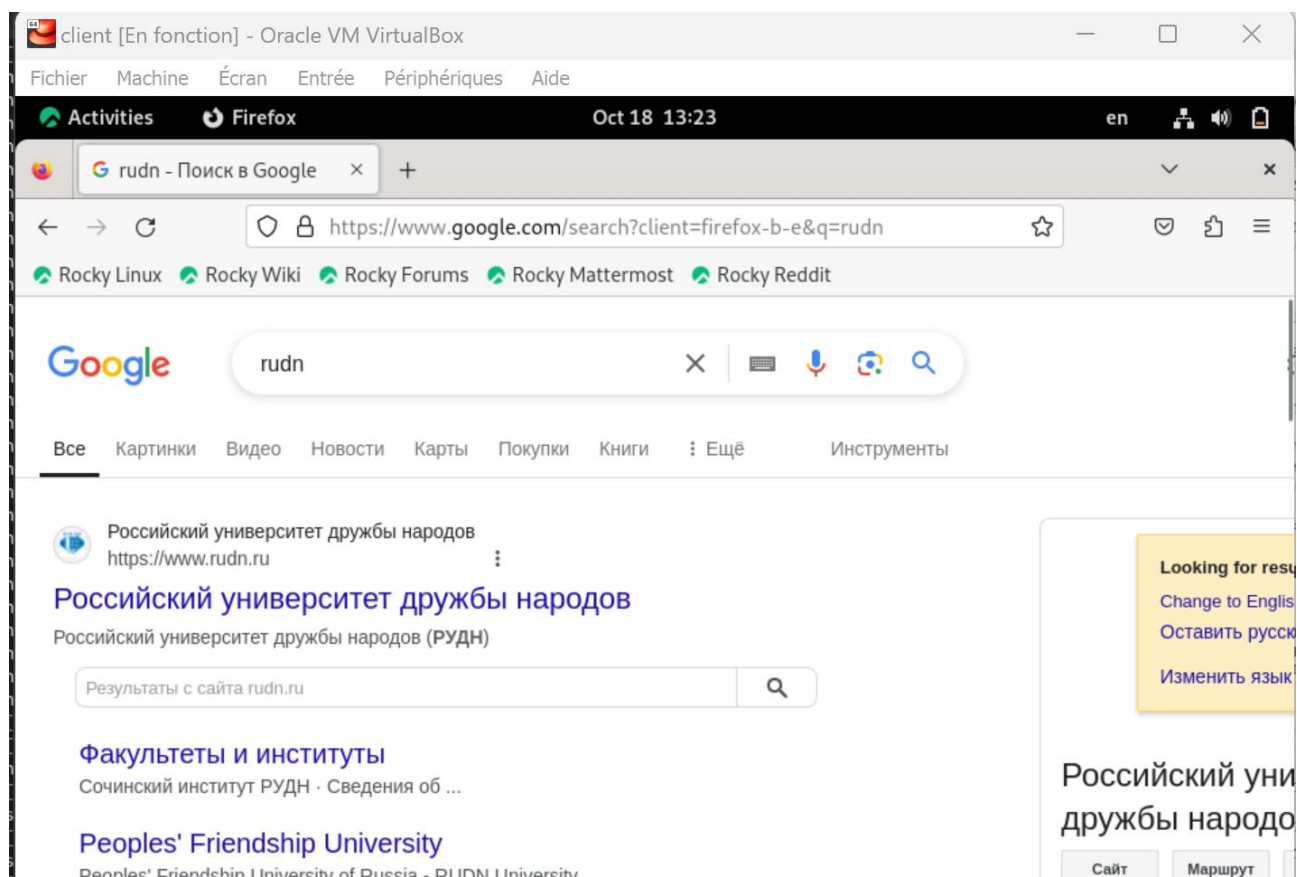


Рис. 3.3. Проверка доступности выхода в Интернет на клиенте.

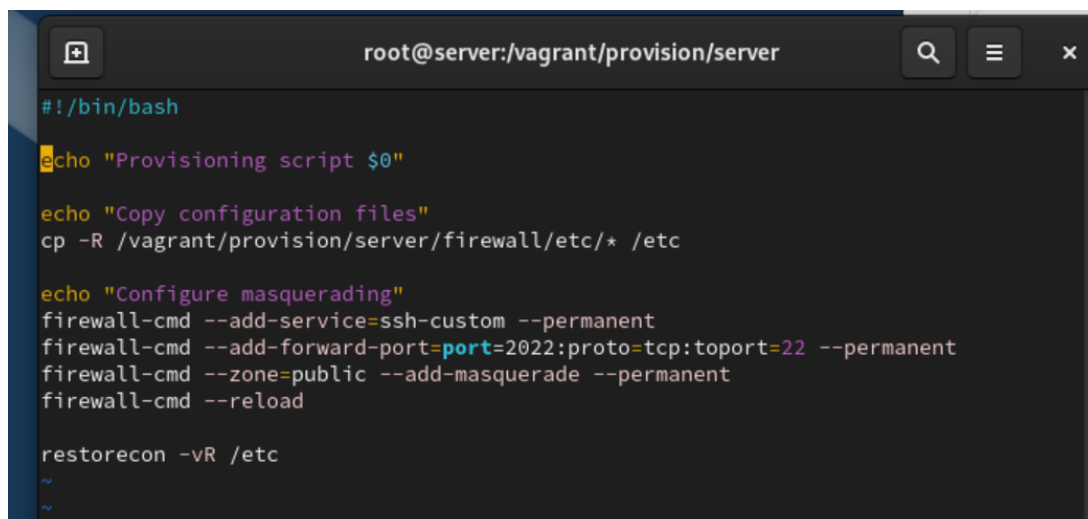
На виртуальной машине server перейдём в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `firewall`, в который поместим в соответствующие подкаталоги конфигурационные файлы `FirewallD`. В каталоге `/vagrant/provision/server` создадим файл `firewall.sh` (рис. 4.1):


```
[root@server.claudely.net ~]# cd /vagrant/provision/server
[root@server.claudely.net server]# mkdir -p /vagrant/provision/server/firewall/etc/firewalld/services
[root@server.claudely.net server]# mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
[root@server.claudely.net server]# cp -r /etc/firewalld/services/ssh-custom.xml /vagrant/provision/server/firewall/etc/firewalld/services/
[root@server.claudely.net server]# cp -r /etc/sysctl.d/90-forward.conf /vagrant/provision/server/firewall/etc/sysctl.d/
[root@server.claudely.net server]#
[root@server.claudely.net server]#
[root@server.claudely.net server]#
```

```
[root@server.claudely.net server]#
[root@server.claudely.net server]# cd /vagrant/provision/server
[root@server.claudely.net server]# touch firewall.sh
[root@server.claudely.net server]# chmod +x firewall.sh
```

Рис. 4.1. Открытие каталога для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создание в нём каталога `firewall`, в который помещаем в соответствующие подкаталоги конфигурационные файлы FirewallD. Создание в каталоге `/vagrant/provision/server` файла `firewall.sh`.

Открыв его на редактирование, пропишем в нём следующий скрипт из лабораторной работы (рис. 4.2):



```
root@server:/vagrant/provision/server
#!/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. 4.2. Открытие файла на редактирование и прописывание в нём скрипта из лабораторной работы.

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера (рис. 4.3):

Fichier	Modifier	Affichage
		<code>preserve_order: true, path: "provision/server/dns.sh"</code>
<code>server.vm.provision</code>		<code>"server dhcp", type: "shell", preserve_order: true, path: "provision/server/dhcp.sh"</code>
<code>server.vm.provision</code>		<code>"server http", type: "shell", preserve_order: true, path: "provision/server/http.sh"</code>
<code>server.vm.provision</code>		<code>"server mysql", type: "shell", preserve_order: true, path: "provision/server/mysql.sh"</code>
<code>server.vm.provision</code>		<code>"server firewall", type: "shell", preserve_order: true, path: "provision/server/firewall.sh"</code>

Рис. 4.3. Добавление записи в конфигурационном файле Vagrantfile.

Вывод:

В ходе выполнения лабораторной работы были получены навыки настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.

Ответы на контрольные вопросы:

1. Где хранятся пользовательские файлы firewalld? - В firewalld пользовательские файлы хранятся в директории /etc/firewalld/.
2. Какую строку надо включить в пользовательский файл службы, чтобы указать порт TCP 2022? - Для указания порта TCP 2022

в пользовательском файле службы, вы можете добавить строку в секцию port следующим образом:

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

3. Какая команда позволяет вам перечислить все службы, доступные в настоящее время на вашем сервере? - **Чтобы перечислить все службы, доступные в настоящее время на сервере с использованием firewalld, используйте команду:**

```
firewall-cmd --get-services
```

4. В чем разница между трансляцией сетевых адресов (NAT) и маскарadingом (masquerading)? - **Разница между трансляцией сетевых адресов (NAT) и маскарadingом (masquerading) заключается в том, что в случае NAT исходный IP-адрес пакета заменяется на IP-адрес маршрутизатора, а в случае маскарadingа используется IP-адрес интерфейса маршрутизатора.**

5. Какая команда разрешает входящий трафик на порт 4404 и перенаправляет его в службу ssh по IP-адресу 10.0.0.10? - **Для разрешения входящего трафика на порт 4404 и перенаправления его на службу SSH по IP-адресу 10.0.0.10, вы можете использовать команды:**

```
firewall-cmd --zone=public --add-port=4404/tcp --permanent
```

```
firewall-cmd --zone=public --add-forward-  
port=port=4404:proto=tcp:toport=22:toaddr=10.0.0.10 --  
permanent
```

```
firewall-cmd --reload
```

6. Какая команда используется для включения маскарадинга IP-пакетов для всех пакетов, выходящих в зону public? - **Для включения маскарадинга IP-пакетов для всех пакетов, выходящих в зону public, используйте следующую команду:**

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

firewall-cmd --reload