

# Лабораторная работа №7

Администрирование сетевых подсистем

---

Иванов Сергей Владимирович, НПИбд-01-23

09 октября 2025

Российский университет дружбы народов, Москва, Россия

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

1. Настройте межсетевой экран виртуальной машины server для доступа к серверу по протоколу SSH не через 22-й порт, а через порт 2022 (см. разделы 7.4.1 и 7.4.2).
2. Настройте Port Forwarding на виртуальной машине server (см. разделы 7.4.3).
3. Настройте маскарading на виртуальной машине server для организации доступа клиента к сети Интернет (см. раздел 7.4.3).
4. Напишите скрипт для Vagrant, фиксирующий действия по расширенной настройке межсетевого экрана. Соответствующим образом внести изменения в Vagrantfile (см. раздел 7.4.4).

## **Выполнение работы**

---

Загрузим операционную систему и перейдем в рабочий каталог с проектом. Запустим виртуальную машину server. (рис. 1).

```
C:\Users\lserg>cd C:\work_asp\svivanov\vagrant  
  
C:\work_asp\svivanov\vagrant>vagrant up server  
Bringing machine 'server' up with 'virtualbox' provider  
==> server: You assigned a static IP ending in ".1" or
```

**Рис. 1:** Запуск server

# Создание пользовательской службы firewalld

На виртуальной машине server войдем под пользователем и откроем терминал. Перейдем в режим суперпользователя. На основе существующего файла описания службы ssh создадим файл с собственным описанием (рис. 2).

```
[svivanov@server.svivanov.net ~]$ sudo -i  
[sudo] пароль для svivanov:  
[root@server.svivanov.net ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml  
[root@server.svivanov.net ~]# cd /etc/firewalld/services/  
[root@server.svivanov.net services]#
```

**Рис. 2:** Создание файла

# Создание пользовательской службы firewalld

Посмотрим содержимое файла службы: cat  
/etc/firewalld/services/ssh-custom.xml (рис. 3)

```
[root@server.svivanov.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 firewalled 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.svivanov.net services]#
```

**Рис. 3:** Файл ssh-custom.xml

## Создание пользовательской службы firewalld

Откроем файл описания службы на редактирование и заменим порт 22 на порт (2022). Скорректируем описание службы для демонстрации, укажем, что это модифицированный файл службы. (рис. 4)

```
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>This is modified file.</description>
  <port protocol="tcp" port="2022"/>
</service>
~
```

Рис. 4: Редактирование файла службы



# Создание пользовательской службы firewalld

Просмотрим список доступных FirewallD служб. Обратим внимание, что новая служба ещё не отображается в списке. (рис. 5)

```
[root@server.svivanov.net services]# firewall-cmd --get-services
0-AD RH-Satellite-6 RH-Satellite-6-capsule afp alvr amanda-client amanda-k5-client amqp amqps anno-1602 anno-1800
apcupsd aseqnet audit ausweisapp2 bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp b
itcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkm
k-agent civilization-iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unica
st dhcp dhcpv6 dhcpv6-client distcc dns dns-over-quit dns-over-tls docker-registry docker-swarm dropbox-lansync e
lasticsearch etcd-client etcd-server factorio finger foreman foreman-proxy freeipa-4 freeipa-ldap freeipa-ldaps f
reeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availability
http http3 https ident imap imapd iperf2 iperf3 ipfs ipp ipp-client ipsec irc ircs iscsi-target isns jenkins kadm
in 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-nodeport-services kube-scheduler kube-s
cheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-netw
ork llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns memcache minecraft minidlna mndp mongodb mosh
mountd mpd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula need-for-speed-most-wanted netbios-ns netdata-dashb
oard nfs nfs3 nmea-0183 nrpe ntp nut opentelemetry openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole ple
```

Рис. 5: Список доступных служб

Перегрузим правила межсетевого экрана с сохранением информации о состоянии и вновь выведем на экран список служб, а также список активных служб (рис. 6)

```
-https wireguard ws-discovery ws-discovery-client ws-discovery-host ws-  
ttp wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zab  
r zabbix-trapper zabbix-web-service zero-k zerotier  
[root@server.svivanov.net services]# firewall-cmd --list-services  
cockpit dhcp dhcpv6-client dns http https ssh  
[root@server.svivanov.net services]#
```

**Рис. 6:** Перезагрузка правил firewall

Добавим новую службу в FirewallD и выведем на экран список активных служб (рис. 7)

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

**Рис. 7:** Добавление службы в FirewallD

Перегрузим правила межсетевого экрана с сохранением информации о состоянии (рис. 8)

```
[root@server.svivanov.net services]# firewall-cmd --add-service=ssh-custom --permanent  
success  
[root@server.svivanov.net services]# firewall-cmd --reload  
success  
[root@server.svivanov.net services]#
```

**Рис. 8:** Перегрузка правил МЭ

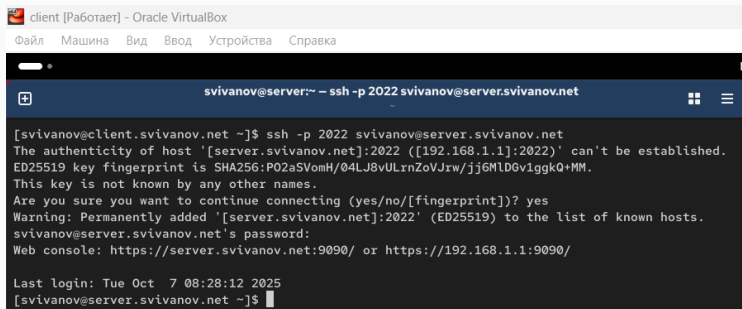
Организуем на сервере переадресацию с порта 2022 на порт 22 (рис. 9)

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

**Рис. 9:** Переадресация портов

# Перенаправление портов

На клиенте попробуем получить доступ по SSH к серверу через порт 2022 (рис. 10)



The screenshot shows a terminal window titled "client [Работает] - Oracle VirtualBox". The terminal displays the command `ssh -p 2022 svivanov@server.svivanov.net` and its output. The output indicates that the host's authenticity cannot be established because the fingerprint is not known. The user is prompted to confirm the connection, and they respond with "yes". A warning message states that the host has been permanently added to the list of known hosts. The terminal then prompts for the password, and the user provides it. Finally, the terminal shows the last login time and the user's prompt at the server.

```
client [Работает] - Oracle VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

svivanov@server:~ – ssh -p 2022 svivanov@server.svivanov.net

[svivanov@client.svivanov.net ~]$ ssh -p 2022 svivanov@server.svivanov.net
The authenticity of host '[server.svivanov.net]:2022 ([192.168.1.1]:2022)' can't be established.
ED25519 key fingerprint is SHA256:PO2aSVomH/04LJ8vULrnZoVJrw/jj6MLDgv1ggkQ+MM.
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.svivanov.net]:2022' (ED25519) to the list of known hosts.
svivanov@server.svivanov.net's password:
Web console: https://server.svivanov.net:9090/ or https://192.168.1.1:9090/

Last login: Tue Oct  7 08:28:12 2025
[svivanov@server.svivanov.net ~]$
```

**Рис. 10:** Доступ клиента по SSH к серверу

## Настройка Port Forwarding и Masquerading

На сервере посмотрим, активирована ли в ядре системы возможность перенаправления IPv4-пакетов. (рис. 11)

```
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
```

**Рис. 11:** Проверка возможностей перенаправления IPv4-пакетов

# Настройка Port Forwarding и Masquerading

Включим перенаправление IPv4-пакетов на сервере (рис. 12)

```
[root@server.svivanov.net services]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
[root@server.svivanov.net services]# sysctl -p /etc/sysctl.d/90-forward.conf
net.ipv4.ip_forward = 1
[root@server.svivanov.net services]#
```

**Рис. 12:** Перенаправление IPv4-пакетов на сервере



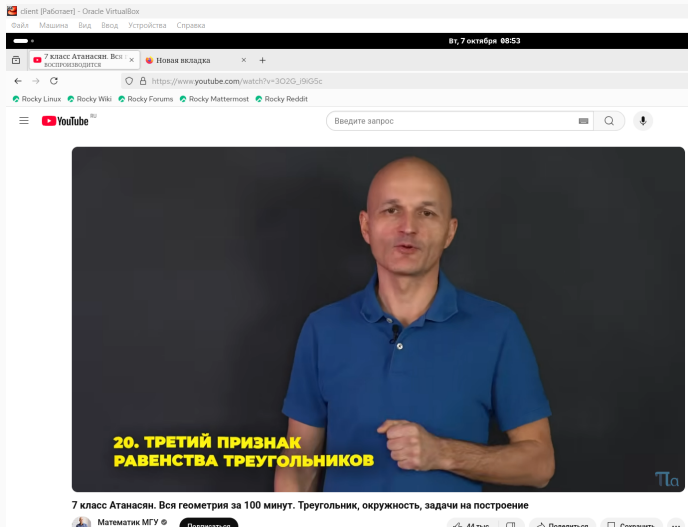
Включим маскарадинг на сервере (рис. 13)

```
[root@server.svivanov.net services]# firewall-cmd --zone=public --add-masquerade --permanent  
success  
[root@server.svivanov.net services]# firewall-cmd --reload  
success  
[root@server.svivanov.net services]#
```

**Рис. 13:** Включение маскарадинга

# Настройка Port Forwarding и Masquerading

На клиенте проверим доступность выхода в Интернет. (рис. 14)



# 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` (рис. 15)

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

**Рис. 15:** Создание каталогов для внесения изменений

# Внесение изменений в настройки внутреннего окружения виртуальной машины

В каталоге `/vagrant/provision/server` создадим файл `firewall.sh`. Открыв его на редактирование, пропишем в нём следующий скрипт. (рис. 16)

```
#!/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. 16: Скрипт `firewall.sh`

## Внесение изменений в настройки внутреннего окружения виртуальной машины

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера: (рис. 17)

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

Рис. 17: Редактирование Vagrantfile

## Вывод

---

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