

# **Отчёт по лабораторной работе 7**

**Расширенные настройки межсетевого экрана**

Суннатилло Махмудов

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	Создание пользовательской службы firewalld . . . . .	7
3.2	Перенаправление портов . . . . .	10
3.3	Настройка Port Forwarding и Masquerading . . . . .	10
3.4	Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	12
<b>4</b>	<b>Вывод</b>	<b>14</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>15</b>
<b>6</b>	<b>Список литературы</b>	<b>17</b>

## Список иллюстраций

3.1	Создание пользовательского файла службы . . . . .	7
3.2	Просмотр исходного содержимого файла службы . . . . .	8
3.3	Редактирование параметров пользовательской службы . . . . .	8
3.4	Просмотр доступных служб до перезагрузки . . . . .	9
3.5	Появление службы ssh-custom после перезагрузки правил . . . . .	9
3.6	Подключение по SSH через перенаправленный порт . . . . .	10
3.7	Включение пересылки пакетов и маскардинга . . . . .	11
3.8	Проверка доступа в Интернет после настройки маскардинга . . .	12
3.9	Создание каталогов и скрипта firewall.sh для сохранения конфигурации . . . . .	13

## **Список таблиц**

# 1 Цель работы

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

## 2 Теоретические сведения

Служба **firewalld** представляет собой динамически управляемый брандмауэр Linux, обеспечивающий гибкое управление сетевой безопасностью без необходимости перезапуска службы.

Она использует концепцию **зон** и **служб**, позволяя применять различные правила фильтрации трафика в зависимости от уровня доверия к сетям.

Файлы описания служб имеют формат **XML** и содержатся в каталогах:

- **/usr/lib/firewalld/services/** — системные шаблоны служб;
- **/etc/firewalld/services/** — пользовательские службы и изменения.

Создание пользовательской службы позволяет задать собственные порты и протоколы, отличные от стандартных, что используется, например, для переноса SSH на нестандартный порт (в работе — **2022**).

Механизм **Port Forwarding** обеспечивает перенаправление трафика с одного порта на другой, а **Masquerading** (маскарадинг) используется для подмены исходных IP-адресов пакетов, позволяя устройствам внутренней сети выходить в Интернет через общий внешний IP.

## 3 Выполнение лабораторной работы

### 3.1 Создание пользовательской службы firewalld

1. На виртуальной машине **server** был выполнен вход под пользователем *smahmudov* и произведён переход в режим суперпользователя.

После этого был создан собственный файл службы на основе системного описания **ssh.xml**:

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

```
[smahmudov@server: smahmudov.net ~]$ sudo -i
[sudo] password for smahmudov:
[root@server: smahmudov.net ~]# cp /usr/lib/firewalld/services/ssh.xml /etc/firewalld/services/ssh-custom.xml
[root@server: smahmudov.net ~]# cd /etc/firewalld/services/
[root@server: smahmudov.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: smahmudov.net services]# █
```

Рис. 3.1: Создание пользовательского файла службы

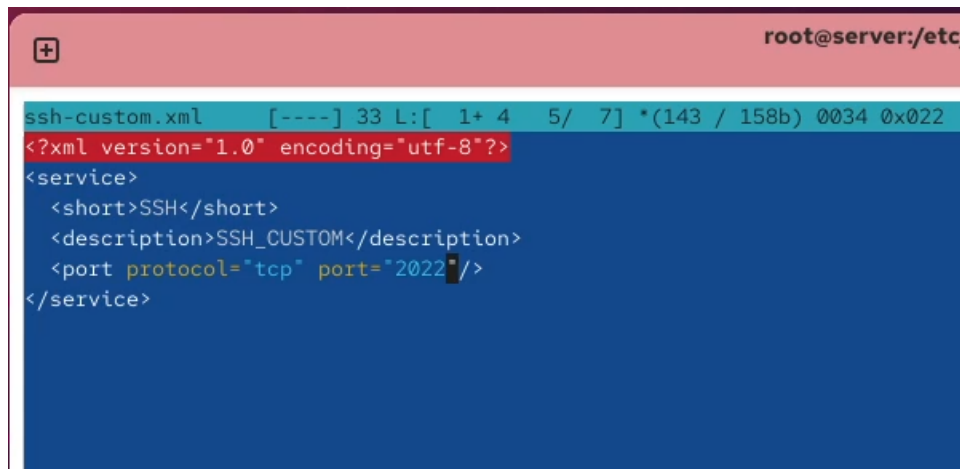
2. Было просмотрено содержимое нового файла **ssh-custom.xml**, чтобы проанализировать структуру описания службы.

Основные элементы XML:

- **<service>** — корневой элемент, в котором задаются параметры службы;
- **<short>** — краткое имя службы;
- **<description>** — описание назначения службы;
- **<port protocol="tcp" port="22"/>** — определяет используемый порт и протокол.

Пример содержимого:

SSH Secure Shell (SSH) is a protocol...



```
ssh-custom.xml [----] 33 L:[ 1+ 4 5/ 7] *(143 / 158b) 0034 0x022
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>SSH_CUSTOM</description>
  <port protocol="tcp" port="2022"/>
</service>
```

Рис. 3.2: Просмотр исходного содержимого файла службы

3. Файл **ssh-custom.xml** был отредактирован: описание изменено для указания, что это модифицированная служба, а номер порта заменён с **22** на **2022**.

SSH SSH\_CUSTOM



```
[root@server:smahmudov.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 ba
cula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter
ceph-mon cfengine checkmk-agent civilization-iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcp
v6-client distcc dns dns-over-quit dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server factorio finger foreman fo
reman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gpsd grafana gre high-availab
ility http http3 https ident imap imaps iperf2 iperf3 ipfs ipp ipp-client ipsec irc ircs iscsi-target ians jenkins kadmin kdeconnect kerberos kibana klogind k
passwd kppop 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-scheduler-secure kube-worker kubenet kubenet-readonly kubelet-worker ldap ldaps libvirt libvirt-tls lightning-network llmnr llm
nr-client llmnr-tcp llmnr-udp managiesieve matrix ndns memcached minecraft minidlna mnpd mongodb mosh mounsd mpd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd ne
bula need-for-speed-most-wanted netbios-ns netdata-dashboard nfs nfs3 nmea-0183 nripe ntp nut opentelemetry openvpn ovirt-imagelo ovirt-storageconsole ovirt-v
mconsole plex pmdc pmpoxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3netsrv ptp pulseaudio p
uppetmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane settlers-histo
ry-collection sip sips slimevr slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh statshv steam-
lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submission supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay synergy sysco
mlan syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-socks transmission-client turn turns upnp-client vdsml vnc-server vrrp waipinator wben-ht
tp wben-https wireguard ws-discovery ws-discovery-client ws-discovery-host ws-discovery-tcp ws-discovery-udp wdd wdd-http weman wsmans xdmcp xmpp-bosh xmpp
-client xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server:smahmudov.net services]#
```

Рис. 3.3: Редактирование параметров пользовательской службы

4. Для проверки наличия новой службы был выполнен просмотр списка всех доступных служб FirewallD:

firewall-cmd –get-services

На данном этапе служба **ssh-custom** ещё не отображалась в списке.



```
[root@server.smahmudov.net services]# firewall-cmd --reload
success
[root@server.smahmudov.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 ba
cula-client bareos-director bareos-filerdemon bareos-storage bb bdp bitcoin-bitcoin-rpc bitcoin-testnet-rpc bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter
ceph-non cfengine checkmk-agent civilization-iv civilization-v cockpit collectd condor-collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpc
v6-client distcc dns dns-over-qluc dns-over-tls docker-registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server factorio finger foreman fo
reman-proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master git gssd grafana gre high-availab
ility http http3 https ident imap imaps iperf2 iperf3 ipfs lpp lpp-client ipsec irc ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogind k
passwd kprop kshell kube-apt kube-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 llmnz llm
nr-client llmnz-tcp llmnz-udp managiesieve matrix mdns memcache minecraft minidlna mndp mongod mosh mountd mpd mqtt mqtt-tls ms-abt mssql murmur mysql nbd ne
bula need-for-speed-most-wanted netbios-ns netdata-dashboard nfs nfs3 nmap-0183 nrpe ntp nut opentelemetry openvpn ovirt-imaged ovirt-storageconsole ovirt-v
mconsole plex pncd pmpoxy pmmapi pmmapi3s pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link ps3metsrv ptp pulseaudio p
upptmaster quassel radius radsec rdp redis redis-sentinel rootd rpc-bind rquotad rsh rsyncd rtsp salt-master samba samba-client samba-dc sane settlers-histo
ry-collection sip sips slimevr slp smtp smtp-submission smtps snmp snmptls snmptls-trap snmptrap spideroak-lansync spotify-sync squid ssdp ssh ssh-custom sta
tsrv steam-lan-transfer steam-streaming stellaris stronghold-crusader stun stuns submission supertuxkart svdrp svn syncthing syncthing-gui syncthing-relay sy
nergy syscomlan syslog syslog-tls telnet tentacle terraria tftp tile38 tinc tor-socks transmission-client turn turns upnp-client vds vnc-server vrrp warpina
tor wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-host ws-discovery-tcp ws-discovery-udp wsd wsd-http wsmans xdmcp xmp
p-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-java-gateway zabbix-server zabbix-trapper zabbix-web-service zero-k zerotier
[root@server.smahmudov.net services]#
```

Рис. 3.4: Просмотр доступных служб до перезагрузки

5. Для обновления конфигурации FirewallD была выполнена команда:

`firewall-cmd --reload`

После этого новая служба появилась в общем списке доступных, что под-  
тверждает успешное считывание изменённого XML-файла.

`firewall-cmd --get-services`

```
[root@server.smahmudov.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh
[root@server.smahmudov.net services]# firewall-cmd --add-service=ssh-custom
success
[root@server.smahmudov.net services]# firewall-cmd --list-services
cockpit dhcp dhcpv6-client dns http https ssh ssh-custom
[root@server.smahmudov.net services]# firewall-cmd --add-service=ssh-custom --permanent
success
[root@server.smahmudov.net services]# firewall-cmd --reload
success
[root@server.smahmudov.net services]#
[root@server.smahmudov.net services]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
success
[root@server.smahmudov.net services]#
```

Рис. 3.5: Появление службы ssh-custom после перезагрузки правил

6. Затем пользовательская служба **ssh-custom** была добавлена в активные:

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

`firewall-cmd --list-services`

После успешного добавления конфигурация была сохранена навсегда:

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

`firewall-cmd --reload`

## 3.2 Перенаправление портов

1. На сервере была организована переадресация с порта **2022** на порт **22**, что позволяет подключаться к SSH через пользовательский порт:

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

После выполнения команды система сообщила об успешном применении перенаправления.

2. На клиентской машине было выполнено подключение по SSH через порт **2022**, что подтвердило корректность настроек:

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

После ввода пароля подключение было установлено, а система вывела приветственное сообщение с адресом веб-консоли сервера.

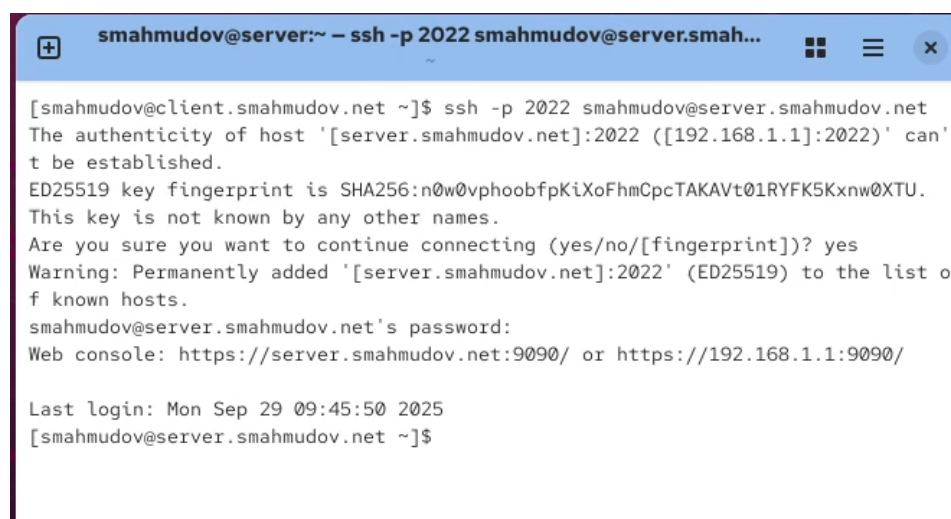


Рис. 3.6: Подключение по SSH через перенаправленный порт

## 3.3 Настройка Port Forwarding и Masquerading

1. На сервере была проверена текущая конфигурация перенаправления IPv4-пакетов:

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

Большинство параметров имели значение **0**, что означало, что пересылка пакетов была отключена.

2. Для включения пересылки IPv4-пакетов был создан конфигурационный файл:

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

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

В результате параметр **net.ipv4.ip\_forward** был установлен в значение **1**, что активировало возможность маршрутизации пакетов.

3. Далее был включён маскарading в публичной зоне FirewallD:

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

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

После применения настроек система выдала сообщение **success**, подтверждающее корректное выполнение команд.

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

Рис. 3.7: Включение пересылки пакетов и маскардинга

- После активации маскерадинга клиентская машина получила доступ в Интернет, что было проверено открытием сайта **rockylinux.org** в браузере.

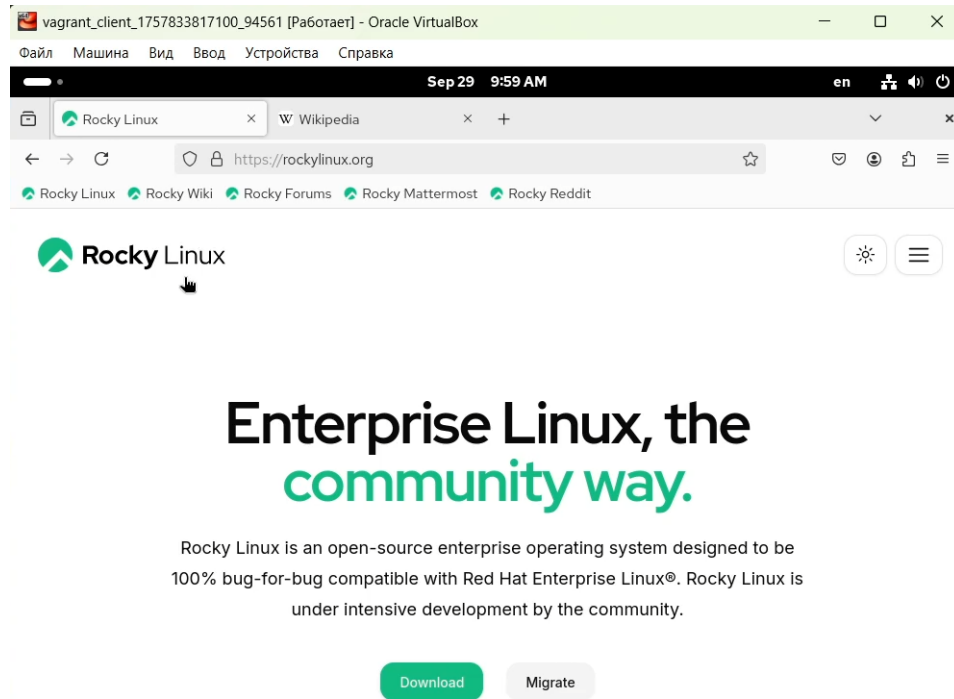


Рис. 3.8: Проверка доступа в Интернет после настройки маскерадинга

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

- На виртуальной машине **server** был выполнен переход в каталог внутреннего окружения:

```
cd /vagrant/provision/server/
```

В нём был создан каталог **firewall** с подкаталогами для хранения конфигурационных файлов:

```
mkdir -p /vagrant/provision/server/firewall/etc/firewalld/services
```

```
mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
```

Затем в созданные директории были скопированы соответствующие конфигурационные файлы:

```
cp -r /etc/firewalld/services/ssh-custom.xml /vagrant/provision/server/firewall/etc/firewalld/services/
cp -r /etc/sysctl.d/90-forward.conf /vagrant/provision/server/firewall/etc/sysctl.d/
```

2. В каталоге **/vagrant/provision/server** был создан скрипт **firewall.sh**, предназначенный для автоматического применения настроек:

```
touch firewall.sh
```

```
chmod +x firewall.sh
```

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

Рис. 3.9: Создание каталогов и скрипта firewall.sh для сохранения конфигурации

## 4 Вывод

В ходе лабораторной работы была выполнена настройка системы управления сетевой безопасностью **firewalld**, включая создание пользовательской службы **ssh-custom**, перенаправление портов и активацию механизма маскардинга. Реализовано подключение по SSH через нестандартный порт **2022** с автоматическим перенаправлением на порт **22**, а также включена маршрутизация и маскардинг IPv4-пакетов.

## 5 Контрольные вопросы

### 1. Где хранятся пользовательские файлы **firewalld**?

Пользовательские файлы служб **firewalld** хранятся в каталоге **/etc/firewalld/services/**. Этот каталог используется для размещения изменённых или собственных XML-файлов описания служб, в отличие от системных шаблонов, находящихся в **/usr/lib/firewalld/services/**.

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

Для указания порта **2022** в пользовательском файле службы необходимо добавить следующую строку в блок **<service>**:

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

Для вывода списка всех доступных служб используется команда:

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

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

**NAT (Network Address Translation)** — общий механизм преобразования IP-адресов, позволяющий устройствам внутренней сети обращаться к внешней, заменяя их внутренние адреса на публичные.

**Маскарадинг (Masquerading)** — частный случай NAT, при котором используется один общий внешний IP-адрес для всех исходящих соединений, при-

чём адрес подставляется динамически (обычно при подключении к интернету через роутер).

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

Для разрешения входящих соединений и перенаправления их на указанный адрес используется команда:

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

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

Для включения маскардинга в публичной зоне применяется следующая команда:

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



## 6 Список литературы

1. NAT: вопросы и ответы. — URL: [https://www.cisco.com/cisco/web/support/RU/9/92/92029\\_nat-faq.html](https://www.cisco.com/cisco/web/support/RU/9/92/92029_nat-faq.html) (дата обр. 13.09.2021).
2. Динамический брандмауэр с использованием FirewallD. — URL: <https://fedoraproject.org/wiki> (дата обр. 13.09.2021).
3. Одом У. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101. — М.: Вильямс, 2017. — 912 с. — (Cisco PressCore Series).
4. Часто задаваемые вопросы по технологии NAT / Сайт поддержки продуктов и технологий компании Cisco. — URL: [https://www.cisco.com/c/ru\\_ru/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html](https://www.cisco.com/c/ru_ru/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html) (дата обр. 13.09.2021).